Guida per gli sviluppatori

AWS Elastic Beanstalk

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Elastic Beanstalk: Guida per gli sviluppatori

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l’immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è AWS Elastic Beanstalk? ... 1
Piattaforme supportate ... 2
Flusso di lavoro di distribuzione delle applicazioni ... 3
Prezzi ... 3
Fasi successive ... 4
Tutorial sulle nozioni di base ... 5
Cosa costruirai ... 7
Fase 1 - Creare un'applicazione ... 7
Fase 2 - Implementa l'applicazione ... 10
Fase 3: Esplora l'ambiente ... 13
Risoluzione dei problemi con i log ... 15
Fase 4 - Aggiorna l'applicazione ... 17
Fase 5 - Ridimensiona la tua applicazione ... 18
Aumentare le impostazioni di capacità ... 19
Verifica l'aumento della capacità ... 20
Eliminazione ... 20
Fasi successive ... 22
Configurazione dell'EB CLI ... 23
Installa EB CLI ... 24
Installazione manuale ... 24
Installa in virtualenv ... 27
Installazione dell'interfaccia a riga di comando EB con Homebrew ... 28
Configurazione dell'interfaccia a riga di comando EB ... 29
File ignorati con .ebignore ... 31
Utilizzo di profili denominati ... 32
Distribuzione di artifact invece della cartella del progetto ... 32
Impostazioni di configurazione e precedenza ... 33
Metadati delle istanze ... 34
Utilizzo dell'interfaccia a riga di comando EB con Git ... 34
Associazione di ambienti Elastic Beanstalk ai rami Git ... 34
Distribuzione di modifiche ... 35
Utilizzo di moduli secondari Git ... 35
Assegnazione di tag Git alla tua versione dell'applicazione ... 36
Comandi della CLI EB ... 37

Opzioni comuni 38
eb abort 39
Description 39
Sintassi 39
Opzioni 39
Output 39
Esempio 39
eb appversion 40
Description 40
Sintassi 40
Opzioni 40
Utilizzo del comando in maniera interattiva 42
Output 43
Esempi 43
eb clone 44
Description 44
Sintassi 45
Opzioni 45
Output 47
Esempio 47
eb codesource 47
Description 47
Sintassi 48
Opzioni 48
Output 48
Esempi 48
eb config 49
Description 49
Sintassi 49
Opzioni 51
Output 54
Esempi 55
eb console 59
Description 59
Sintassi 59
Opzioni 59

eb create 59
Description 59
Sintassi 60
Opzioni 61
Output 74
Esempi 74
eb deploy 76
Description 76
Sintassi 77
Opzioni 77
Output 78
Esempio 78
eb events 79
Description 79
Sintassi 79
Opzioni 79
Output 79
Esempio 79
eb health 80
Description 80
Sintassi 81
Opzioni 81
Output 81
Esempio 81
eb init 83
Description 83
Sintassi 83
Opzioni 83
CodeBuild supporto 86
Output 86
Esempio 86
eb labs 88
Description 88
eb list 88
Description 88
Sintassi 88

Opzioni ... 88
Output ... 89
Esempio 1 ... 89
Esempio 2 ... 89
eb local ... 89
Description ... 89
Sintassi ... 90
Opzioni ... 90
Output ... 91
Esempi ... 92
eb logs ... 93
Description ... 93
Sintassi ... 93
Opzioni ... 94
Output ... 97
Esempi ... 97
eb migrate ... 98
Description ... 98
Sintassi ... 99
Sottocomandi ... 100
Opzioni ... 101
Output ... 109
Esempi ... 109
eb open ... 113
Description ... 113
Sintassi ... 114
Opzioni ... 114
Output ... 114
eb platform ... 114
Description ... 114
Utilizzo della piattaforma eb per piattaforme personalizzate ... 115
Utilizzo di piattaforme eb per gli ambienti ... 122
eb printenv ... 124
Description ... 124
Sintassi ... 125
Opzioni ... 125

Output 125
Esempio 125
eb restore 125
Description 125
Sintassi 125
Opzioni 126
Output 126
Esempio 126
eb scale 127
Description 127
Sintassi 127
Opzioni 127
Output 127
Esempio 127
eb setenv 128
Description 128
Sintassi 128
Opzioni 128
Output 128
Esempio 128
eb ssh 129
Description 129
Sintassi 130
Opzioni 130
Output 131
Esempio 131
eb status 132
Description 132
Sintassi 132
Opzioni 132
Output 133
Esempio 133
eb swap 134
Description 134
Sintassi 134
Opzioni 134

Output ... 135
Esempi ... 135
eb tags ... 135
Description ... 135
Sintassi ... 136
Opzioni ... 136
Output ... 138
Esempi ... 138
eb terminate ... 139
Description ... 139
Sintassi ... 140
Opzioni ... 140
Output ... 141
Esempio ... 141
eb upgrade ... 142
Description ... 142
Sintassi ... 142
Opzioni ... 142
Output ... 142
Esempio ... 143
eb use ... 143
Description ... 143
Sintassi ... 143
Opzioni ... 143
Concetti ... 145
Applicazione ... 146
Versione dell'applicazione ... 146
Ambiente ... 146
Livello di ambiente ... 146
Configurazione dell'ambiente ... 147
Configurazione salvata ... 147
Platform (Piattaforma) ... 147
Ambienti del server Web ... 147
Ambienti lavoratore ... 149
Considerazioni di natura progettuale ... 150
Scalabilità ... 151

Sicurezza ... 151
Storage persistente ... 152
Tolleranza ai guasti ... 153
Distribuzione di contenuti ... 154
Applicazione di patch e aggiornamenti software ... 154
Connettività ... 154
Gestione delle applicazioni ... 156
Console di gestione delle applicazioni ... 157
Gestione delle versioni delle applicazioni ... 158
Creazione di versioni delle applicazioni ... 159
Eliminazione delle versioni delle applicazioni ... 159
Ciclo di vita delle versioni ... 161
Tagging delle versioni delle applicazioni ... 163
Creazione di un bundle di origine ... 165
Creazione di un bundle di origine dalla riga di comando ... 166
Creazione di un bundle di origine con Git ... 167
Compressione dei file in Mac OS X Finder o in Windows Explorer ... 167
Creazione di un bundle di origine per un'applicazione .NET ... 168
Test del bundle di origine ... 169
Costruire con CodeBuild ... 170
Creazione di un’applicazione ... 170
Compilazione e distribuzione del codice dell'applicazione ... 170
Tagging delle applicazioni ... 172
Aggiunta di tag durante la creazione di applicazioni ... 173
Gestione dei tag di un'applicazione esistente ... 173
Applicazione di tag alle risorse ... 175
Risorse cui è possibile associare tag ... 176
Propagazione di tag ai modelli di avvio ... 176
Gestione degli ambienti ... 178
Console di gestione dell'ambiente ... 179
Accesso alla console ... 180
Pannello di panoramica dell'ambiente ... 181
Dettagli sull'ambiente ... 182
Operazioni ambiente ... 186
Creazione di ambienti ... 188
Procedura guidata per la creazione del nuovo ambiente ... 193

Clonazione di un ambiente ... 215
Terminazione di un ambiente ... 217
Con AWS CLI ... 219
Con l'API ... 221
URL Launch Now ... 225
Composizione di ambienti ... 229
Composizione degli ambienti ... 232
Distribuzioni ... 234
Scelta di una policy di distribuzione ... 236
Distribuzione di una nuova versione dell'applicazione ... 238
Ridistribuzione di una versione precedente ... 238
Altri modi per distribuire l'applicazione ... 239
Opzioni di implementazione ... 239
Implementazioni blu/verdi ... 248
Integrazione CI/CD ... 250
Modifiche di configurazione ... 253
Aggiornamenti continui ... 255
Aggiornamenti non modificabili ... 260
Aggiornamenti della piattaforma ... 264
Metodo 1: aggiornamento della versione della piattaforma dell'ambiente ... 268
Metodo 2: eseguire una distribuzione Blue/Green ... 269
Aggiornamenti gestiti ... 271
Aggiornamento di un ambiente legacy ... 278
Migrare a AL2 023/ AL2 ... 280
Domande frequenti sul ritiro della piattaforma ... 299
Annullamento di un aggiornamento ... 303
Ricompilazione di un ambiente ... 305
Ricompilazione di un ambiente in esecuzione ... 305
Ricompilazione di un ambiente terminato ... 306
Tipi di ambiente ... 308
Ambiente scalabile e con bilanciamento del carico ... 308
Ambiente di istanza singola ... 308
Modifica del tipo di ambiente ... 309
Ambienti lavoratore ... 310
Daemon SQS dell'ambiente lavoratore ... 313
Code di lettere morte ... 314

Attività periodiche ... 315
Usa Amazon CloudWatch per la scalabilità automatica nei livelli di ambiente di lavoro ... 317
Configurazione degli ambienti lavoratore ... 317
Collegamenti tra ambienti ... 321
Ripristino da uno stato di stack non valido ... 323
Risolvere l'errore ... 323
Perché si verifica l'errore ... 324
Configurazione degli ambienti ... 325
Risorse fornite ... 325
Configurazione con la console ... 327
Pagina di configurazione ... 327
Pagina Review Changes (Controlla modifiche) ... 329
EC2 Istanze Amazon ... 330
Tipi di EC2 istanze Amazon ... 331
Configurazione con la console ... 333
Gestione dei gruppi di sicurezza EC2 ... 341
Configurazione con AWS CLI ... 343
Configurazione con namespace ... 354
IMDS ... 355
Gruppo Auto Scaling ... 358
Modelli di avvio ... 360
Supporto per istanze Spot ... 363
Triggers ... 380
Operazioni pianificate ... 381
Impostazione del controllo dello stato ... 386
Bilanciatore del carico ... 387
Classic Load Balancer ... 389
Application Load Balancer ... 401
Application Load Balancer condiviso ... 422
Network Load Balancer ... 442
Configurazione del dual-stack ... 455
Configurazione dei log di accesso ... 466
Database ... 466
Ciclo di vita del database ... 468
Aggiunta di un'istanza database Amazon RDS all'ambiente tramite la console ... 468
Connessione al database ... 470

Configurazione di un'istanza database RDS integrata tramite la console 471
Configurazione di un'istanza database RDS integrata tramite i file di configurazione 472
Disaccoppiamento di un'istanza database RDS tramite la console 472
Disaccoppiamento di un'istanza database RDS tramite i file di configurazione 476
Sicurezza 478
Configurazione della sicurezza dell'ambiente 478
Spazi dei nomi della configurazione di sicurezza degli ambienti 481
Etichettatura degli ambienti 481
Aggiunta di tag durante la creazione dell'ambiente 482
Gestione di tag di un ambiente esistente 483
Variabili di ambiente e impostazioni del software 485
Configurare le impostazioni specifiche della piattaforma 486
Configurazione delle proprietà dell'ambiente (variabili di ambiente) 487
Spazi dei nomi delle impostazioni software 489
Accesso alle proprietà dell'ambiente 491
Debug 492
Visualizzazione dei log 495
Notifications 497
Configurazione delle notifiche utilizzando la console Elastic Beanstalk 499
Configurazione delle notifiche tramite le opzioni di configurazione 499
Configurazione delle autorizzazioni per l'invio di notifiche 502
Amazon VPC 504
Configurazione delle impostazioni VPC nella console Elastic Beanstalk 504
Spazio dei nomi aws:ec2:vpc 507
Migrazione da EC2 -Classic a un VPC 508
Nome dominio 513
Configurazione degli ambienti (avanzata) 515
Opzioni di configurazione 516
Priorità 516
Valori consigliati 517
Prima della creazione di un ambiente 519
Durante la creazione 525
Dopo la creazione 532
Opzioni generali 543
Opzioni specifiche della piattaforma 641
Opzioni personalizzate 655

.Ebextensions ... 657
Impostazioni opzioni ... 659
Server Linux ... 661
Server Windows ... 679
Risorse personalizzate ... 689
Configurazioni salvate ... 717
Tagging delle configurazioni salvate ... 723
`env.yaml` ... 725
Immagine personalizzata ... 728
Creazione di un'AMI personalizzata ... 729
Gestione di un'AMI personalizzata ... 733
Pulizia di un'AMI personalizzata ... 734
AMI basata su una piattaforma ritirata ... 734
File statici ... 741
Configurare i file statici utilizzando la console ... 741
Configurare i file statici utilizzando le opzioni di configurazione ... 742
HTTPS ... 743
Certificati server ... 745
Termina HTTPS sul sistema di bilanciamento del carico ... 752
Termina HTTPS sull'istanza ... 756
End-to-end crittografia ... 792
Passthrough TCP ... 797
Reindirizzamento da HTTP a HTTPS ... 798
Piattaforme ... 800
Glossario delle piattaforme ... 800
Modello di responsabilità condivisa ... 804
Policy di supporto per la piattaforma ... 805
Rami della piattaforma ritirati ... 805
Oltre il periodo di tolleranza di 90 giorni ... 806
Pianificazione della piattaforma ... 807
Risorse per la pianificazione ... 807
Prossime versioni di Platform Branch ... 808
Ritiro della pianificazione dei rami della piattaforma ... 808
Cronologia dei rami di piattaforma ritirati ... 810
Cronologia del server e del sistema operativo ... 815
Piattaforme supportate ... 817

Piattaforme supportate e cronologia dei componenti ... 818
Piattaforme Linux ... 818
Versioni di Amazon Linux supportate ... 819
Elenco delle piattaforme Elastic Beanstalk Linux ... 822
Flusso di lavoro di distribuzione dell'istanza ... 822
Flusso di lavoro di distribuzione delle istanze per ECS e versioni successive AL2 ... 825
Strumenti di script della piattaforma ... 828
Estensione delle piattaforme Linux ... 838
Buildfile e Procfile ... 838
Hook della piattaforma ... 840
File di configurazione ... 843
Configurazione del proxy inverso ... 843
Esempio di applicazione con estensioni ... 848
Distribuzione di.NET (Windows) ... 850
QuickStart per.NET Core su Windows ... 851
Vostro AWS account ... 852
Prerequisiti ... 852
Passaggio 1: Creare un'applicazione.NET Core per Windows ... 853
Passaggio 2: Esegui l'applicazione localmente ... 853
Passaggio 3: distribuire l'applicazione.NET Core su Windows con l'EB CLI ... 854
Passaggio 4: Esegui l'applicazione su Elastic Beanstalk ... 855
Fase 5: rimozione ... 855
AWS risorse per la tua applicazione ... 856
Fasi successive ... 856
Esegui la distribuzione con la console ... 857
QuickStart per ASP.NET ... 857
Vostro AWS account ... 858
Prerequisiti ... 858
Fase 1: Creare un' ASP.NET applicazione ... 859
Passaggio 2: Esegui l'applicazione localmente ... 860
Passaggio 3: Implementa l' ASP.NET applicazione con AWS Toolkit for Visual Studio ... 860
Passaggio 4: Esegui l'applicazione su Elastic Beanstalk ... 861
Fase 5: rimozione ... 861
AWS risorse per la tua applicazione ... 861
Fasi successive ... 862
Esegui la distribuzione con la console ... 862

Ambiente di sviluppo ... 863
Installazione di un IDE ... 863
Installazione di AWS Toolkit for Visual Studio ... 863
Piattaforma .NET ... 863
Configurazione dell'ambiente.NET nella console Elastic Beanstalk ... 864
Spazio dei nomi aws:elasticbeanstalk:container:dotnet:apppool ... 865
Migrazione tra versioni principali ... 866
Manifest di distribuzione ... 869
EC2 Avvio rapido ... 895
Aggiunta di un database ... 896
Aggiunta di un'istanza database all'ambiente ... 896
Download di un driver ... 898
Connessione a un database ... 898
La AWS Toolkit for Visual Studio ... 899
Test in locale ... 900
Creazione di un ambiente Elastic Beanstalk ... 900
Terminazione di un ambiente ... 910
Distribuzione ... 911
Gestione degli ambienti ... 914
Gestione degli account ... 927
Esegui il debug ... 928
Monitorare ... 930
Strumenti di distribuzione ... 932
Migrazione dell'applicazione locale ... 932
Raccomandazioni relative ai componenti ritirati ... 932
Filiali di Windows 2012 R2 ritirate ... 933
Compatibilità con TLS 1.2 ... 933
Distribuzione di.NET core (Linux) ... 935
QuickStart per.NET Core su Linux ... 935
Vostro AWS account ... 936
Prerequisiti ... 937
Passaggio 1: Creare un'applicazione.NET Core su Linux ... 937
Passaggio 2: Esegui l'applicazione localmente ... 938
Fase 3: Distribuire l'applicazione.NET Core su Linux con l'EB CLI ... 938
Fase 4: Esegui l'applicazione su Elastic Beanstalk ... 940
Fase 5: rimozione ... 855

AWS risorse per la tua applicazione ... 940
Fasi successive ... 941
Esegui la distribuzione con la console ... 941
Ambiente di sviluppo ... 942
Installazione dell'SDK NET Core ... 942
Installazione di un IDE ... 863
Installazione di AWS Toolkit for Visual Studio ... 863
Il core.NET sulla piattaforma Linux ... 942
Considerazioni su .NET Core sulla piattaforma Linux ... 943
Configurazione dell'ambiente .NET Core su Linux ... 944
NET Core nello spazio dei nomi di configurazione Linux ... 945
Raggruppamento delle applicazioni ... 946
Procfile ... 947
Server proxy ... 948
Il AWS Toolkit for Visual Studio ... 949
Prerequisiti ... 950
Creazione di un nuovo progetto di applicazione ... 950
Creazione di un ambiente Elastic Beanstalk e distribuzione dell'applicazione ... 951
Terminazione di un ambiente ... 953
Gestione degli ambienti ... 954
Monitorare ... 967
Migrazione da Windows a Linux ... 969
Considerazioni sulla migrazione a .NET Core su piattaforma Linux ... 969
Implementazione di Go ... 970
QuickStart per Go ... 970
Vostro AWS account ... 971
Prerequisiti ... 971
Passaggio 1: creare un'applicazione Go ... 972
Fase 2: Implementa l'applicazione Go con l'EB CLI ... 973
Fase 3: Esegui l'applicazione su Elastic Beanstalk ... 973
Fase 4: pulizia ... 855
AWS risorse per la tua applicazione ... 974
Fasi successive ... 975
Esegui la distribuzione con la console ... 975
Ambiente di sviluppo ... 975
Installazione di Go ... 975

Installazione dell' AWS SDK for Go ... 976
La piattaforma Go ... 976
Configurazione dell'ambiente Go ... 977
Spazi dei nomi per la configurazione di Go ... 978
Procfile ... 980
Buildfile ... 982
Configurazione del proxy ... 983
Implementazione di Java ... 985
QuickStart per Java ... 986
Vostro AWS account ... 987
Prerequisiti ... 987
Passaggio 1: creare un'applicazione Java ... 988
Fase 2: Esegui l'applicazione localmente ... 991
Fase 3: Distribuire l'applicazione Java con l'EB CLI ... 991
Fase 4: Esegui l'applicazione su Elastic Beanstalk ... 993
Fase 5: rimozione ... 993
AWS risorse per la tua applicazione ... 993
Fasi successive ... 994
Esegui la distribuzione con la console ... 994
QuickStart per Java su Tomcat ... 995
Vostro AWS account ... 995
Prerequisiti ... 996
Fase 1: Creare un'applicazione Java JSP ... 996
Fase 2: Distribuire l'applicazione Java JSP con l'EB CLI ... 997
Fase 3: Esegui l'applicazione su Elastic Beanstalk ... 998
Fase 4: pulizia ... 855
AWS risorse per la tua applicazione ... 998
Fasi successive ... 999
Esegui la distribuzione con la console ... 999
Ambiente di sviluppo ... 1000
Installazione di Java Development Kit ... 1000
Installazione di un container Web ... 1000
Download di librerie ... 1000
Installazione dell' AWS SDK for Java ... 1001
Installazione di un IDE o un editor di testo ... 1001
Esempi di applicazioni e tutorial ... 1002

Avvio di un ambiente con un'applicazione Java di esempio ... 1002
Fasi successive ... 1009
Piattaforma Tomcat ... 1010
Configurazione dell'ambiente Tomcat ... 1012
Spazio dei nomi della configurazione Tomcat ... 1015
Raggruppamento di file WAR ... 1018
Strutturazione della cartella di progetto ... 1019
Configurazione del proxy ... 1022
Piattaforma Java SE ... 1028
Configurazione dell'ambiente Java SE ... 1030
Spazio dei nomi per la configurazione di Java SE ... 1031
Buildfile ... 1033
Procfile ... 1033
Configurazione del proxy ... 1035
Aggiunta di un database ... 1038
Download del driver JDBC ... 1040
Connessione a un database (piattaforme Java SE) ... 1041
Connessione a un database (piattaforme Tomcat) ... 1042
Risoluzione dei problemi di connessione del database ... 1045
Resources ... 1047
Distribuzione di Node.js ... 1049
QuickStart per Node.js ... 1049
Vostro AWS account ... 1050
Prerequisiti ... 1051
Fase 1: Creare un' Node.js applicazione ... 1051
Passaggio 2: Esegui l'applicazione localmente ... 1052
Fase 3: Implementazione Node.js dell'applicazione con l'EB CLI ... 1052
Fase 4: Esegui l'applicazione su Elastic Beanstalk ... 1053
Fase 5: rimozione ... 855
AWS risorse per la tua applicazione ... 1054
Fasi successive ... 1055
Esegui la distribuzione con la console ... 1055
Ambiente di sviluppo ... 1055
Installa Node.js ... 1056
Conferma l'installazione di npm ... 1056
Installa l' AWS SDK per Node.js ... 1056

Installazione del generatore Express ... 1056
Configurazione di un framework e un server Express ... 1057
Piattaforma Node.js ... 1058
Configurazione dell'ambiente Node.js ... 1060
Spazi dei nomi per la configurazione di Node.js ... 1062
Procfile ... 1066
Configurazione delle dipendenze ... 1067
File NPM shrinkwrap ... 1072
Configurazione del proxy ... 1072
Esempi di applicazioni e tutorial ... 1076
Avvio di un ambiente con un'applicazione Node.js di esempio ... 1076
Fasi successive ... 1078
Tutorial - Express ... 1078
Prerequisiti ... 1078
Creazione di un ambiente Elastic Beanstalk ... 1079
Aggiornamento dell'applicazione per usare Express ... 1081
Aggiornamento dell'applicazione per usare Amazon RDS. ... 1085
Eliminazione ... 1090
Tutorial – Express con clustering ... 1090
Prerequisiti ... 1090
Creazione di un ambiente Elastic Beanstalk ... 1091
Aggiornamento dell'applicazione per usare Express ... 1093
Elimina ... 1107
Tutorial - Node.js con DynamoDB ... 1108
Prerequisiti ... 1109
Creazione di un ambiente Elastic Beanstalk ... 1109
Aggiungi le autorizzazioni alle istanze dell'ambiente ... 1112
Implementazione dell'applicazione di esempio ... 1113
Creazione di una tabella DynamoDB ... 1115
Aggiornamento dei file di configurazione dell'applicazione ... 1116
Configurazione dell'ambiente per l'elevata disponibilità ... 1118
Pulizia ... 1118
Fasi successive ... 1119
Aggiunta di un database ... 1120
Aggiunta di un'istanza database all'ambiente ... 1120
Download di un driver ... 1121

Connessione a un database ... 1122
Resources ... 1123
Implementazione di PHP ... 1124
QuickStart per PHP ... 1124
Vostro AWS account ... 1125
Prerequisiti ... 1125
Fase 1: Creare un'applicazione PHP ... 1125
Passaggio 2: Esegui l'applicazione localmente ... 1126
Passaggio 3: inizializza e distribuisci l'applicazione PHP ... 1126
Passaggio 4: sfoglia la tua applicazione cloud ... 1127
Passaggio 5: Aggiorna e ridistribuisci l'applicazione ... 1127
Eliminazione ... 1128
Fasi successive ... 1129
Piattaforma PHP ... 1129
Installazione dell' AWS SDK for PHP ... 1130
Considerazioni per PHP 8.1 su Amazon Linux 2 ... 1130
Configurazione dell'ambiente PHP ... 1133
Namespace per la configurazione ... 1135
Installazione delle dipendenze ... 1136
Aggiornamento di Composer ... 1137
Estensione di php.ini ... 1139
Esempi avanzati di ... 1139
Aggiunta di un database ... 1140
Tutorial - Laravel ... 1143
Tutorial - CakePHP ... 1153
Tutorial - Symfony ... 1161
Tutorial - Produzione HA ... 1167
Tutorial - HA WordPress ... 1178
Tutorial - HA Drupal ... 1195
Implementazione di Python ... 1213
QuickStart per Python ... 1213
Vostro AWS account ... 1214
Prerequisiti ... 1214
Fase 1: Creare un'applicazione Python ... 1216
Fase 2: Esegui l'applicazione localmente ... 1216
Passaggio 3: distribuisci la tua applicazione Python con l'EB CLI ... 1217

Fase 4: Esegui l'applicazione su Elastic Beanstalk 1218
Fase 5: rimozione 855
AWS risorse per la tua applicazione 1218
Fasi successive 1219
Esegui la distribuzione con la console 1220
Ambiente di sviluppo 1220
Prerequisiti 1221
Uso di un ambiente virtuale 1221
Configurazione di un progetto Python per Elastic Beanstalk 1223
Piattaforma Python 1223
Configurazione dell'ambiente Python 1225
Spazio dei nomi di configurazione di Python 1227
L'`python3`eseguibile 1229
Procfile 1229
Specifica delle dipendenze 1230
Tutorial - Flask 1232
Prerequisiti 1233
Configurazione di un ambiente virtuale Python con Flask 1233
Creazione di un'applicazione Flask 1235
Distribuzione del sito con la CLI EB 1237
Pulizia 1240
Fasi successive 1241
Tutorial - Django 1241
Prerequisiti 1242
Configurazione di un ambiente virtuale Python e installazione di Django 1242
Creazione di un progetto Django 1244
Configurazione dell'applicazione Django per Elastic Beanstalk 1246
Distribuzione del sito con la CLI EB 1247
Aggiornamento di un'applicazione 1250
Eliminazione 1254
Fasi successive 1255
Aggiunta di un database 1255
Aggiunta di un'istanza database all'ambiente 1256
Download di un driver 1257
Connessione a un database 1257
Resources 1258

Implementazione di Ruby ... 1259
Ambiente di sviluppo ... 1259
Installazione di Ruby ... 1260
Installazione dell' AWS SDK per Ruby ... 1261
Installazione di un IDE o un editor di testo ... 1261
Piattaforma Ruby ... 1262
Configurazione dell'ambiente Ruby ... 1263
Spazi dei nomi della configurazione Ruby ... 1266
Gemfile ... 1267
Procfile ... 1268
Tutorial - Rails ... 1269
Prerequisiti ... 1270
Conoscenza di base di Elastic Beanstalk ... 1270
Avvio di un ambiente Elastic Beanstalk ... 1271
Installazione di Rails e generazione di un sito Web ... 1273
Configurazione delle impostazioni di Rails ... 1277
Distribuzione dell'applicazione ... 1278
Pulizia ... 1278
Fasi successive ... 1279
Tutorial - Sinatra ... 1279
Prerequisiti ... 1279
Avvio di un ambiente Elastic Beanstalk ... 1280
Scrittura di un sito Web Sinatra Basic ... 1282
Distribuzione dell'applicazione ... 1283
Pulizia ... 1284
Fasi successive ... 1284
Aggiunta di un database ... 1284
Aggiunta di un'istanza database all'ambiente ... 1285
Download di un adattatore ... 1286
Connessione a un database ... 1287
Distribuzione con Docker ... 1288
Filiali della piattaforma Docker ... 1288
Ramificazioni della piattaforma ritirate in esecuzione sull'AMI Amazon Linux (AL1) ... 1289
Ramificazione della piattaforma Docker ... 1290
QuickStart per Docker ... 1290
QuickStart per Docker Compose ... 1296

Configurazione dell'immagine Docker ... 1305
Ramo della piattaforma gestita ECS ... 1311
Panoramica della piattaforma Docker gestita da ECS ... 1312
Risorse Amazon ECS create da Elastic Beanstalk ... 1313
`Dockerrun.aws.jsonfile` v2 ... 1313
Immagini Docker ... 1314
Distribuzioni di container non riuscite ... 1314
Estensione delle piattaforme Docker basate su ECS ... 1315
Configurazione Docker gestita da ECS per Elastic Beanstalk ... 1315
Esercitazione - Docker gestito da ECS ... 1328
Migrazione AL2 a ECS in esecuzione su 023 ... 1337
Autenticazione con archivi di immagini ... 1339
Repository Amazon ECR ... 1339
Gestione dei segreti AWS ... 1341
File Dockerrun.aws.json ... 1343
Immagini pubbliche di Amazon ECR ... 1346
Configurazione dell'ambiente ... 1347
Configurazione del software in ambienti Docker ... 1347
Riferimento alle variabili di ambiente nei container ... 1349
Utilizzo della funzione di interpolazione per le variabili di ambiente con Docker Compose .. 1351
Generazione di log per report sanitari avanzati con Docker Compose ... 1352
Registrazione personalizzata del contenitore Docker con Docker Compose ... 1352
Immagini Docker ... 1354
Configurazione degli aggiornamenti gestiti per gli ambienti Docker ... 1354
Spazio dei nomi di configurazione di Docker ... 1355
Configurazione Docker su AMI Amazon Linux (precedente ad Amazon Linux 2) ... 1356
Piattaforme legacy ... 1358
(Legacy) Migrazione a Docker in esecuzione su Amazon Linux 2 ... 1358
Contenitori Docker (Legacy) GlassFish ... 1363
Ambienti di monitoraggio ... 1373
Console di monitoraggio ... 1373
Grafici di monitoraggio ... 1374
Personalizzazione della console di monitoraggio ... 1375
Monitoraggio dello stato con CLI ... 1376
Lettura dell'output ... 1380
Visualizzazione dello stato Interattivo ... 1382

Opzioni di visualizzazione dello stato Interattivo ... 1384
Report dello stato di base ... 1384
Colori dello stato ... 1385
Controlli dello stato di Elastic Load Balancing ... 1386
Controlli dello stato per ambienti a singola istanza e con piano lavoratore ... 1387
Controlli aggiuntivi ... 1387
CloudWatch Metriche Amazon ... 1387
Monitoraggio e report avanzati sull'integrità ... 1389
Agente di stato Elastic Beanstalk ... 1391
Fattori che determinano lo stato delle istanze e dell'ambiente ... 1392
Personalizzazione delle regole di controllo dello stato ... 1395
Ruoli dello stato migliorato ... 1395
Autorizzazione stato avanzata ... 1396
Eventi dello stato migliorato ... 1397
Comportamento del reporting dello stato migliorato durante aggiornamenti, distribuzioni e dimensionamento ... 1398
Abilitazione dello stato avanzato ... 1399
Console dello stato ... 1402
Colori e stati indicanti l'integrità ... 1407
Parametri dell'istanza ... 1410
Regole dello stato migliorato ... 1413
CloudWatch ... 1418
Utenti API ... 1427
Formato dei log dello stato migliorato ... 1429
Notifiche e risoluzione dei problemi ... 1433
AI-Powered Analisi dell'ambiente ... 1435
Gestione degli allarmi ... 1440
Visualizza la cronologia delle modifiche ... 1442
Visualizzazione di eventi ... 1443
Console ... 1444
Riga di comando ... 1444
Monitoraggio delle istanze ... 1445
Visualizzazione dei log dalle istanze ... 1446
Posizione dei log sulle istanze Amazon EC2 ... 1448
Posizione dei log in Amazon S3 ... 1449
Impostazioni della rotazione dei log in Linux ... 1450

Estensione della configurazione predefinita per i task dei log ... 1451
Streaming di file di registro su Amazon CloudWatch Logs ... 1454
Log di distribuzione ... 1455
Operazioni supportate ... 1456
Contenuto dei log ... 1456
Console ... 1458
File di registro delle istanze ... 1459
AWS Integrazione dei servizi ... 1462
Panoramica dell'architettura ... 1462
CloudFront ... 1463
CloudTrail ... 1464
Informazioni su Elastic Beanstalk in CloudTrail ... 1464
Informazioni sulle voci dei file di log di Elastic Beanstalk ... 1465
CloudWatch ... 1466
CloudWatch Registri ... 1467
Prerequisito, ad esempio lo streaming dei log su Logs CloudWatch ... 1468
Come Elastic Beanstalk configura i log CloudWatch ... 1469
Trasmissione dei log delle istanze su Logs CloudWatch ... 1474
Risoluzione dei problemi CloudWatch di integrazione dei log ... 1476
Streaming di informazioni sullo stato dell'ambiente ... 1478
EventBridge ... 1481
Monitora una risorsa Elastic Beanstalk con EventBridge ... 1482
Esempio di modelli di eventi Elastic Beanstalk ... 1484
Esempio di eventi Elastic Beanstalk ... 1487
Mappatura dei campi evento Elastic Beanstalk ... 1488
AWS Config ... 1491
Configurazione AWS Config ... 1492
Configurazione AWS Config per registrare risorse Elastic Beanstalk ... 1492
Visualizzazione dei dettagli di configurazione di Elastic Beanstalk in AWS Config console . 1493
Valutazione delle risorse Elastic Beanstalk utilizzando AWS Config rules ... 1497
DynamoDB ... 1498
ElastiCache ... 1498
Amazon EFS ... 1499
File di configurazione ... 1500
File system crittografati ... 1501
Applicazioni di esempio ... 1501

Rimozione dei file system ... 1502
IAM ... 1502
Profili delle istanze ... 1503
Ruoli di servizio ... 1508
Uso di ruoli collegati ai servizi ... 1516
Policy utente ... 1528
Formato ARN ... 1536
Risorse e condizioni ... 1538
Controllo degli accessi basato su tag ... 1583
Esempio di policy gestite ... 1588
Esempio di policy specifiche per le risorse ... 1592
Accesso ai bucket S3 tra ambienti ... 1603
Amazon RDS ... 1605
Amazon RDS nel VPC predefinito ... 1607
Credenziali Amazon RDS e Secrets Manager ... 1613
Eliminazione di un'istanza Amazon RDS esterna ... 1613
Simple Storage Service (Amazon S3) ... 1614
Il bucket di account cliente Elastic Beanstalk Amazon S3 ... 1614
Contenuto del bucket di account cliente Elastic Beanstalk Amazon S3 ... 1614
Eliminazione di oggetti nel bucket Amazon S3 di Elastic Beanstalk ... 1615
Eliminazione del bucket Amazon S3 di Elastic Beanstalk ... 1616
Archivio parametri Secrets Manager e Systems Manager ... 1617
Recupera i segreti nelle variabili di ambiente ... 1618
Autorizzazioni IAM richieste ... 1628
Utilizzo di Secrets Manager e Systems Manager Parameter Store ... 1630
Risoluzione dei problemi segreti e delle variabili di ambiente ... 1632
Amazon VPC ... 1634
VPC pubblico ... 1635
VPC pubblico/privato ... 1636
VPC privato ... 1637
Host bastione ... 1639
Amazon RDS ... 1644
Endpoint VPC ... 1652
Policy di endpoint VPC ... 1656
Sicurezza ... 1666
Protezione dei dati ... 1667

Crittografia dei dati ... 1668
Riservatezza di Internet ... 1669
Gestione dell'identità e degli accessi ... 1669
AWS politiche gestite ... 1669
Registrazione di log e monitoraggio ... 1686
Reporting sullo stato migliorato ... 1686
Registri delle EC2 istanze di Amazon ... 1686
Notifiche per l'ambiente ... 1686
CloudWatch Allarmi Amazon ... 1687
AWS CloudTrail registri ... 1687
AWS X-Ray debug ... 1687
Convalida della conformità ... 1687
Resilienza ... 1688
Sicurezza dell'infrastruttura ... 1689
Modello di responsabilità condivisa ... 1689
Best practice di sicurezza ... 1689
Best practice relative alla sicurezza preventiva ... 1690
Best practice relative alla sicurezza di rilevamento ... 1691
Permissions ... 1693
Ruoli richiesti ... 1693
Politiche e ruoli opzionali ... 1693
Ruolo del servizio ... 1694
`AWSElasticBeanstalkEnhancedHealth` ... 1509
`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` ... 1509
Profilo dell'istanza ... 1696
Policy gestite ... 1696
Creazione di un profilo di istanza EC2 ... 1697
Policy utente ... 1697
Esempi e tutorial ... 1698
Migrazione delle applicazioni IIS ... 1701
Prerequisiti ... 1702
Glossario sulla migrazione ... 1704
Termini per Windows, IIS e.NET ... 1704
Termini di Elastic Beanstalk ... 1708
Termini in Python ... 1712
Migrazioni IIS di base ... 1713

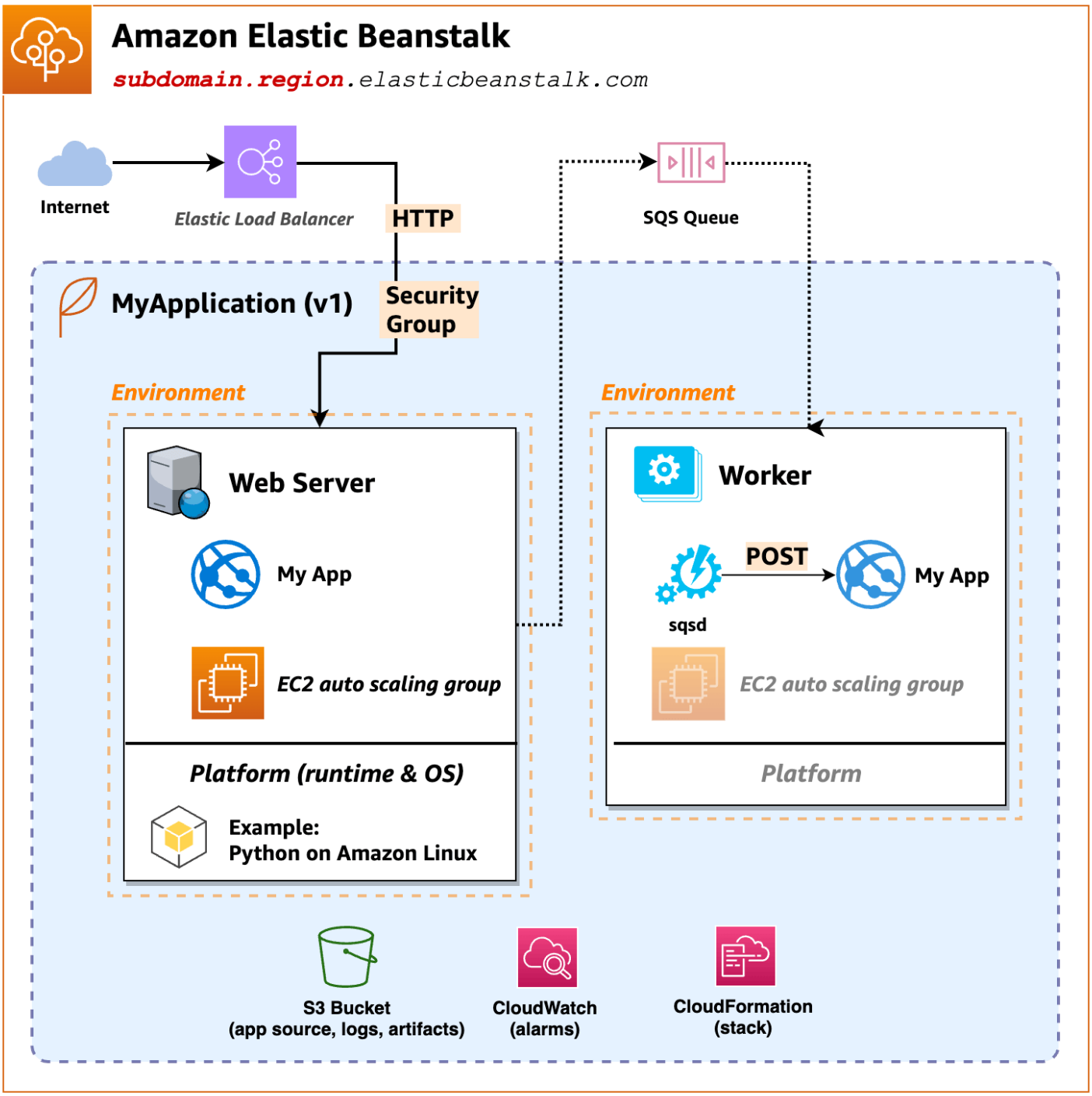
Esplorazione del tuo ambiente IIS ... 1714
Preparativi per la migrazione ... 1715
La tua prima migrazione ... 1716
Controllo della migrazione ... 1717
Monitoraggio dell'avanzamento ... 1717
Verifica della migrazione ... 1718
Gestione degli artefatti della migrazione ... 1719
Configurazione della rete ... 1719
Configurazione VPC ... 1719
Implementazioni multisito con configurazioni di porte ... 1724
Configurazione e dipendenze condivise ... 1726
Best practice ... 1727
Risoluzione dei problemi ... 1727
Configurazione di sicurezza ... 1728
Configurazione del profilo dell'istanza ... 1728
Gestione dei ruoli di servizio ... 1729
Configurazione del gruppo di sicurezza ... 1730
Integrazione del certificato SSL ... 1731
Autenticazione Windows ... 1731
Best practice e risoluzione dei problemi ... 1732
Mappatura della migrazione da IIS a Elastic Beanstalk ... 1734
Siti e applicazioni IIS in Elastic Beanstalk ... 1735
Gestione delle directory virtuali e dei percorsi delle applicazioni ... 1736
Riscrittura degli URL e routing delle richieste di applicazioni (ARR) ... 1737
Struttura degli artefatti di migrazione ... 1738
Scenari di migrazione avanzati ... 1739
Migrazioni multisito con Application Request Routing (ARR) ... 1739
Migrazioni multisito senza ARR utilizzando il routing basato su host ... 1743
Gestione delle directory virtuali ... 1747
Impostazioni personalizzate del pool di applicazioni ... 1751
Distribuzione delle versioni precedenti ... 1753
Risoluzione dei problemi e diagnostica ... 1754
Associazione di una coppia di chiavi all'ambiente EC2 ... 1754
Accesso ai log ... 1755
Accesso agli artefatti lato client ... 1755
Monitoraggio dello stato dell'ambiente ... 1755

EC2 ottimizzazione delle prestazioni 1756
Configurazione del volume EBS 1756
Problemi e soluzioni comuni 1757
Ottenere supporto 1758
Opzioni di migrazione: EB CLI e MGN 1758
Quando utilizzare ciascuna opzione di migrazione 1759
Confronto dei flussi di migrazione 1760
Conclusioni 1761
Risoluzione dei problemi 1762
Utilizzo dell'analisi dell'ambiente basata sull'intelligenza artificiale 1763
Utilizzo dello strumento Systems Manager 1763
Informazioni generali 1765
Variabili di ambiente per i segreti 1766
Creazione dell'ambiente 1766
Distribuzioni 1767
Integrità 1768
Bilanciatori di carico configurati a doppio stack 1770
Configurazione 1770
Docker 1771
Domande frequenti 1772
Risoluzione dei problemi di distribuzione 1773
Errori di distribuzione 1774
Resources 1777
Applicazioni di esempio 1778
AWS SDK per Java 1778
AWS SDK per .NET 1779
AWS Toolkit for Visual Studio 1779
AWS SDK per JavaScript in Node.js 1779
AWS SDK per PHP 1780
AWS SDK per Python (Boto) 1780
AWS SDK per Ruby 1780
Contenuti archiviati 1781
Piattaforme personalizzate (ritirate) 1781
Cronologia dei documenti 1799
..... mdcccix

Che cos'è AWS Elastic Beanstalk?

Con Elastic Beanstalk puoi distribuire applicazioni web AWS nel cloud su una varietà di piattaforme supportate. Crei e distribuisci le tue applicazioni. Elastic Beanstalk effettua il provisioning EC2 delle istanze Amazon, configura il bilanciamento del carico, imposta il monitoraggio dello stato e ridimensiona dinamicamente l'ambiente.

Oltre agli ambienti di server Web, Elastic Beanstalk fornisce anche ambienti di lavoro che è possibile utilizzare per elaborare i messaggi da una coda Amazon SQS, utili per attività asincrone o di lunga durata. Per ulteriori informazioni, consulta Ambienti worker Elastic Beanstalk.



Piattaforme supportate

Elastic Beanstalk supporta applicazioni sviluppate in Go, Java, .NET, Node.js, PHP, Python e Ruby. Elastic Beanstalk supporta anche i contenitori Docker, in cui è possibile scegliere il linguaggio di programmazione e le dipendenze delle applicazioni. Quando distribuisci la tua applicazione, Elastic

Beanstalk crea la versione della piattaforma supportata selezionata e fornisce una o AWS più risorse, come le istanze EC2 Amazon, nel tuo account per eseguire l'applicazione. AWS

Puoi interagire con Elastic Beanstalk tramite la AWS Command Line Interface console Elastic Beanstalk, la () o l'EB CLI AWS CLI, uno strumento a riga di comando di alto livello progettato specificamente per Elastic Beanstalk.

Puoi eseguire la maggior parte delle attività di distribuzione, come la modifica delle dimensioni della tua flotta di EC2 istanze Amazon o il monitoraggio dell'applicazione, direttamente dall'interfaccia web Elastic Beanstalk (console).

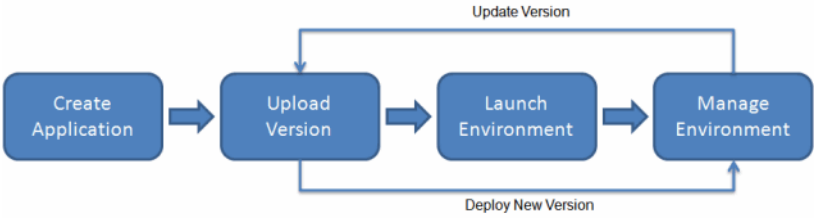
Per ulteriori informazioni su come distribuire un'applicazione Web di esempio utilizzando Elastic Beanstalk, consulta. Scopri come iniziare a usare Elastic Beanstalk

Flusso di lavoro di distribuzione delle applicazioni

Per utilizzare Elastic Beanstalk, devi creare un'applicazione, quindi caricare il pacchetto sorgente dell'applicazione su Elastic Beanstalk. Successivamente, fornisci informazioni sull'applicazione ed Elastic Beanstalk avvia automaticamente un ambiente e crea e AWS configura le risorse necessarie per eseguire il codice.

Dopo aver creato e distribuito l'applicazione e avviato l'ambiente, è possibile gestire l'ambiente e distribuire nuove versioni dell'applicazione. Le informazioni sull'applicazione, tra cui metriche, eventi e stato dell'ambiente, sono rese disponibili tramite la console Elastic Beanstalk e le interfacce a riga di comando. APIs

Il diagramma seguente illustra il flusso di lavoro di Elastic Beanstalk:



Prezzi

Non sono previsti costi aggiuntivi per l'utilizzo di Elastic Beanstalk. Paghi solo per le AWS risorse di base utilizzate dall'applicazione. Per informazioni dettagliate sui prezzi, consulta la pagina dei dettagli del servizio Elastic Beanstalk.

Fasi successive

Ti consigliamo il tutorial per iniziare a usare Elastic Beanstalk. Tutorial sulle nozioni di base Il tutorial illustra la creazione, la visualizzazione e l'aggiornamento di un'applicazione Elastic Beanstalk di esempio.

Scopri come iniziare a usare Elastic Beanstalk

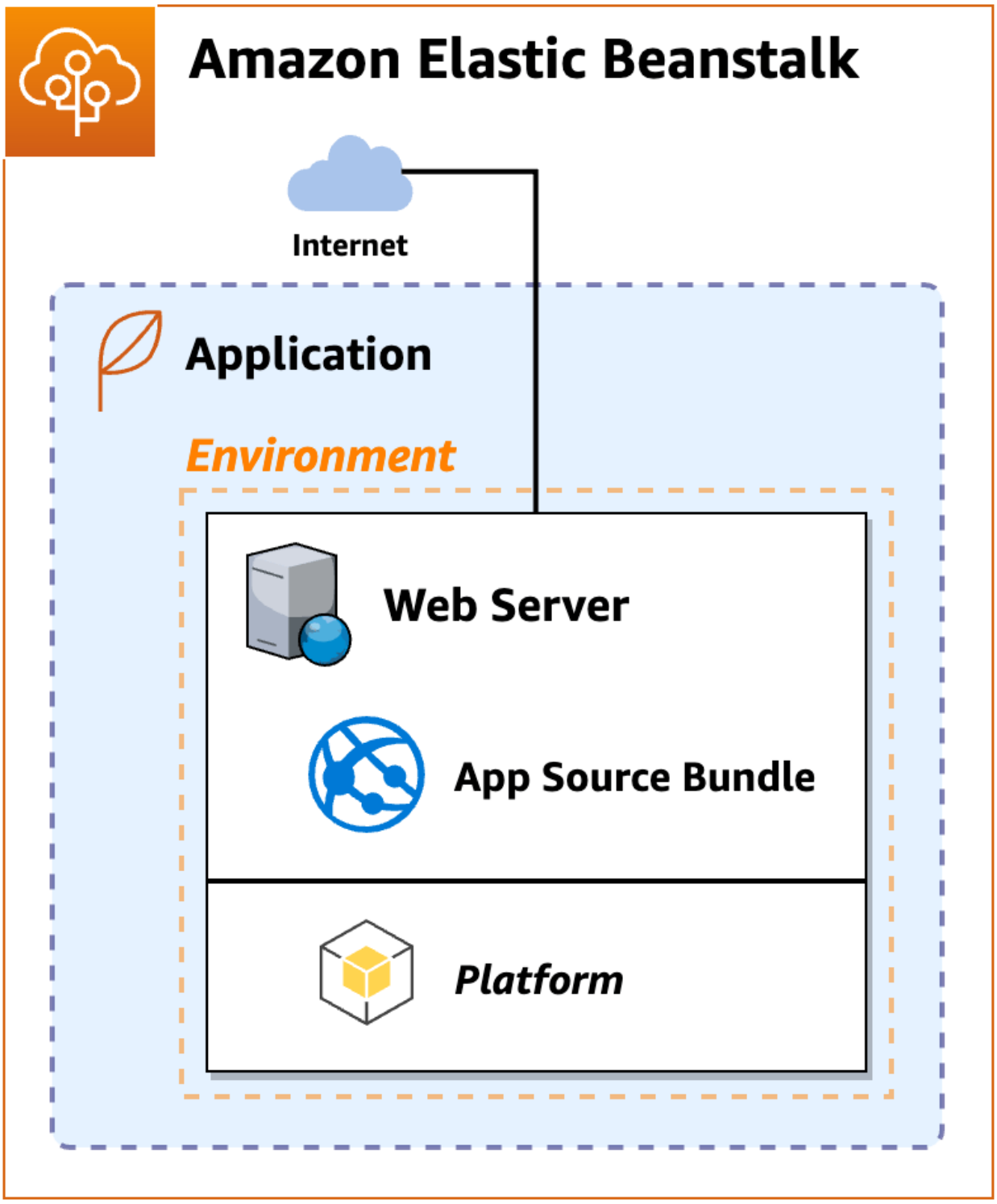
Con Elastic Beanstalk puoi distribuire, monitorare e scalare applicazioni e servizi Web. In genere, svilupperai il codice localmente e poi lo distribuirai su istanze di server Amazon EC2. Queste istanze, chiamate anche ambienti, vengono eseguite su piattaforme che possono essere aggiornate tramite la console o la AWS riga di comando.

Per iniziare, consigliamo di distribuire un'applicazione di esempio preconfigurata direttamente dalla console. Quindi, puoi imparare a sviluppare localmente e distribuire dalla riga di comando di. the section called “QuickStart per PHP”

L'utilizzo di Elastic Beanstalk è gratuito, ma le tariffe standard AWS si applicano alle risorse create nel corso di questo tutorial fino a quando non le elimini alla fine. Le spese totali sono in genere inferiori a un dollaro. Per informazioni su come ridurre al minimo i costi, consulta Piano gratuito di AWS..

Dopo aver completato questo tutorial, comprenderai le nozioni di base per creare, configurare, distribuire, aggiornare e monitorare un'applicazione Elastic Beanstalk con ambienti in esecuzione su istanze Amazon EC2.

Durata stimata: 35-45 minuti


Amazon Elastic Beanstalk
Internet
Application
Environment
Web Server
App Source Bundle
Platform

Cosa costruirai

La tua prima applicazione Elastic Beanstalk sarà costituita da un singolo ambiente Amazon EC2 che esegue l'esempio PHP su una piattaforma gestita da PHP.

Applicazione Elastic Beanstalk

Un'applicazione Elastic Beanstalk è un contenitore per i componenti Elastic Beanstalk, inclusi gli ambienti in cui il codice dell'applicazione viene eseguito su piattaforme fornite e gestite da Elastic Beanstalk o in contenitori personalizzati forniti dall'utente.

Ambiente

Un ambiente Elastic Beanstalk è una AWS raccolta di risorse che funzionano insieme, inclusa un'istanza Amazon EC2. Quando crei un ambiente, Elastic Beanstalk fornisce le risorse necessarie al tuo account. AWS

Platform (Piattaforma)

Una piattaforma è una combinazione di sistema operativo, runtime del linguaggio di programmazione, server Web, server applicazioni e componenti Elastic Beanstalk aggiuntivi. Elastic Beanstalk fornisce piattaforme gestite oppure puoi fornire la tua piattaforma in un container.

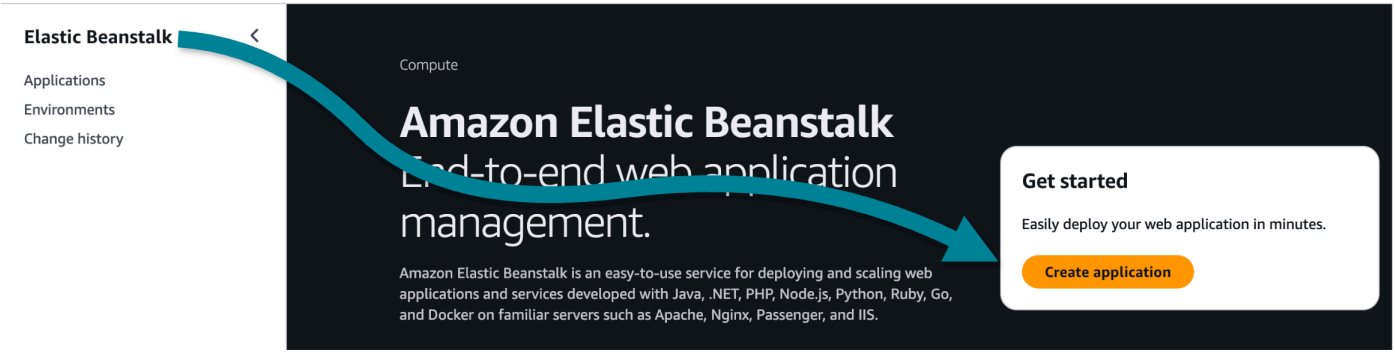
Elastic Beanstalk supporta piattaforme per diversi linguaggi di programmazione, server dell'applicazione e container Docker. Quando crei un ambiente, devi scegliere la piattaforma. È possibile aggiornare la piattaforma, ma non è possibile modificare la piattaforma per un ambiente.

> **Cambio di piattaforma**
>
> Se è necessario modificare i linguaggi di programmazione, è necessario creare e passare a un nuovo ambiente su una piattaforma diversa.

Fase 1 - Creare un'applicazione

Per creare la tua applicazione di esempio, utilizzerai la procedura guidata Create application (Creazione di un'applicazione) della console. Questa consente di creare un'applicazione Elastic Beanstalk e avviare un ambiente al suo interno.

Promemoria: un ambiente è una raccolta di AWS risorse necessarie per eseguire il codice dell'applicazione.



Come creare un’applicazione

1. Apri la console Elastic Beanstalk.
2. Scegli Crea applicazione.
3. Per Application name (Nome applicazione), immetti **`getting-started-app`**.

La console prevede un processo in sei fasi per la creazione di un'applicazione e la configurazione di un ambiente. Per iniziare rapidamente, dovrai concentrarti solo sui primi due passaggi, dopodiché potrai passare alla revisione e alla creazione dell'applicazione e dell'ambiente.

Per configurare un ambiente

1. In Informazioni sull'ambiente, per Nome ambiente immettere:**`gs-app-web-env`**.
2. Per Piattaforma, scegli la piattaforma PHP.
3. Per Codice dell'applicazione e preimpostazioni, accetta le impostazioni predefinite (Applicazione di esempio e Istanza singola), quindi scegli Avanti.

Per configurare l'accesso al servizio

Successivamente, sono necessari due ruoli. Un ruolo di servizio consente a Elastic Beanstalk di monitorare le istanze EC2 e aggiornare la piattaforma dell'ambiente. Un ruolo di profilo di istanza EC2 consente attività come la scrittura di log e l'interazione con altri servizi.

Per creare o selezionare il ruolo Service

1. Se hai già creato un ruolo di servizio e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa Ruolo di servizio e salta il resto di questi passaggi per creare un ruolo di servizio.
2. Se non vedi alcun valore elencato per il ruolo di servizio o desideri crearne uno nuovo, continua con i passaggi successivi.
3. Per Ruolo di servizio, scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Environment.
6. Scegli Next (Successivo).
7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`
8. Scegli Crea ruolo.
9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il ruolo di servizio appena creato.

Per creare o selezionare un profilo di istanza EC2

1. Se hai già creato un profilo di istanza EC2 e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa del profilo dell'istanza EC2 e salta il resto di questi passaggi per creare un profilo di istanza EC2.
2. Se non vedi alcun valore elencato per il profilo dell'istanza EC2 o desideri crearne uno nuovo, continua con i passaggi successivi.
3. Scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Compute.
6. Scegli Next (Successivo).
7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`

- `AWSElasticBeanstalkMulticontainerDocker`

8. Scegli Crea ruolo.
9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il profilo di istanza EC2 appena creato.

Per completare la configurazione e la creazione dell'applicazione

1. Salta la coppia di chiavi EC2.

   Ti mostreremo altri modi per connetterti alle tue istanze Amazon EC2 tramite la console.
2. Scegli Salta alla revisione per passare a diversi passaggi opzionali.

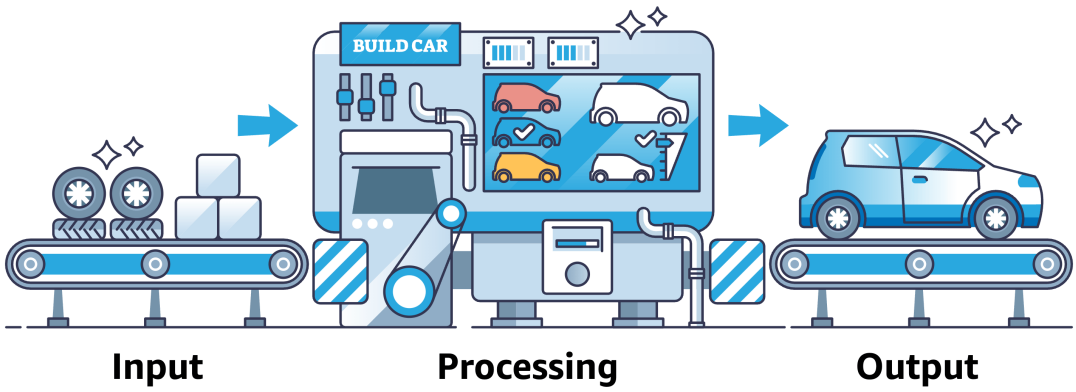
   Passaggi opzionali: rete, database, parametri di scalabilità, configurazione avanzata per aggiornamenti, monitoraggio e registrazione.
3. Nella pagina di revisione che mostra un riepilogo delle tue scelte, scegli Invia.

> **Complimenti!**
>
> Hai creato un'applicazione e configurato un ambiente! Ora devi aspettare che le risorse vengano distribuite.

Fase 2 - Implementa l'applicazione

Quando crei un'applicazione, Elastic Beanstalk configura gli ambienti per te. Devi solo sederti e aspettare.

La creazione delle risorse può richiedere fino a cinque minuti per la distribuzione iniziale. Gli aggiornamenti richiederanno meno tempo perché nello stack verranno implementate solo le modifiche.

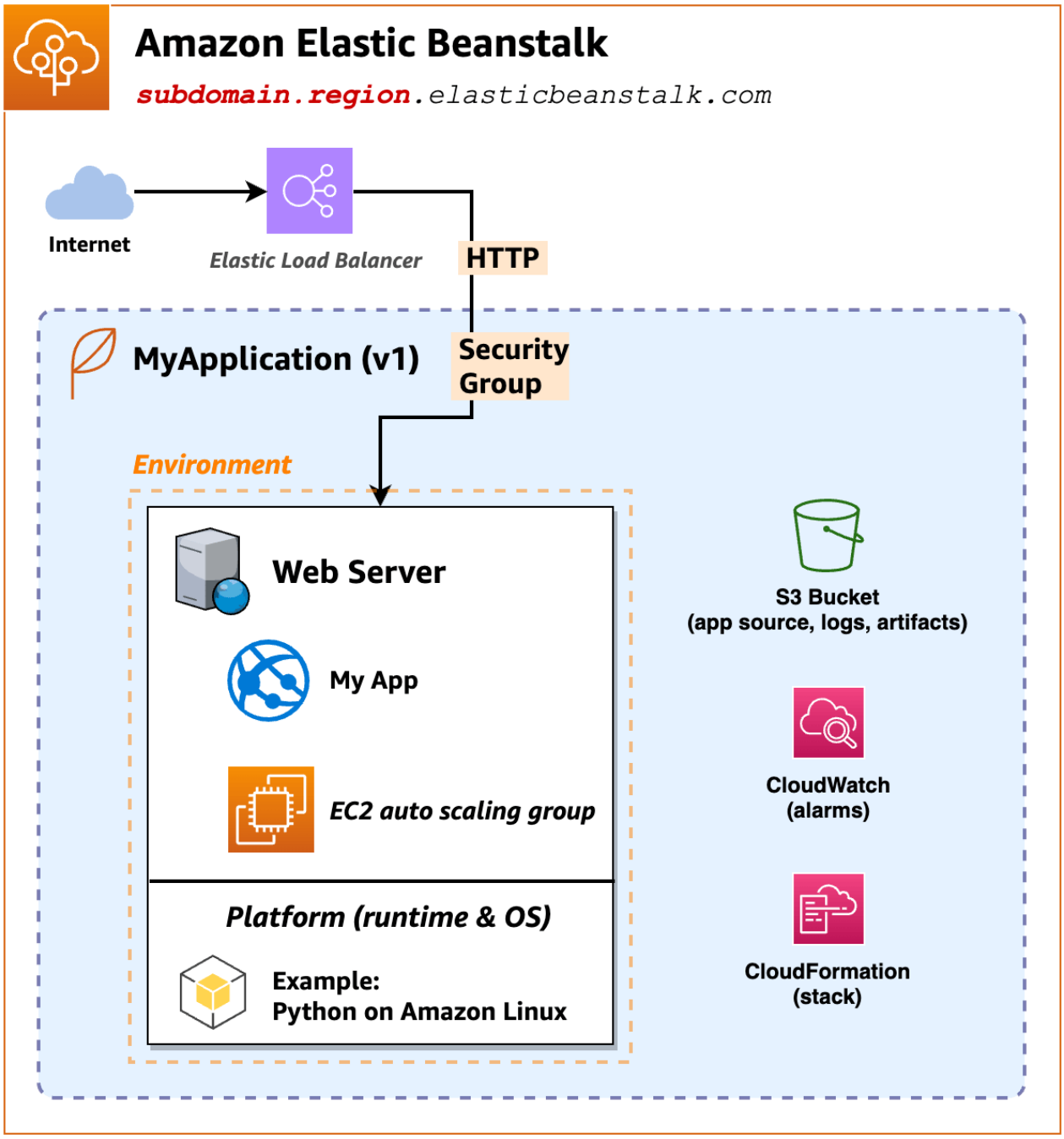


Quando crei l'applicazione di esempio, Elastic Beanstalk crea le seguenti risorse :

- Istanza EC2: una macchina virtuale Amazon EC2 configurata per eseguire app Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un set diverso di software, file di configurazione e script per supportare una versione linguistica, un framework, un contenitore web o una combinazione di questi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso per inoltrare il traffico web alla tua app web, fornire risorse statiche e generare log di accesso ed errori. Puoi connetterti alle tue istanze Amazon EC2 per visualizzare la configurazione e i log.
- Gruppo di sicurezza dell'istanza: verrà creato un gruppo di sicurezza Amazon EC2 per consentire le richieste in entrata sulla porta 80, in modo che il traffico in entrata su un sistema di bilanciamento del carico possa raggiungere la tua app Web.
- Bucket Amazon S3: una posizione di archiviazione per il codice sorgente, i log e altri artefatti.
- CloudWatch Allarmi Amazon: vengono creati due CloudWatch allarmi per monitorare il carico sulle istanze e ridimensionarle verso l'alto o verso il basso secondo necessità.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per distribuire le risorse nell'ambiente e apportare modifiche alla configurazione. È possibile visualizzare il modello di definizione delle risorse nella console.CloudFormation
- Nome di dominio: un nome di dominio che indirizza alla tua app web nel formato:`subdomain.region.elasticbeanstalk.com`.

Elastic Beanstalk crea l'applicazione, avvia un ambiente, crea una versione dell'applicazione, quindi distribuisce il codice nell'ambiente. Durante il processo, la console tiene traccia dei progressi e visualizza lo stato degli eventi nella scheda Eventi.



Dopo aver distribuito tutte le risorse, lo stato dell'ambiente dovrebbe cambiare. **0k**

> La tua applicazione è pronta!
>
> Dopo aver modificato lo stato di integrità dell'applicazione**0k**, puoi accedere al sito Web dell'applicazione Web.

Fase 3 - Esplora l'ambiente Elastic Beanstalk

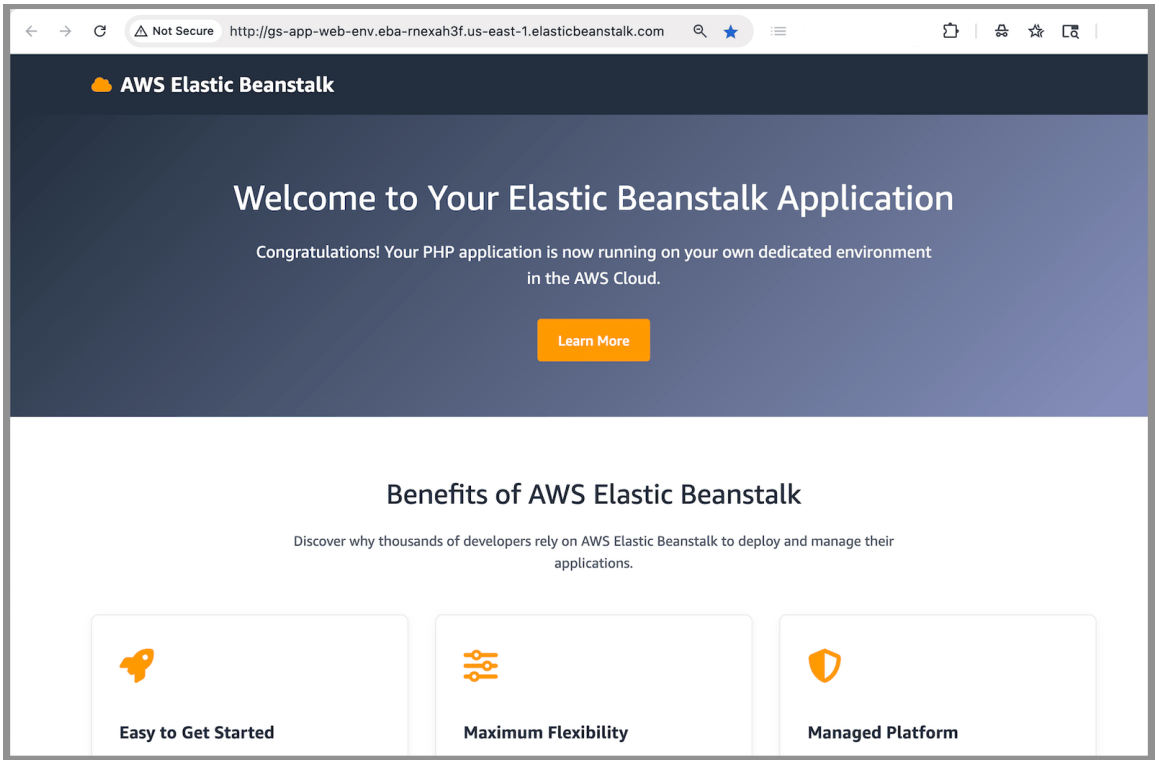
Inizierai a esplorare l'ambiente applicativo distribuito dalla pagina Panoramica dell'ambiente nella console.

Per visualizzare l'ambiente e l'applicazione

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegli Vai all'ambiente per sfogliare la tua applicazione!

   (Puoi anche scegliere il link URL elencato per Domain per sfogliare l'applicazione.)

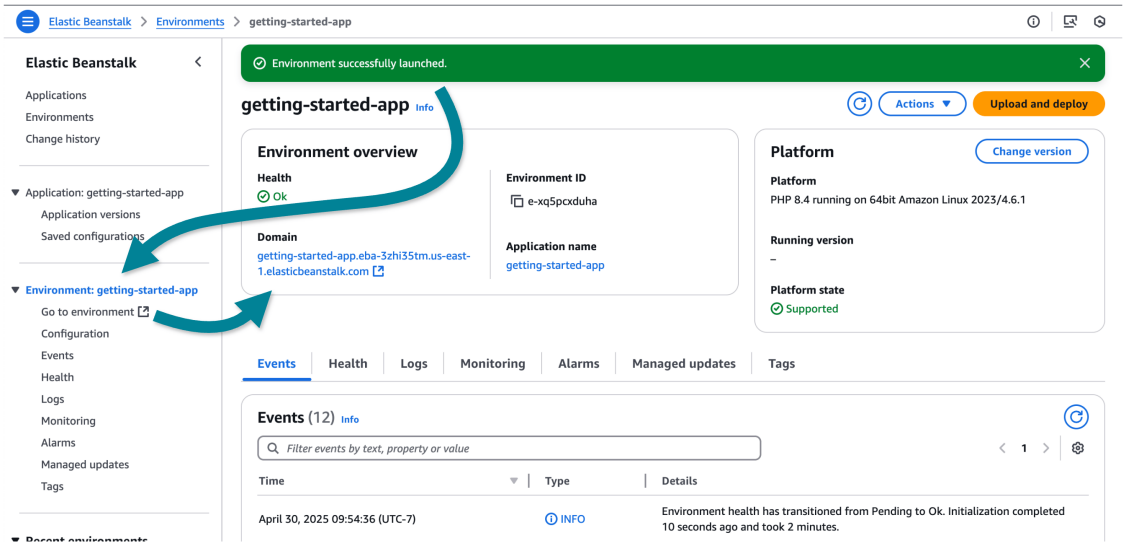
   La connessione sarà HTTP (non HTTPS), quindi potresti visualizzare un avviso nel tuo browser.



Tornando alla console Elastic Beanstalk, la parte superiore mostra la panoramica dell'ambiente con informazioni di primo livello sull'ambiente, tra cui nome, URL del dominio, stato di salute attuale, versione in esecuzione e piattaforma su cui è in esecuzione l'applicazione. La versione

e la piattaforma in esecuzione sono essenziali per la risoluzione dei problemi dell'applicazione attualmente distribuita.

Dopo il riquadro di panoramica, vedrai l'attività ambientale recente nella scheda Eventi.



Mentre Elastic Beanstalk AWS crea le risorse e avvia l'applicazione, l'ambiente è in uno stato. `Pending` I messaggi di stato relativi agli eventi di lancio vengono aggiunti continuamente all'elenco degli eventi.

Il dominio dell'ambiente è l'URL dell'applicazione Web distribuita. Nel riquadro di navigazione a sinistra, vai all'ambiente ti porta anche al tuo dominio. Allo stesso modo, il riquadro di navigazione a sinistra contiene collegamenti che corrispondono alle varie schede.

Prendi nota del link Configurazione nel riquadro di navigazione a sinistra, che mostra un riepilogo dei valori delle opzioni di configurazione dell'ambiente, raggruppati per categoria.

> **Impostazioni di configurazione dell'ambiente**
>
> Prendi nota del link Configurazione nel riquadro di navigazione a sinistra. Puoi visualizzare e modificare impostazioni dettagliate dell'ambiente, come ruoli di servizio, rete, database, scalabilità, aggiornamenti della piattaforma gestita, memoria, monitoraggio dello stato, distribuzione continua, registrazione e altro ancora!

Le varie schede contengono informazioni dettagliate sull'ambiente:

- Eventi: visualizza un elenco aggiornato di informazioni e messaggi di errore dal servizio Elastic Beanstalk e da altri servizi per le risorse del tuo ambiente.
- Health: visualizza lo stato e le informazioni dettagliate sullo stato delle istanze Amazon EC2 che eseguono la tua applicazione.
- Logs (Log): recupera e scarica i log da Amazon EC2 nel tuo ambiente. Puoi recuperare i log completi o le attività recenti. I log recuperati saranno disponibili per 15 minuti.
- Monitoraggio: visualizza le statistiche relative all'ambiente, come la latenza media e l'utilizzo della CPU.
- Allarmi: visualizza e modifica gli allarmi configurati per le metriche ambientali.
- Aggiornamenti gestiti: visualizza le informazioni sugli aggiornamenti imminenti e completati della piattaforma gestita e sulla sostituzione delle istanze.
- Tag: visualizza e modifica le coppie chiave-valore applicate al tuo ambiente.

> **Note**
>
> I collegamenti nel riquadro di navigazione della console mostreranno la scheda corrispondente.

Risoluzione dei problemi con i log

Per risolvere i comportamenti imprevisti o eseguire il debug delle distribuzioni, potresti voler controllare i log nei tuoi ambienti.

Puoi richiedere 100 righe di tutti i file di registro nella scheda Logs della console Elastic Beanstalk. In alternativa, puoi connetterti direttamente all'istanza Amazon EC2 e tracciare i log in tempo reale.

Per richiedere i log (console Elastic Beanstalk)

1. Accedi al tuo ambiente nella console Elastic Beanstalk.
2. Scegli la scheda Registri o il menu di navigazione a sinistra, quindi scegli Request logs.
3. Seleziona Ultime 100 righe.
4. Dopo aver creato i registri, scegli il link Download per visualizzarli nel browser.

Nei log, trova il log e annota la directory per il log di accesso nginx.

Aggiungi una policy per abilitare le connessioni ad Amazon EC2

Prima di poterti connettere, devi aggiungere una policy che abiliti le connessioni ad Amazon EC2 con Session Manager.

1. Passare alla IAM console (Console IAM).
2. Trova e seleziona il **`aws-elasticbeanstalk-ec2-role`** ruolo.
3. Scegli Aggiungi autorizzazione, quindi Allega politiche.
4. Cerca una politica predefinita che inizi con il seguente testo:**`AmazonSSMManagedEC2Instance`**, quindi aggiungila al ruolo.

Per connetterti ad Amazon EC2 con Session Manager

1. Passa alla console Amazon EC2.
2. Scegli Istanze, quindi seleziona la tua `gs-app-web-env` istanza.
3. Scegli Connect, quindi Session Manager.
4. Scegli Connetti.

Dopo esserti connesso all'istanza, avvia una shell bash e segui i log:

1. Esegui il comando **`bash`**.
2. Esegui il comando **`cd /var/log/nginx`**.
3. Esegui il comando **`tail -f access.log`**.
4. Nel tuo browser, vai all'URL del dominio dell'applicazione. Aggiorna.

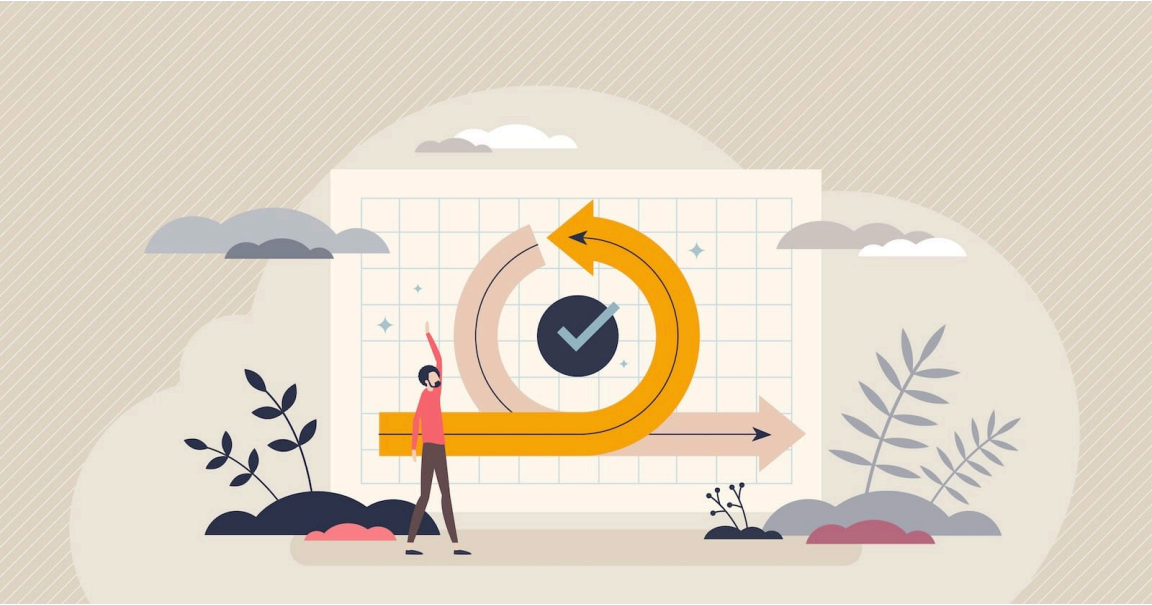
> Congratulazioni, siete connessi!
>
> Dovresti vedere le voci di registro dell'istanza aggiornate ogni volta che aggiorni la pagina.

> Il pulsante Connect non funziona?
>
> Se il pulsante di connessione non è disponibile, torna a IAM e verifica di aver aggiunto la policy necessaria al ruolo.

Fase 4 - Aggiorna l'applicazione

Alla fine, vorrai aggiornare la tua applicazione. È possibile distribuire una nuova versione dell'applicazione in qualsiasi momento, finché nell'ambiente non sono in corso operazioni di aggiornamento.

La versione dell'applicazione con cui hai avviato questo tutorial si chiama Sample Application (Applicazione di esempio).



Per aggiornare la versione dell'applicazione

1. Scarica la seguente applicazione PHP di esempio:

   PHP — php-v2.zip
2. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
3. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
4. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
5. Seleziona Scegli file, quindi carica il pacchetto sorgente dell'applicazione di esempio che hai scaricato.

   La console compila automaticamente l'etichetta Version con una nuova etichetta univoca, incrementando automaticamente un numero intero finale. Se scegli la tua etichetta di versione, assicurati che sia unica.

6. Seleziona Implementa.

Mentre Elastic Beanstalk distribuisce il tuo file sulle tue istanze Amazon EC2, puoi visualizzare lo stato della distribuzione nella pagina di panoramica dell'ambiente. Mentre la versione dell'applicazione viene aggiornata, lo stato di Health dell'ambiente è grigio. Al termine della distribuzione, Elastic Beanstalk esegue un controllo dello stato dell'applicazione. Quando l'applicazione risponde al controllo dello stato, viene considerata integra e lo stato ritorna verde. La panoramica dell'ambiente mostra la nuova Running Version (Versione in esecuzione): il nome fornito come Version label (Etichetta versione).

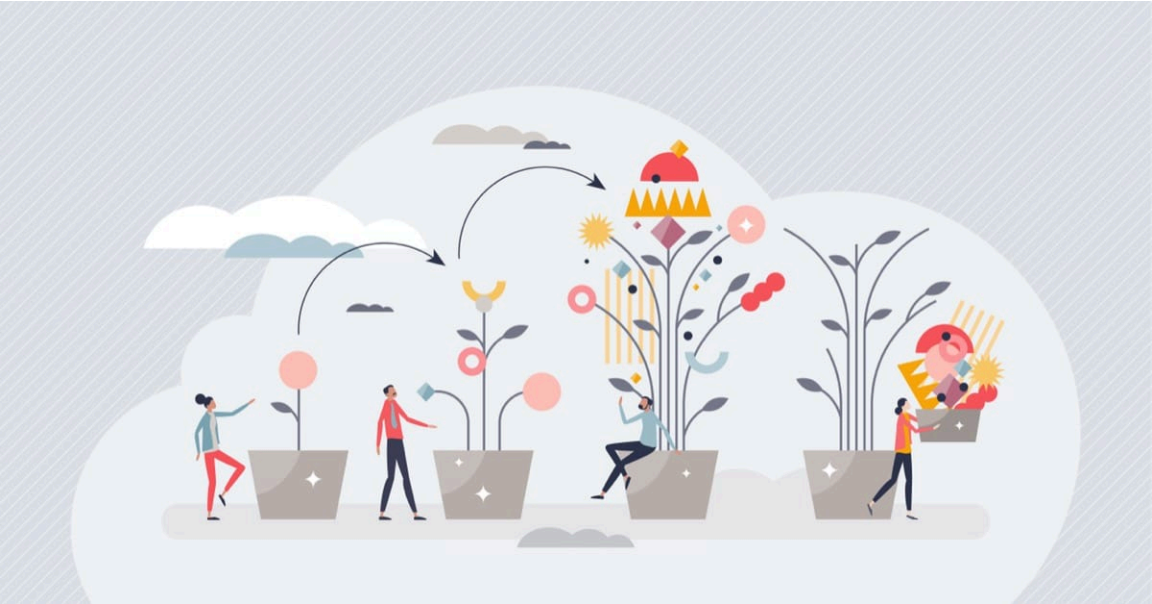
Inoltre, Elastic Beanstalk carica la nuova versione dell'applicazione e la aggiunge alla tabella delle versioni dell'applicazione. Per visualizzare la tabella, scegliere Application versions (Versioni applicazione) in getting-started-app nel riquadro di navigazione.

> **Aggiornamento riuscito!**
>
> Dovresti vedere un messaggio «v2" aggiornato dopo aver aggiornato il browser.
> Se desideri modificare tu stesso il codice sorgente, decomprimi, modifica e poi ricomprimi il pacchetto sorgente. Su macOS, usa il seguente comando dall'interno della tua directory php con -X per escludere attributi di file aggiuntivi:
> `zip -X -r ../php-v2.zip .`

Fase 5 - Ridimensiona la tua applicazione

È possibile personalizzare l'ambiente per renderlo più appropriato all'applicazione. Ad esempio, in presenza di un'applicazione con elevati requisiti di elaborazione, è possibile modificare il tipo di istanza Amazon Elastic Compute Cloud (Amazon EC2) che esegue l'applicazione. Per applicare le modifiche di configurazione, Elastic Beanstalk esegue un aggiornamento dell'ambiente.

Alcune modifiche di configurazione sono semplici e possono essere eseguite rapidamente. Alcune modifiche richiedono l'eliminazione e la ricreazione AWS delle risorse, operazione che può richiedere diversi minuti. Durante la modifica delle impostazioni di configurazione, Elastic Beanstalk avvisa dei potenziali tempi di inattività dell'applicazione.



Aumentare le impostazioni di capacità

In questo esempio di modifica della configurazione, puoi modificare le impostazioni di capacità dell'ambiente. Verrà configurato un ambiente scalabile con bilanciamento del carico che dispone di un numero di istanze Amazon EC2 da due a quattro nel proprio gruppo Auto Scaling, controllando infine che la modifica sia stata eseguita correttamente. Elastic Beanstalk crea un'ulteriore istanza Amazon EC2, aggiungendola alla singola istanza creata inizialmente. Quindi, Elastic Beanstalk associa entrambe le istanze al load balancer dell'ambiente. Di conseguenza, la reattività dell'applicazione è migliorata e la sua disponibilità aumenta.

Per modificare la capacità dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Instance traffic and scaling (Traffico e dimensionamento delle istanze), scegli Edit (Modifica).
5. Comprimi la sezione Instances (Istanze), in modo da poter visualizzare più facilmente la sezione Capacity (Capacità). Nella sezione Auto Scaling group (Gruppo con scalabilità automatica), modifica Environment type (Tipo di ambiente) in Load balanced (Con bilanciamento del carico).
6. Nella riga Istanze, modifica Min in e Max in**2**. **4**

7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

   Se vieni avvisato che l'aggiornamento sostituirà tutte le istanze correnti. Scegli Conferma.

L'aggiornamento dell'ambiente potrebbe richiedere alcuni minuti. Dovresti vedere diversi aggiornamenti nell'elenco degli eventi. Tieni d'occhio l'evento Nuova configurazione implementata con successo nell'ambiente.

Verifica l'aumento della capacità

Una volta completato l'aggiornamento dell'ambiente e dopo che l'ambiente è pronto, Elastic Beanstalk ha avviato automaticamente una seconda istanza per soddisfare la nuova impostazione di capacità minima.

Per verificare l'aumento della capacità

1. Scegli Health dall'elenco delle schede o dal riquadro di navigazione a sinistra.
2. Consulta la sezione Integrità migliorata delle istanze.

> Hai appena ampliato la tua gamma!
>
> Con due istanze Amazon EC2, la capacità dell'ambiente è raddoppiata e ci sono voluti solo pochi minuti.

Pulizia dell'ambiente Elastic Beanstalk

Per assicurarti che non ti vengano addebitati costi per i servizi che non utilizzi, elimina tutte le versioni dell'applicazione e chiudi gli ambienti, eliminando anche le AWS risorse che l'ambiente ha creato per te.

Per eliminare l'applicazione e tutte le risorse associate

1. Elimina tutte le versioni dell'applicazione.
   a. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
   b. Nel riquadro di navigazione, scegliere Applications (Applicazioni), quindi selezionare getting-started-app.
   c. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Application versions (Versioni applicazione).
   d. Nella pagina Application Versions (Versioni applicazione) selezionare tutte le versioni dell'applicazione da eliminare.
   e. Scegli Azioni, quindi Elimina.
   f. Attivare Delete versions from Amazon S3 (Elimina versioni da Amazon S3).
   g. Scegliere Delete (Elimina), quindi scegliere Done (Fatto).
2. Terminare l'ambiente.
   a. Nel riquadro di navigazione, scegli getting-started-app, quindi scegli nell'elenco degli ambienti. GettingStartedApp-env
   b. Seleziona Actions (Operazioni), quindi Terminate Environment (Termina ambiente).

   c. Conferma di voler terminare GettingStartedApp-envdigitando il nome dell'ambiente, quindi scegli Termina.

3. Eliminare l'applicazione getting-started-app.

   a. Nel riquadro di navigazione, scegliere getting-started-app.
   b. Scegliere Actions (Operazioni), quindi Delete application (Elimina applicazione).
   c. Confermare l'eliminazione di getting-started-app digitando il nome dell'applicazione, quindi scegliere Delete (Elimina).

> **Complimenti!**
>
> Hai distribuito con successo un'applicazione di esempio nel AWS Cloud, caricato una nuova versione, modificato la sua configurazione per aggiungere una seconda istanza di Auto Scaling e AWS ripulito le tue risorse!

Fasi successive

Per scoprire come utilizzare lo strumento da riga di **eb** comando per automatizzare la distribuzione del codice su Elastic Beanstalk, ti suggeriamo di continuare con. the section called “QuickStart per PHP”

Successivamente, potresti voler esaminare come configurare la connessione HTTPS, vedi. the section called “HTTPS”

Configurazione dell'interfaccia a riga di comando EB (EB CLI) per gestire Elastic Beanstalk

L'EB CLI è un'interfaccia a riga di comando che fornisce comandi interattivi per creare, aggiornare e monitorare ambienti. AWS Elastic Beanstalk Il progetto open source EB CLI è su Github: `aws/aws-elastic-beanstalk-cli`

Dopo aver installato l'interfaccia a riga di comando EB e aver configurato una directory di progetto, puoi creare ambienti con un singolo comando:

```
$ eb create my-beanstalk-environment
```

Ti consigliamo di installarlo con lo script di configurazione, scopri come farlo. the section called "Installa EB CLI"

AWS CLI Fornisce accesso diretto a Elastic APIs Beanstalk di basso livello. Sebbene potente, è anche prolisso e meno preferito rispetto alla CLI EB. Ad esempio, la creazione di un ambiente con AWS CLI richiede la seguente serie di comandi:

```
$ aws elasticbeanstalk check-dns-availability \
    --cname-prefix my-cname
$ aws elasticbeanstalk create-application-version \
    --application-name my-application \
    --version-label v1 \
    --source-bundle S3Bucket=amzn-s3-demo-bucket,S3Key=php-proxy-sample.zip
$ aws elasticbeanstalk create-environment \
    --cname-prefix my-cname \
    --application-name my-app \
    --version-label v1 \
    --environment-name my-env \
    --solution-stack-name "64bit Amazon Linux 2023 v4.5.0 running Ruby 3.4"
```

Installa EB CLI con lo script di configurazione (consigliato)

> **Consigliamo lo script di installazione**
>
> Si consiglia di utilizzare lo script di installazione per configurare l'EB CLI e le sue dipendenze e prevenire potenziali conflitti con altri pacchetti Python.

Prerequisiti: Git, Python e virtualenv

Per scaricare e utilizzare lo script di installazione

1. Clonare il repository.

```
git clone https://github.com/aws/aws-elastic-beanstalk-cli-setup.git
```

2. Installa o aggiorna l'EB CLI.

   macOS/Linux in Bash o Zsh

```
python ./aws-elastic-beanstalk-cli-setup/scripts/ebcli_installer.py
```

   Windows nella nostra finestra di comando PowerShell

```
python .\aws-elastic-beanstalk-cli-setup\scripts\ebcli_installer.py
```

3. Verifica che l'interfaccia a riga di comando EB sia installata correttamente.

```
$ eb --version
EB CLI 3.21.0 (Python 3.12)
```

Per istruzioni complete sull'installazione, consulta il `aws/aws-elastic-beanstalk-cli-setup`repository su GitHub.

Installazione manuale dell'interfaccia a riga di comando EB

Puoi installare l'EB CLI su Linux, macOS e Windows con il `pip` gestore di pacchetti per Python che fornisce l'installazione, l'aggiornamento e la rimozione dei pacchetti Python e delle loro dipendenze.

> **Consigliamo lo script di installazione**
>
> Si consiglia di utilizzare il Installa EB CLI per configurare l'EB CLI e prevenire i conflitti di dipendenza.

Prerequisito: è necessario che sia installata una versione supportata di Python. Puoi scaricarlo dalla pagina dei download di Python sul sito Web di Python.

Per installare l'EB CLI (manualmente)

1. Eseguire il seguente comando seguente.

```
$ pip install awsebcli --upgrade --user
```

   L'opzione `--upgrade` indica a `pip` di aggiornare i requisiti già installati. L'opzione `--user` indica a `pip` di installare il programma in una sottodirectory della directory utente per evitare di modificare le librerie utilizzate dal sistema operativo.

> **Risoluzione dei problemi**
>
> Se si verificano problemi quando si tenta di installare l'interfaccia a riga di comando EB con `pip`, è possibile installare l'interfaccia a riga di comando EB in un ambiente virtuale per isolare lo strumento e le relative dipendenze oppure utilizzare una versione di Python diversa da quella utilizzata normalmente.

2. Aggiungi il percorso dell'eseguibile alla variabile `PATH`:
   - Su Linux e macOS:

     Linux – `~/.local/bin`

     macOS – `~/Library/Python/3.12/bin`

     Per modificare la variabile `PATH` (Linux, Unix o macOS):

     a. Trova lo script del profilo della tua shell nella tua cartella utente. Se non sei sicuro di quale sia la tua shell, esegui `echo $SHELL`.

```
$ ls -a ~
```

```
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash – `.bash_profile`, `.profile` o `.bash_login`.
- Zsh - `.zshrc`
- Tcsh – `.tcshrc`, `.cshrc` o `.login`.

b. Aggiungi un comando di esportazione allo script del tuo profilo. L'esempio seguente aggiunge il percorso rappresentato da *`LOCAL_PATH`* alla `PATH` variabile corrente.

```
export PATH=LOCAL_PATH:$PATH
```

c. Carica lo script del profilo descritto nella prima fase nella sessione corrente. L'esempio seguente carica lo script di profilo rappresentato da*`PROFILE_SCRIPT`*.

```
$ source ~/PROFILE_SCRIPT
```

- In Windows:

  Python 3.12 — `%USERPROFILE%\AppData\Roaming\Python\Python312\Scripts`

  Versioni precedenti di Python – `%USERPROFILE%\AppData\Roaming\Python\Scripts`

  Per modificare la variabile `PATH` (Windows):

  a. Premi il tasto Windows, quindi digita **`environment variables`**.

  b. Seleziona Edit environment variables for your account (Modifica variabili di ambiente per l'account).

  c. Seleziona PATH (PERCORSO), quindi Edit (Modifica).

  d. Aggiungi percorsi nel campo Variable value (Valore variabile), separati da punto e virgola. Ad esempio: *`C:\item1\path`***`;C:\item2\path`**

  e. Fai doppio clic su OK per applicare le nuove impostazioni.

  f. Chiudi tutte le finestre del prompt dei comandi in esecuzione, quindi riapri una finestra del prompt dei comandi.

3. Verifica che l'interfaccia a riga di comando EB sia installata correttamente eseguendo eb --version.

```
$ eb --version
EB CLI 3.21.0 (Python 3.12)
```

L'interfaccia a riga di comando EB viene aggiornata regolarmente per aggiungere funzionalità che supportano le caratteristiche più recenti di Elastic Beanstalk. Per eseguire l'aggiornamento alla versione più recente dell'interfaccia a riga di comando EB, esegui nuovamente il comando di installazione.

```
$ pip install awsebcli --upgrade --user
```

Se devi disinstallare l'interfaccia a riga di comando EB, usa `pip uninstall`.

```
$ pip uninstall awsebcli
```

Installazione dell'interfaccia a riga di comando EB in un ambiente virtuale

È possibile evitare conflitti di versione dei requisiti con altri pacchetti `pip` installando l'interfaccia a riga di comando EB in un ambiente virtuale.

Per installare l'interfaccia a riga di comando EB in un ambiente virtuale

1. Innanzitutto, installa con. `virtualenv` `pip`

   ```
   $ pip install --user virtualenv
   ```

2. Crea un ambiente virtuale.

   ```
   $ virtualenv ~/eb-ve
   ```

   Utilizza l'opzione `-p` per usare un eseguibile Python diverso da quello predefinito.

   ```
   $ virtualenv -p /usr/bin/python3.12 ~/eb-ve
   ```

3. Attiva l'ambiente virtuale.

   Linux, Unix o macOS

   ```
   $ source ~/eb-ve/bin/activate
   ```

   Windows

```
$ %USERPROFILE%\eb-ve\Scripts\activate
```

4. Installa l'interfaccia a riga di comando EB.

```
(eb-ve)$ pip install awsebcli --upgrade
```

5. Verifica che l'interfaccia a riga di comando EB sia installata correttamente.

```
$ eb --version
EB CLI 3.23.0 (Python 3.12)
```

Puoi utilizzare il comando `deactivate` per chiudere l'ambiente virtuale. Ogni volta che avvii una nuova sessione, esegui di nuovo il comando di attivazione.

Per eseguire l'upgrade alla versione più recente, esegui nuovamente il comando di installazione.

```
(eb-ve)$ pip install awsebcli --upgrade
```

Installazione dell'interfaccia a riga di comando EB con Homebrew

La versione più recente dell'interfaccia a riga di comando EB è generalmente disponibile da `Homebrew` un paio di giorni dopo che appare in `pip`.

> **Consigliamo lo script di installazione**
>
> Si consiglia di utilizzare il Installa EB CLI per configurare l'EB CLI e prevenire i conflitti di dipendenza.

Per installare l'interfaccia a riga di comando EB con **`Homebrew`**

1. Accertati di disporre della versione più recente di `Homebrew`.

```
$ brew update
```

2. Esegui `brew install awsebcli`.

```
$ brew install awsebcli
```

3. Verifica che l'interfaccia a riga di comando EB sia installata correttamente.

```
$ eb --version
EB CLI 3.21.0 (Python 3.12)
```

Configurazione dell'interfaccia a riga di comando EB

Dopo aver installato l'EB CLI, siete pronti per configurare la directory del progetto e l'EB CLI eseguendolaeb init. L'esempio seguente mostra i passaggi di configurazione quando si esegue eb init per la prima volta in una cartella di progetto denominata. eb

Per inizializzare un progetto dell'interfaccia a riga di comando EB

1. In primo luogo, l'interfaccia a riga di comando EB ti chiede di selezionare una regione. Scegli la tua regione preferita.

```
~/eb $ eb init
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3
```

2. Se richiesto, fornisci la tua chiave di accesso e la chiave segreta in modo che l'EB CLI possa gestire le risorse per te. Le chiavi di accesso vengono create nella AWS Identity and Access Management console. Se non disponi delle chiavi, consulta l'argomento che descrive come ottenere le credenziali di sicurezza in Riferimenti generali di Amazon Web Services.

```
You have not yet set up your credentials or your credentials are incorrect.
You must provide your credentials.
(aws-access-id): AKIAJOUAASEXAMPLE
(aws-secret-key): 5ZRIrtTM4ciIAvd4EXAMPLEDtm+PiPSzpoK
```

3. Un'applicazione Elastic Beanstalk è una risorsa che contiene un set di versioni dell'applicazione (origine), ambienti e configurazioni salvate associati a una singola applicazione Web. Ogni volta

che distribuisci il codice sorgente per Elastic Beanstalk utilizzando la CLI EB, una nuova versione dell'applicazione viene creata e aggiunta all'elenco.

```
Select an application to use
1) [ Create new Application ]
(default is 1): 1
```

4. Il nome predefinito dell'applicazione è il nome della cartella in cui esegui eb init. Inserisci un nome che descrive il tuo progetto.

```
Enter Application Name
(default is "eb"): eb
Application eb has been created.
```

5. Seleziona una piattaforma che corrisponda al framework o al linguaggio con cui è stata sviluppata la tua applicazione Web. Se non hai iniziato a sviluppare un'applicazione, scegli una piattaforma che ti interessa. Visualizzerai come avviare un'applicazione di esempio a breve, e puoi sempre modificare questa impostazione in un secondo momento.

```
Select a platform.
1) .NET Core on Linux
2) .NET on Windows Server
3) Docker
4) Go
5) Java
6) Node.js
7) PHP         <== select platform by number
8) Packer
9) Python
10) Ruby
11) Tomcat
(make a selection):7
```

6. Seleziona un ramo specifico della piattaforma.

```
Select a platform branch.
1) PHP 8.4 running on 64bit Amazon Linux 2023
2) PHP 8.3 running on 64bit Amazon Linux 2023
3) PHP 8.2 running on 64bit Amazon Linux 2023
4) PHP 8.1 running on 64bit Amazon Linux 2023
5) PHP 8.1 running on 64bit Amazon Linux 2
```

```
(default is 1):1
```

7. Scegli yes (sì) per assegnare una coppia di chiavi SSH alle istanze nel tuo ambiente Elastic Beanstalk. In questo modo ci si connette direttamente per la risoluzione dei problemi.

```
Do you want to set up SSH for your instances?
(y/n): y
```

8. Seleziona una coppia di chiavi esistente o creane una nuova. Per utilizzare eb init per creare una nuova coppia di chiavi, ssh-keygen deve essere installato nel computer locale e disponibile dalla riga di comando. L'EB CLI registra EC2 per te la nuova coppia di chiavi con Amazon e archivia la chiave privata localmente in una cartella denominata `.ssh` nella tua directory utente.

```
Select a keypair.
1) [ Create new KeyPair ]
(default is 1): 1
```

L'installazione dell'interfaccia a riga di comando EB è ora configurata e pronta all'uso.

Configurazione avanzata

- File ignorati con .ebignore
- Utilizzo di profili denominati
- Distribuzione di artifact invece della cartella del progetto
- Impostazioni di configurazione e precedenza
- Metadati delle istanze

File ignorati con .ebignore

Puoi stabilire che l'interfaccia a riga di comando EB ignori alcuni file nella directory di progetto aggiungendo il file `.ebignore` alla directory. Questo file funziona come un file `.gitignore`. Quando distribuisci la directory del progetto in Elastic Beanstalk e crei una nuova versione dell'applicazione, la CLI EB non include i file specificati da `.ebignore` nel bundle di origine creato.

Se `.ebignore` non è presente ma è disponibile `.gitignore`, l'interfaccia a riga di comando EB ignora i file specificati in `.gitignore`. Se `.ebignore` è presente, l'interfaccia a riga di comando EB non legge `.gitignore`.

Quando `.ebignore` è presente, l'interfaccia a riga di comando EB non utilizza i comandi git per creare il bundle di origine. Questo significa che la CLI EB ignora i file specificati in `.ebignore` e include tutti gli altri file. In particolare, include i file sorgente non eseguiti.

> **Note**
>
> In Windows, se aggiungi `.ebignore`, l'interfaccia a riga di comando EB segue i collegamenti simbolici e include il file collegato durante la creazione di un bundle di origine. Si tratta di un problema noto e verrà risolto in un aggiornamento futuro.

Utilizzo di profili denominati

Se archivi le credenziali come un profilo denominato in un file `credentials` o `config`, puoi utilizzare l'opzione `--profile` per specificare un profilo in modo esplicito. Ad esempio, il comando seguente crea una nuova applicazione utilizzando il profilo `user2`.

```
$ eb init --profile user2
```

Puoi inoltre modificare il profilo predefinito impostando la variabile di ambiente `AWS_EB_PROFILE`. Quando questa variabile è impostata, l'interfaccia a riga di comando EB legge le credenziali del profilo specificato anziché `default` o eb-cli.

Linux, macOS o Unix

```
$ export AWS_EB_PROFILE=user2
```

Windows

```
> set AWS_EB_PROFILE=user2
```

Distribuzione di artifact invece della cartella del progetto

Puoi chiedere all'interfaccia a riga di comando EB di distribuire un file WAR o ZIP generato come parte di un processo di creazione separato aggiungendo le seguenti righe `.elasticbeanstalk/config.yml` alla cartella del tuo progetto.

```
deploy:
```

```
  artifact: path/to/buildartifact.zip
```

Se configuri l'interfaccia a riga di comando EB nel tuo archivio Git e non esegui il commit dell'artefatto all'origine, utilizza l'opzione `--staged` per distribuire la compilazione più recente.

```
~/eb$ eb deploy --staged
```

Impostazioni di configurazione e precedenza

L'EB CLI utilizza una catena di provider per cercare AWS le credenziali in diversi punti, tra cui variabili di ambiente di sistema o utente e file di configurazione locali AWS .

L'interfaccia a riga di comando EB cerca le credenziali e le impostazioni di configurazione nell'ordine seguente:

1. Opzioni della riga di comando: specifica un profilo denominato utilizzando `--profile` per sovrascrivere le impostazioni predefinite.
2. Variabili di ambiente: `AWS_ACCESS_KEY_ID` e `AWS_SECRET_ACCESS_KEY`.
3. Il file AWS delle credenziali: si trova `~/.aws/credentials` nei sistemi Linux e OS X o nei sistemi Windows. `C:\Users\`*`USERNAME`*`\.aws\credentials` Questo file può contenere più profili denominati in aggiunta a un profilo predefinito.
4. File di configurazione della AWS CLI: disponibile nei `~/.aws/config` sistemi Linux e OS X o nei `C:\Users\`*`USERNAME`*`\.aws\config` sistemi Windows. Questo file può contenere un profilo predefinito, profili denominati e parametri di configurazione AWS CLI specifici per ciascuno di essi.
5. File di configurazione legacy della CLI EB: disponibile in `~/.elasticbeanstalk/config` nei sistemi Linux e OS X o in `C:\Users\`*`USERNAME`*`\.elasticbeanstalk\config` nei sistemi Windows.
6. Credenziali del profilo di istanza: queste credenziali possono essere utilizzate su EC2 istanze Amazon con un ruolo di istanza assegnato e vengono fornite tramite il servizio di metadati Amazon EC2 . Il profilo dell'istanza deve disporre delle autorizzazioni per utilizzare Elastic Beanstalk.

Se il file delle credenziali contiene un profilo denominato con il nome "eb-cli", l'interfaccia a riga di comando EB preferirà quel profilo al profilo predefinito. Se non vengono trovati profili o viene trovato un profilo che non dispone delle autorizzazioni per utilizzare Elastic Beanstalk, la CLI EB richiede di immettere le chiavi.

Metadati delle istanze

Per utilizzare l'EB CLI da un' EC2 istanza Amazon, crea un ruolo che abbia accesso alle risorse necessarie e assegna quel ruolo all'istanza quando viene lanciata. Avvia l'istanza e installa l'interfaccia a riga di comando di EB utilizzando `pip`.

```
~$ sudo pip install awsebcli
```

`pip` è preinstallato su Amazon Linux.

L'interfaccia a riga di comando di EB legge le credenziali dai metadata dell'istanza. Per ulteriori informazioni, consulta Concedere alle applicazioni eseguite su Amazon EC2 Instances l'accesso alle AWS risorse nella IAM User Guide.

Utilizzo dell'interfaccia a riga di comando EB con Git

L'EB CLI fornisce un'integrazione Git che consente di associare rami di codice a ambienti specifici in Elastic Beanstalk per organizzare le distribuzioni.

Per installare Git e inizializzare il tuo repository Git

1. Scarica la versione più recente di Git dal sito http://git-scm.com.
2. Inizializza il tuo repository Git digitando quanto segue:

   ```
   ~/eb$ git init
   ```

   Ora l'interfaccia a riga di comando EB riconoscerà che la tua applicazione è configurata con Git.
3. Se non hai già eseguito eb init, fallo adesso:

   ```
   ~/eb$ eb init
   ```

Associazione di ambienti Elastic Beanstalk ai rami Git

Puoi associare un ambiente diverso a ogni ramo del codice. Quando esegui il checkout di un ramo, le modifiche vengono distribuite all'ambiente associato. Ad esempio, puoi digitare quanto segue per associare il tuo ambiente di produzione al ramo principale e un ambiente di sviluppo separato al ramo di sviluppo:

```
~/eb$ git checkout mainline
~/eb$ eb use prod
~/eb$ git checkout develop
~/eb$ eb use dev
```

Distribuzione di modifiche

Per impostazione predefinita, l'interfaccia a riga di comando EB distribuisce il commit più recente nel ramo attuale, utilizzando l'ID di commit e il messaggio rispettivamente come etichetta di versione dell'applicazione e descrizione. Se desideri eseguire la distribuzione nel tuo ambiente senza impegnarti, puoi utilizzare l'opzione `--staged` per distribuire le modifiche aggiunte all'area di gestione temporanea.

Per distribuire le modifiche senza eseguire il commit

1. Aggiungi file nuovi e modificati all'area di gestione temporanea:

   ```
   ~/eb$ git add .
   ```

2. Distribuisci le modifiche per fasi con eb deploy:

   ```
   ~/eb$ eb deploy --staged
   ```

Se hai configurato l'interfaccia a riga di comando EB per distribuire un artefatto e non esegui il commit dell'artefatto al tuo repository git, utilizza l'opzione `--staged` per distribuire la build più recente.

Utilizzo di moduli secondari Git

Alcuni progetti di codice traggono vantaggio dalla disponibilità di moduli secondari Git, repository all'interno del repository di primo livello. Quando distribuisci il tuo codice utilizzando eb create o eb deploy, l'interfaccia a riga di comando EB può includere moduli secondari nel file zip della versione dell'applicazione e caricarli con il resto del codice.

Puoi controllare l'inclusione di moduli secondari utilizzando l'opzione `include_git_submodules` nella sezione `global` del file di configurazione dell'interfaccia a riga di comando EB, `.elasticbeanstalk/config.yml` nella tua cartella di progetto.

Per includere moduli secondari, impostare questa opzione su `true`:

```
global:
  include_git_submodules: true
```

Quando l'opzione `include_git_submodules` è mancante o è impostata su `false`, l'interfaccia a riga di comando EB non include moduli secondari nel file .zip caricato.

Per ulteriori dettagli sui moduli secondari Git, vedi Git Tools - Submodules.

> **Comportamento predefinito**
>
> Quando esegui eb init per configurare il tuo progetto, l'interfaccia a riga di comando EB aggiunge l'opzione `include_git_submodules` e la impostata su `true`. Questo garantisce che eventuali moduli secondari del progetto siano inclusi nelle distribuzioni. L'interfaccia a riga di comando EB non sempre supporta l'inclusione di moduli secondari. Per evitare una modifica accidentale e indesiderabile dei progetti, esistita prima che aggiungessimo il supporto dei moduli secondari, l'interfaccia a riga di comando EB non include moduli secondari quando l'opzione `include_git_submodules` è mancante. Se disponi di uno di questi progetti esistenti e desideri includere moduli secondari nelle tue distribuzioni, aggiungi l'opzione e impostala su `true` come descritto in questa sezione.

> **CodeCommit comportamento**
>
> L'integrazione di Elastic Beanstalk con CodeCommit non supporta i sottomoduli in questo momento. Se hai abilitato l'integrazione con il tuo ambiente CodeCommit, i sottomoduli non sono inclusi nelle distribuzioni.

Assegnazione di tag Git alla tua versione dell'applicazione

Puoi utilizzare un tag Git come etichetta di versione per identificare la versione dell'applicazione in esecuzione nel tuo ambiente. Ad esempio, digita quanto segue:

```
~/eb$ git tag -a v1.0 -m "My version 1.0"
```

Riferimento ai comandi CLI EB

Puoi utilizzare l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) per eseguire una serie di operazioni finalizzate alla distribuzione e alla gestione di applicazioni e ambienti Elastic Beanstalk. La CLI EB si integra con Git per distribuire il codice sorgente dell'applicazione che è sotto il controllo del codice sorgente Git. Per ulteriori informazioni, consulta Configurazione dell'interfaccia a riga di comando EB (EB CLI) per gestire Elastic Beanstalk e Utilizzo dell'interfaccia a riga di comando EB con Git.

Comandi

- Opzioni comuni
- eb abort
- eb appversion
- eb clone
- eb codesource
- eb config
- eb console
- eb create
- eb deploy
- eb events
- eb health
- eb init
- eb labs
- eb list
- eb local
- eb logs
- eb migrate
- eb open
- eb platform
- eb printenv
- eb restore
- eb scale

- eb setenv
- eb ssh
- eb status
- eb swap
- eb tags
- eb terminate
- eb upgrade
- eb use

Opzioni comuni

È possibile utilizzare le opzioni seguenti con tutti i comandi della CLI EB.

Nome	Descrizione
`--debug`	Stampa informazioni per il debug.
`-h`, `--help`	Mostra il messaggio di Aiuto. Tipo: string Impostazione predefinita: nessuna
`--no-verify-ssl`	Ignora la verifica del certificato SSL. Utilizza questa opzione se hai problemi relativi all'uso della CLI con un proxy.
`--profile`	Usa un profilo specifico dal tuo file di AWS credenziali.
`--quiet`	Elimina tutto l'output dal comando.
`--region`	Usa la regione specificata.
`-v`, `--verbose`	Visualizza informazioni dettagliate.

eb abort

Description

Annulla l'aggiornamento quando le modifiche di configurazione dell'ambiente per le istanze sono ancora in corso.

> **Note**
>
> Se disponi di più di due ambienti che sono sottoposti a un aggiornamento, ti verrà richiesto di selezionare il nome dell'ambiente di cui si desidera eseguire il rollback delle modifiche.

Sintassi

eb abort

eb abort ***environment-name***

Opzioni

Nome	Descrizione
Opzioni comuni	

Output

Il comando mostra un elenco di ambienti in fase di aggiornamento e ti verrà richiesto di scegliere l'aggiornamento che desideri interrompere. Se solo un ambiente è attualmente in fase di aggiornamento, non è necessario specificare il nome dell'ambiente. Se l'operazione va a buon fine, il comando ripristina le modifiche di configurazione dell'ambiente. Il processo di rollback prosegue fino a quando tutte le istanze nell'ambiente hanno la configurazione dell'ambiente precedente o fino a quando il processo di rollback dà esito negativo.

Esempio

L'esempio seguente annulla l'aggiornamento della piattaforma.

```
$ eb abort
Aborting update to environment "tmp-dev".
<list of events>
```

eb appversion

Description

Il comando`appversion` di CLI EB gestisce le versioni dell'applicazione Elastic Beanstalk. È possibile creare una nuova versione dell'applicazione senza distribuire, eliminare una versione dell'applicazione o creare la policy del ciclo di vita della versione dell'applicazione. Se invochi il comando senza opzioni, passa alla modalità interattiva.

Utilizza l'opzione `--create` per creare una nuova versione dell'applicazione.

Utilizza l'opzione `--delete` per eliminare una versione dell'applicazione.

Utilizza l'opzione `lifecycle` per visualizzare o creare la policy del ciclo di vita della versione dell'applicazione. Per ulteriori informazioni, consulta the section called “Ciclo di vita delle versioni”.

Sintassi

eb appversion

eb appversion [-c | --create]

eb appversion [-d | --delete] ***`version-label`***

eb appversion lifecycle [-p | --print]

Opzioni

Nome	Descrizione Tipo: String
-un *`application-name`* or	Il nome dell'applicazione. Se non viene trovata un'applicazione con il nome specificato, la CLI EB crea una versione dell'appl icazione per una nuova applicazione.

Nome	Descrizione Tipo: String
--nome_applicazione *application-name*	Applicabile solo con l'opzione `--create`. Tipo: String
-c oppure --creare	Crea una nuova versione dell'applicazione.
-d *version-label* or --delete *version-label*	Elimina la versione dell'applicazione *version-label* etichetta ta.
`-l` *version_label* or `--label` *version_label*	Specifica un'etichetta da utilizzare per la versione che la CLI EB crea. Se non si utilizza questa opzione, la CLI EB genera una nuova etichetta univoca. Se fornisci un'etichetta di versione, assicurati che sia unica. Applicabile solo con l'opzione `--create`. Tipo: string
ciclo di vita	Richiamare l'editor predefinito per creare una nuova policy del ciclo di vita della versione dell'applicazione. Utilizza questa policy per evitare di raggiungere la quota della versione dell'applicazione.
lifecycle -p oppure lifecycle --print	Visualizza la policy attuale del ciclo di vita dell'applicazione.

Nome	Descrizione Tipo: String

`-m "version_description "`
 	
or
 	
`--message "version_description "`
	La descrizione per la versione dell'applicazione. È racchiusa tra virgolette doppie.
	
	Applicabile solo con l'opzione `--create`.
	
	Tipo: string

`-p`
 	
oppure
 	
`--process`
	Pre-elabora e convalida l'ambiente il manifest e i file di configurazione dell'ambiente nel bundle di origine. La convalida dei file di configurazione può identificare i problemi. Si consiglia di eseguire questa operazione prima di distribuire la versione dell'applicazione nell'ambiente.
	
	Applicabile solo con l'opzione `--create`.

`--source codecommit/ repository-name/branch-name`	CodeCommit repository e branch. Applicabile solo con l'opzione `--create`.
`--staged`	Usa i file in scena nell'indice git, invece del commit HEAD, per creare la versione dell'applicazione. Applicabile solo con l'opzione `--create`.
`--timeout minutes`	Il numero di minuti prima che il comando scada. Applicabile solo con l'opzione `--create`.
Opzioni comuni	

Utilizzo del comando in maniera interattiva

Se utilizzi il comando senza argomenti, l'output visualizza le versioni dell'applicazione. Sono elencati in ordine cronologico inverso, con l'ultima versione elencata per prima. Consulta la sezione Examples

(Esempi) per gli esempi su come appare lo schermo. Nota che la riga di stato viene visualizzata nella parte inferiore. La riga di stato mostra le informazioni sensibili al contesto.

Premi d per eliminare una versione dell'applicazione, premi `l` per gestire la policy del ciclo di vita per la tua applicazione o premi q per uscire senza apportare modifiche.

> **Note**
>
> Se la versione è stata distribuita in un ambiente qualsiasi, non è possibile eliminarla.

Output

Il comando con l'opzione `--create` visualizza un messaggio di conferma che la versione dell'applicazione è stata creata.

Il comando con l'`--delete`*`version-label`*opzione visualizza un messaggio che conferma che la versione dell'applicazione è stata eliminata.

Esempi

L'esempio seguente mostra la finestra interattiva di un'applicazione senza distribuzioni.

```
No Environment Specified                                  Application Name: versions
Environment Status: Unknown Health Unknown
Current version # deployed: None

 #   Version Label   Date Created       Age       Description                                appversion
 3   v4              2016/12/22 13:28   56 secs   new features
 2   v3              2016/12/22 13:27   1 min     important update
 1   v1              2016/12/15 23:51   6 days    wow

(Commands: Quit, Delete, Lifecycle, ▼▲ ◀▶)
```

L'esempio seguente mostra la finestra interattiva per un'applicazione con la quarta versione, con l'etichetta della versione Sample Application (Applicazione di esempio), distribuita.

```
Sample-env                                          Application Name: versions
Environment Status: Launching Health Green
Current version # deployed: 4

 #   Version Label        Date Created       Age      Description                                  appversion
 4   Sample Application   2016/12/22 13:30   2 mins   -
 3   v4                   2016/12/22 13:28   4 mins   new features
 2   v3                   2016/12/22 13:27   5 mins   important update
 1   v1                   2016/12/15 23:51   6 days   wow

(Commands: Quit, Delete, Lifecycle, ▼▲ ◀▶)
```

L'esempio seguente mostra l'output di un eb appversion lifecycle -p comando, dove *ACCOUNT-ID* è l'ID dell'account dell'utente:

```
Application details for: lifecycle
  Region: sa-east-1
  Description: Application created from the EB CLI using "eb init"
  Date Created: 2016/12/20 02:48 UTC
  Date Updated: 2016/12/20 02:48 UTC
  Application Versions: ['Sample Application']
  Resource Lifecycle Config(s):
    VersionLifecycleConfig:
      MaxCountRule:
        DeleteSourceFromS3: False
        Enabled: False
        MaxCount: 200
      MaxAgeRule:
        DeleteSourceFromS3: False
        Enabled: False
        MaxAgeInDays: 180
    ServiceRole: arn:aws:iam::ACCOUNT-ID:role/aws-elasticbeanstalk-service-role
```

eb clone

Description

Clona un ambiente in un nuovo ambiente in modo che abbiano impostazioni identiche.

> **Note**
>
> Per impostazione predefinita, indipendentemente dalla versione dello stack di soluzioni dell'ambiente da cui si crea il clone, il comando eb clone crea l'ambiente di clone con lo

stack di soluzioni più recente. Puoi evitarlo includendo l'opzione `--exact` quando esegui il comando.

> **Important**
>
> Gli ambienti Elastic Beanstalk clonati non trasferiscono i gruppi di sicurezza in ingresso, lasciando l'ambiente aperto a tutto il traffico Internet. Dovrai ristabilire i gruppi di sicurezza in ingresso per l'ambiente clonato.
> È possibile visualizzare le risorse che potrebbero non essere clonate controllando lo stato di deriva della configurazione dell'ambiente. Per ulteriori informazioni, consulta Rileva la deriva su un intero CloudFormation stack nella Guida per l'utente.AWS CloudFormation

Sintassi

eb clone

eb clone ***environment-name***

Opzioni

Nome	Descrizione
`-n` *`string`* or `--clone_name` *`string`*	Nome da assegnare all'ambiente clonato.
`-c` *`string`* or `--cname` *`string`*	Prefisso CNAME da aggiungere all'ambiente clonato.
`--envvars`	Proprietà dell'ambiente in un elenco separato da virgole con il formato =. *`name value`*

Nome	Descrizione
	Tipo: String Vincoli: • Le coppie chiave-valore devono essere separate da virgole. • Le chiavi e i valori possono contenere qualsiasi carattere alfabetico in qualsiasi lingua, qualsiasi carattere numerico, separatore sottointeso e i seguenti simboli: _ . : / + \ - @ • Le chiavi possono contenere fino a 128 caratteri. I valori possono contenere fino a 256 caratteri. • Per chiavi e valori viene fatta distinzione tra maiuscole e minuscole. • I valori non possono corrispondere al nome dell'ambiente. • I valori non possono includere `aws:` o `elasticbeanstalk:` . • La dimensione totale di tutte le proprietà dell'ambiente non può superare 4096 byte.
`--exact`	Impedisce a Elastic Beanstalk di aggiornare la versione dello stack di soluzioni per il nuovo ambiente clone alla versione più recente disponibile (per la piattaforma dell'ambiente originale).
`--scale` *`number`*	Il numero di istanze per da eseguire nell'ambiente clone quando viene avviato.
`--tags` *`name`*`=`*`value`*	Tag per le risorse del tuo ambiente in un elenco separato da virgole con il formato =. *`name value`*
`--timeout`	Il numero di minuti prima che il comando scada.
Opzioni comuni	

Output

Se riesce, il comando crea un ambiente che ha le stesse impostazioni dell'ambiente originale o con modifiche specificate dalle opzioni eb clone.

Esempio

L'esempio seguente clona l'ambiente specificato.

```
$ eb clone
Enter name for Environment Clone
(default is tmp-dev-clone):
Enter DNS CNAME prefix
(default is tmp-dev-clone):
Environment details for: tmp-dev-clone
  Application name: tmp
  Region: us-west-2
  Deployed Version: app-141029_144740
  Environment ID: e-vjvrqnn5pv
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev-clone.elasticbeanstalk.com
  Updated: 2014-10-29 22:00:23.008000+00:00
Printing Status:
2018-07-11 21:04:20    INFO: createEnvironment is starting.
2018-07-11 21:04:21    INFO: Using elasticbeanstalk-us-west-2-888888888888 as Amazon S3
 storage bucket for environment data.
...
2018-07-11 21:07:10    INFO: Successfully launched environment: tmp-dev-clone
```

eb codesource

Description

Configura l'EB CLI per la distribuzione da CodeCommit un repository oppure CodeCommit disabilita l'integrazione e carica il bundle di origine dal computer locale.

> **Note**
>
> Alcune regioni non offrono. AWS CodeCommit L'integrazione tra Elastic CodeCommit Beanstalk e non funziona in queste regioni.
> Per informazioni sui AWS servizi offerti in ciascuna regione, consulta la Tabella delle regioni.

Sintassi

eb codesource

eb codesource codecommit

eb codesource local

Opzioni

Nome	Descrizione
Opzioni comuni	

Output

eb codesourceti chiede di scegliere tra CodeCommit integrazione e distribuzioni standard.

eb codesource codecommit avvia la configurazione interattiva dell'archivio per l'integrazione di CodeCommit .

eb codesource local mostra la configurazione originale e disabilita l'integrazione di CodeCommit.

Esempi

Utilizza eb codesource codecommit per configurare l' CodeCommit integrazione per il ramo corrente.

```
~/my-app$ eb codesource codecommit
Select a repository
1) my-repo
2) my-app
3) [ Create new Repository ]
(default is 1): 1
```

```
Select a branch
1) mainline
2) test
3) [ Create new Branch with local HEAD ]
(default is 1): 1
```

Usa eb codesource local per disabilitare CodeCommit l'integrazione per il ramo corrente.

```
~/my-app$ eb codesource local
Current CodeCommit setup:
  Repository: my-app
  Branch: mainline
Default set to use local sources
```

eb config

Description

Gestisce le impostazioni di configurazione attive e le configurazioni salvate dell'ambiente. È possibile utilizzare questo comando per caricare, scaricare o elencare le configurazioni salvate dell'ambiente. È inoltre possibile utilizzarlo per scaricare, visualizzare o aggiornare le impostazioni di configurazione attive.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando modifica anche le impostazioni di configurazione del generatore. Questa operazione viene eseguita in base ai valori impostati in `platform.yaml`.

> **Note**
>
> eb config non visualizza le proprietà di ambiente. Per impostare proprietà di ambiente che puoi leggere dalla tua applicazione, utilizza eb setenv.

Sintassi

Di seguito sono riportate alcune parti della sintassi utilizzata per il comando eb config per funzionare con le impostazioni di configurazione attive dell'ambiente. Per esempi specifici, consulta la sezione Esempi più avanti in questo argomento.

- eb config— Visualizza le impostazioni di configurazione attive dell'ambiente in un editor di testo configurato come variabile di ambiente EDITOR. Quando salvi le modifiche al file e chiudi l'editor, l'ambiente viene aggiornato con le impostazioni delle opzioni nel file.

> **Note**
>
> Se non hai configurato una variabile di ambiente EDITOR, la CLI EB visualizza le impostazioni delle opzioni nell'editor di default per i file YAML.

- eb config ***environment-name***— Visualizza e aggiorna la configurazione per l'ambiente denominato. La configurazione viene visualizzata in un editor di testo configurato o nei file YAML dell'editor predefinito.
- eb config save— Salva le impostazioni di configurazione attive per l'ambiente corrente in `.elasticbeanstalk/saved_configs/` con il nome di file`[configuration-name].cfg.yml`. Per impostazione predefinita, l'EB CLI salva le impostazioni di configurazione con un nome *`configuration-name`* basato sul nome dell'ambiente. Puoi specificare un nome di configurazione differente includendo l'opzione `--cfg` con il nome di configurazione desiderato quando esegui il comando.

  Puoi assegnare un tag alla configurazione salvata utilizzando l'opzione `--tags`.
- eb config **`--display`**— Scrive le impostazioni di configurazione attive di un ambiente su stdout anziché su un file. Per impostazione predefinita, vengono visualizzate le impostazioni di configurazione sul terminale.
- eb config **`--update`** ***`configuration_string`*** **`|`** ***`file_path`***— Aggiorna le impostazioni di configurazione attive per l'ambiente corrente con le informazioni specificate nel *`configuration_string`* o all'interno del file identificato da*`file_path`*.

> **Note**
>
> Le opzioni `--display` e `--update` offrono flessibilità per la lettura e la revisione delle impostazioni di configurazione di un ambiente a livello di programmazione.

Quanto segue descrive la sintassi per l'utilizzo del comando eb config con le configurazioni salvate. Per alcuni esempi, consulta la sezione Esempi più avanti in questo argomento.

- eb config get ***`config-name`***— Scarica la configurazione salvata con nome da Amazon S3.

- eb config delete ***config-name*** — Elimina la configurazione salvata con nome da Amazon S3. La elimina anche localmente, se lo hai già scaricato.
- eb config list – Elenca le configurazioni salvate presenti in Amazon S3.
- eb config put ***filename*** – Carica la configurazione salvata con nome in un bucket Amazon S3. *filename*Deve avere l'estensione del file `.cfg.yml`. Per specificare il nome di file senza un percorso, puoi salvare il file nella cartella `.elasticbeanstalk` o nella cartella `.elasticbeanstalk/saved_configs/` prima di eseguire il comando. In alternativa, è possibile specificare il *filename* percorso completo.

Opzioni

Nome	Descrizione
`--cfg` *config-name*	Il nome da utilizzare per una configurazione salvata. Questa opzione funziona solo con eb config save.

`-d`
 	
oppure
 	
`--display`
	Visualizza le impostazioni di configurazione per l'ambiente corrente (scrive su stdout).
	
	Utilizza con l'opzione `--format` per specificare l'output da utilizzare in JSON o YAML. Se non lo specifichi, l'output è in formato YAML.
	
	Questa opzione funziona solo se utilizzi il comando eb config senza gli altri sottocomandi.

`-f` *format_type* oppure `--format` *format_type*	Specifica il formato di visualizzazione. I valori validi sono JSON o YAML. Il valore predefinito è YAML. Questa opzione funziona solo con l'opzione `--display` .
`--tags` *key1*=*value1*[,*ke*	Tag da aggiungere alla configurazione salvata. Quando devi specificare i tag nell'elenco, specificali come coppie chiave=va lore e separali con una virgola.

Nome	Descrizione
	Per ulteriori informazioni, consulta Tagging delle configurazioni salvate. Questa opzione funziona solo con eb config save.
`--timeout` *`timeout`*	Il numero di minuti prima che il comando scada.

Nome	Descrizione

`-u` *`configuration_string`* | *`file_path`*
 	
oppure
 	
`--update` *`configuration_string`* | *`file_path`*
	Aggiorna le impostazioni di configurazione attive per l'ambiente corrente.
	
	Questa opzione funziona solo se utilizzi il comando eb config senza gli altri sottocomandi.
	
	Il parametro *`configuration_string`* | *`file_path`* è del tipo stringa. La stringa fornisce l'elenco degli spazi dei nomi e delle opzioni corrispondenti da aggiungere, aggiornare o rimuovere dalle impostazioni di configurazione per l'ambiente. In alternativa, la stringa di input può rappresentare un file che contiene le stesse informazioni.
	
	Per specificare un nome di file, la stringa di input deve seguire il formato `"file://<` *`path`*`><`*`filename`*`>"`. Per specificare il nome del file senza un *`path`*, salva il file nella cartella in cui esegui il comando. In alternativa, puoi specificare il nome di file fornendo il percorso completo.
	
	Le informazioni di configurazione devono soddisfare le seguenti condizioni. Almeno una delle sezioni, OptionSettings oOptionsToRemove, è richiesta. Utilizza OptionSettings per aggiungere o modificare le opzioni. UtilizzaOptionsToRemove per rimuovere le opzioni da uno spazio dei nomi. Per esempi specifici, consulta la sezione Esempi più avanti in questo argomento.
	
	Example
	
	Formato YAML
	
	`OptionSettings:`
	`  namespace1:`
	`  option-name-1:` *`option-value-1`*
	`  option-name-2:` *`option-value-2`*
	`  ...`
	`OptionsToRemove:`
	`  namespace1:`
	`  option-name-1`

Nome	Descrizione

	`option-name-2`
	`...`
	
	Example
	
	Formato JSON
	
	`{`
	`"OptionSettings": {`
	`"namespace1": {`
	`"option-name-1": " option-value-1 ",`
	`"option-name-2": " option-value-2 ",`
	`...`
	`}`
	`},`
	`"OptionsToRemove": {`
	`"namespace1": {`
	`"option-name-1",`
	`"option-name-2",`
	`...`
	`}`
	`}`
	`}`

Opzioni comuni	

Output

Se il comando eb config o eb config ***environment-name*** viene eseguito correttamente senza aggiungere sottocomandi o opzioni, il comando visualizza le impostazioni delle opzioni correnti nell'editor di testo configurato come variabile di ambiente EDITOR. Se non hai configurato una variabile di ambiente EDITOR, la CLI EB visualizza le impostazioni delle opzioni nell'editor di default per i file YAML.

Quando salvi le modifiche al file e chiudi l'editor, l'ambiente viene aggiornato con le impostazioni delle opzioni nel file. Viene visualizzato il seguente output per confermare l'aggiornamento della configurazione.

```
$ eb config myApp-dev
    Printing Status:
    2021-05-19 18:09:45    INFO    Environment update is starting.
    2021-05-19 18:09:55    INFO    Updating environment myApp-dev's configuration
 settings.
    2021-05-19 18:11:20    INFO    Successfully deployed new configuration to
 environment.
```

Se il comando viene eseguito correttamente con l'opzione `--display`, visualizza le impostazioni di configurazione per l'ambiente corrente (scrive su stdout).

Se il comando viene eseguito senza errori con il parametro `get`, visualizza il percorso della copia locale che hai scaricato.

Se il comando viene eseguito senza errori con il parametro `save`, visualizza il percorso del file salvato.

Esempi

Questa sezione descrive come modificare l'editor di testo utilizzato per visualizzare e modificare il file delle impostazioni.

Per Linux e UNIX, il seguente esempio imposta vim come editor:

```
$ export EDITOR=vim
```

Per Linux e UNIX, il seguente esempio imposta l'editor installato in `/usr/bin/kate`.

```
$ export EDITOR=/usr/bin/kate
```

Per Windows, il seguente esempio imposta Notepad++ come editor.

```
> set EDITOR="C:\Program Files\Notepad++\Notepad++.exe
```

Questa sezione fornisce esempi per il comando eb config quando viene eseguito con sottocomandi.

L'esempio seguente elimina la configurazione salvata denominata `app-tmp`.

```
$ eb config delete app-tmp
```

L'esempio seguente scarica la configurazione salvata denominata app-tmp dal bucket Amazon S3.

```
$ eb config get app-tmp
```

L'esempio seguente elenca i nomi delle configurazioni archiviate nel bucket Amazon S3.

```
$ eb config list
```

L'esempio seguente carica la copia locale della configurazione salvata denominata app-tmp nel bucket Amazon S3.

```
$ eb config put app-tmp
```

L'esempio seguente salva le impostazioni di configurazione dell'ambiente attualmente in esecuzione. Se non indichi un nome per la configurazione salvata, Elastic Beanstalk assegna un nome al file di configurazione in base al nome dell'ambiente. Ad esempio, con un ambiente denominato tmp-dev, il nome sarebbe `tmp-dev.cfg.yml`. Elastic Beanstalk salva il file nella cartella `/.elasticbeanstalk/saved_configs/`.

```
$ eb config save
```

L'esempio seguente mostra come utilizzare l'opzione `--cfg` per salvare le impostazioni di configurazione dell'ambiente tmp-dev in un file denominato `v1-app-tmp.cfg.yml`. Elastic Beanstalk salva il file nella cartella `/.elasticbeanstalk/saved_configs/`. Se non specifichi un nome di ambiente, Elastic Beanstalk salva le impostazioni di configurazione dell'ambiente attualmente in esecuzione.

```
$ eb config save tmp-dev --cfg v1-app-tmp
```

Questa sezione fornisce esempi per il comando eb config quando viene eseguito senza sottocomandi.

Il comando seguente visualizza le impostazioni delle opzioni dell'ambiente corrente in un editor di testo.

```
$ eb config
```

Il comando seguente visualizza le impostazioni delle opzioni per l'ambiente my-env in un editor di testo.

```
$ eb config my-env
```

Nell'esempio seguente vengono visualizzate le impostazioni delle opzioni per l'ambiente corrente. Viene emesso nel formato YAML perché non è stato specificato alcun formato specifico con l'opzione `--format`.

```
$ eb config --display
```

L'esempio seguente aggiorna le impostazioni delle opzioni per l'ambiente corrente con le specifiche del file denominato `example.txt`. Il file è in formato YAML o JSON. La CLI EB rileva automaticamente il formato del file.

- L'opzione Minsize è impostata su 1 per lo spazio dei nomi `aws:autoscaling:asg`.
- La dimensione del batch per lo spazio dei nomi `aws:elasticbeanstalk:command` è impostata a 30%.
- Rimuove l'impostazione dell'opzione IdleTimeout: None dallo spazio dei nomi`AWSEBV2LoadBalancer.aws:elbv2:loadbalancer`.

```
$ eb config --update "file://example.txt"
```

Example- nome file: example.txt - formato YAML

```
OptionSettings:
  'aws:elasticbeanstalk:command':
    BatchSize: '30'
    BatchSizeType: Percentage
  'aws:autoscaling:asg':
    MinSize: '1'
OptionsToRemove:
  'AWSEBV2LoadBalancer.aws:elbv2:loadbalancer':
    IdleTimeout
```

Example- nome file: example.txt - formato JSON

```
{
    "OptionSettings": {
        "aws:elasticbeanstalk:command": {
            "BatchSize": "30",
```

```
            "BatchSizeType": "Percentage"
        },
        "aws:autoscaling:asg": {
            "MinSize": "1"
        }
    },
    "OptionsToRemove": {
        "AWSEBV2LoadBalancer.aws:elbv2:loadbalancer": {
            "IdleTimeout"
        }
    }
}
```

Gli esempi seguenti aggiornano le impostazioni delle opzioni per l'ambiente corrente. Il comando imposta l'opzione Minsize a 1 per lo spazio dei nomi `aws:autoscaling:asg`.

> **Note**
>
> Questi esempi sono specifici di Windows. PowerShell Essi sfuggono alle occorrenze letterali delle doppie virgolette (") precedendolo con uno slash (\). Sistemi operativi e ambienti a riga di comando diversi potrebbero avere sequenze di escape diverse. Per questo motivo, consigliamo di utilizzare l'opzione file mostrata negli esempi precedenti. La specifica delle opzioni di configurazione in un file non richiede caratteri di escape ed è coerente tra i diversi sistemi operativi.

L'esempio seguente è in formato JSON. La CLI EB rileva se il formato è in JSON o YAML.

```
PS C:\Users\myUser\EB_apps\myApp-env>eb config --update '{\"OptionSettings\":
{\"aws:autoscaling:asg\":{\"MaxSize\":\"1\"}}}'
```

L'esempio seguente è in formato YAML. Per immettere la stringa YAML nel formato corretto, il comando include la spaziatura e i end-of-line ritorni richiesti in un file YAML.

- Termina ogni riga con il tasto «invio» o «indietro».
- Inizia la seconda riga con due spazi e inizia la terza riga con quattro spazi.

```
PS C:\Users\myUser\EB_apps\myApp-env>eb config --update 'OptionSettings:
>>  aws:autoscaling:asg:
```

```
>>    MinSize: \"1\"'
```

eb console

Description

Apre un browser per visualizzare il pannello di controllo di configurazione dell'ambiente nella Console di gestione di Elastic Beanstalk.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando consente di visualizzare anche la configurazione dell'ambiente di generazione, come specificato in `platform.yaml` nella Console di gestione di Elastic Beanstalk.

Sintassi

eb console

eb console ***environment-name***

Opzioni

Nome	Descrizione
Opzioni comuni	

eb create

Description

Crea un nuovo ambiente e distribuisce una versione dell'applicazione nell'ambiente.

> **Note**
>
> - Per utilizzare eb create in una applicazione .NET, è necessario creare un pacchetto di distribuzione come descritto in Creazione di un bundle di origine per un'applicazione .NET, quindi impostare la configurazione della CLI per distribuire il pacchetto come un artefatto, come descritto in Distribuzione di artifact invece della cartella del progetto.

- La creazione di ambienti con CLI EB richiede un ruolo del servizio. Puoi creare un ruolo di servizio tramite la creazione di un ambiente nella console Elastic Beanstalk. Se non disponi di un ruolo del servizio, CLI EB tenta di crearne uno durante l'esecuzione di `eb create`.

Puoi distribuire la versione dell'applicazione da alcune origini:

- Per impostazione predefinita: dal codice sorgente dell'applicazione nella directory di progetto locale.
- Utilizzando l'opzione `--version`: da una versione dell'applicazione già esistente nell'applicazione.
- Quando la directory del progetto non include codice dell'applicazione oppure usando l'opzione `--sample`: implementazione da un'applicazione di esempio, specifica per la piattaforma dell'ambiente.

Sintassi

eb create

eb create ***environment-name***

La lunghezza del nome di un ambiente deve essere compresa tra 4 e 40 caratteri. Il nome può includere solo lettere, numeri e trattini (-). I nomi degli ambienti non possono iniziare né terminare con un trattino.

Se includi un nome di ambiente nel comando, la CLI EB non richiede di effettuare alcuna selezione né di creare un ruolo di servizio.

Se esegui il comando senza un argomento del nome ambiente, questo viene eseguito in un flusso interattivo e ti viene richiesto di immettere o selezionare i valori per alcune impostazioni. Se in questo flusso interattivo distribuisci un'applicazione di esempio, la CLI EB ti chiede anche se vuoi scaricare l'applicazione di esempio nella directory di progetto locale. Dopo il download, potrai utilizzare la CLI EB con il nuovo ambiente in un secondo momento per eseguire operazioni che richiedono il codice dell'applicazione, ad esempio eb deploy.

Alcuni prompt di flusso interattivi vengono visualizzati solo in determinate condizioni. Ad esempio, se si sceglie di utilizzare un Application Load Balancer, e l'account dispone di almeno un Application Load Balancer condivisibile, Elastic Beanstalk visualizza un messaggio che chiede se si desidera

utilizzare un sistema di bilanciamento del carico condiviso. Questo prompt non viene visualizzato se nell'account non è presente un Application Load Balancer condivisibile.

Opzioni

Nessuna di queste opzioni è obbligatoria. Se esegui eb create senza opzioni, la CLI EB richiede di immettere o selezionare un valore per ogni impostazione.

Nome	Descrizione

`-d`
oppure
`--branch_default`
	Imposta l'ambiente come l'ambiente predefinito per il repository corrente.

`--cfg` *`config-name`*	Utilizza le impostazioni della piattaforma da una configurazione salvata in `.elasticbeanstalk/saved_configs/` o nel bucket Amazon S3. Specifica solo il nome del file, senza l'estensione `.cfg.yml`.
`-c` *`subdomain-name`* or `--cname` *`subdomain-name`*	Nome del sottodominio da anteporre alla voce DNS CNAME che instrada al sito Web. Tipo: string Impostazione predefinita: nome dell'ambiente

`-db`
oppure
`--database`
	Collega un database all'ambiente. Se esegui eb create con l'opzione `--database`, ma senza le opzioni `--database.username` e `--database.password`, la CLI EB richiede il nome utente e la password del database master.

`-db.engine` *`engine`* or `--database.engine` *`engine`*	Il tipo di motore di database. Se esegui eb create con questa opzione, la CLI EB avvia l'ambiente con un database collegato. Questo accade anche se non hai eseguito il comando con l'opzione `--database`. Tipo: string

Nome	Descrizione
	Valori validi: `mysql`, `oracle-se1` , `postgres`, `sqlserver-ex` , `sqlserver-web` , `sqlserver-se`
`-db.i` *`instance_type`* or `--database.instance` *`instance_type`*	Il tipo di EC2 istanza Amazon da utilizzare per il database. Se esegui eb create con questa opzione, la CLI EB avvia l'ambiente con un database collegato. Questo accade anche se non hai eseguito il comando con l'opzione `--database` . Tipo: String Valori validi: Amazon RDS supporta un set standard di istanze database. Per selezionare un'istanza database appropria ta per il motore database, è necessario tenere conto di alcune considerazioni specifiche. Per ulteriori informazi oni, consulta la pagina relativa alle classi di istanze database nella Guida per l'utente di Amazon RDS.
`-db.pass` *`password`* or `--database.password` *`password`*	Password per il database. Se esegui eb create con questa opzione, la CLI EB avvia l'ambiente con un database collegato. Questo accade anche se non hai eseguito il comando con l'opzione `--database` .

Nome	Descrizione
`-db.size` *`number_of_gigabytes`* or `--database.size` *`number_of_gigabytes`*	Numero di gigabyte (GB) da allocare per lo storage del database. Se esegui eb create con questa opzione, la CLI EB avvia l'ambiente con un database collegato. Questo accade anche se non hai eseguito il comando con l'opzione `--database` . Tipo: numero Valori validi: • MySQL: da `5` a `1024`. Il valore di default è `5`. • Postgres: da `5` a `1024`. Il valore di default è `5`. • Oracle: da `10` a `1024`. Il valore di default è `10`. • Microsoft SQL Server Express Edition – `30`. • Microsoft SQL Server Web Edition – `30`. • Microsoft SQL Server Standard Edition – `200`.
`-db.user` *`username`* or `--database.username` *`username`*	Nome utente per il database. Se esegui eb create con questa opzione, la CLI EB avvia l'ambiente con un database collegato anche se non hai eseguito il comando con l'opzione `--database` . Se esegui eb create con l'opzione `--database` , ma senza le opzioni `--database.username` e `--database.password` , la CLI EB richiede il nome utente e la password del database master.
`-db.version` *`version`* or `--database.version` *`version`*	Utilizzato per specificare la versione del motore di database. Se questo flag è presente, l'ambiente viene avviato con un database con il numero di versione specificato, anche se il flag `--database` non è presente.

Nome	Descrizione
`--elb-type` *`type`*	Tipo di sistema di bilanciamento del carico. Tipo: string Valori validi: `classic`, `application` , `network` Default: `application`

`-es`
 	
oppure
 	
`--enable-spot`
	Abilitare le richieste di istanze Spot per l'ambiente in uso. Per ulteriori informazioni, consulta Gruppo Auto Scaling.
	
	Opzioni correlate:
	
	• `--instance-types`
	• `--on-demand-base-capacity`
	• `--on-demand-above-base-capacity`
	• `--spot-max-price`

`--env-group-suffix` *`groupname`*	Il nome del gruppo a cui aggiungere il nome dell'ambi ente. Da utilizzare solo con ambienti Compose.
`--envvars`	Proprietà dell'ambiente in un elenco separato da virgole con il formato =. *`name value`* Per informazioni sui limiti, consulta Configurazione delle proprietà dell'ambiente (variabili di ambiente).
`-ip` *`profile_name`* or `--instance_profile` *`profile_name`*	Il profilo dell'istanza con il ruolo IAM con le credenziali di sicurezza temporanee necessarie all'applicazione per accedere alle risorse. AWS

Nome	Descrizione

`-it`
 	
or
 	
`--instance-types` *`type1`*`[,`*`type2`*` ...]`
	Un elenco separato da virgole dei tipi di EC2 istanze Amazon che desideri utilizzare nel tuo ambiente. Se non si specifica questa opzione, Elastic Beanstalk fornisce i tipi di istanza predefiniti.
	
	Per ulteriori informazioni, consulta EC2 Istanze Amazon e Gruppo Auto Scaling.
	
	Important
	La CLI EB applica questa opzione solo alle istanze Spot. A meno che questa opzione non venga utilizzata con l'opzione `--enable-spot` , la CLI EB la ignora. Per specificare un tipo di istanza per un'istanza on demand, utilizzare l'opzione `--intance-type` (senza "s").

`-i`
 	
or
 	
`--instance_type`
	Il tipo di EC2 istanza Amazon che desideri venga utilizzat o dal tuo ambiente. Se non si specifica questa opzione, Elastic Beanstalk fornisce un tipo di istanza predefinito.
	
	Per ulteriori informazioni, consulta EC2 Istanze Amazon.
	
	Important
	La CLI EB applica questa opzione solo alle istanze on demand. Non utilizzare questa opzione con `--enable-spot` , perché la CLI EB la ignora. Per specificare un tipo di istanza per un'istanza Spot, utilizzare l'opzione `--intance-types` (senza "s").

Nome	Descrizione
`-k` *`key_name`* or `--keyname` *`key_name`*	Il nome della coppia di EC2 chiavi Amazon da utilizzar e con il client Secure Shell (SSH) per accedere in modo sicuro alle EC2 istanze Amazon che eseguono l'applica zione Elastic Beanstalk. Se includi questa opzione con il comando eb create, il valore fornito sovrascrive qualsiasi nome di chiave specificato con eb init. Valori validi: un nome chiave esistente registrato con Amazon EC2
`-im` *`number-of-instances`* or `--min-instances` *`number-of-instances`*	Il numero minimo di EC2 istanze Amazon di cui hai bisogno per il tuo ambiente. Tipo: Number (integer) Default: 1 Valori validi: da 1 a `10000`
`-ix` *`number-of-instances`* or `--max-instances` *`number-of-instances`*	Il numero massimo di EC2 istanze Amazon che il tuo ambiente può avere. Tipo: Number (integer) Default: `4` Valori validi: da 1 a `10000`
`--modules` *`component-a component-b`*	Un elenco degli ambienti di componente da creare. Da utilizzare solo con ambienti Compose.

Nome	Descrizione

`-sb`
 	
oppure
 	
`--on-demand-base-capacity`
	Il numero minimo di istanze on demand di cui il gruppo Auto Scaling effettua il provisioning prima di considerare le istanze Spot durante la scalabilità dell'ambiente.
	
	Questa opzione può essere specificata solo con l'opzione `--enable-spot` . Per ulteriori informazioni, consulta Gruppo Auto Scaling.
	
	Tipo: Number (integer)
	
	Default: `0`
	
	Valori validi: `0` su `--max-instances` (quando assente: opzione `MaxSize` nello spazio dei nomi `aws:autoscaling:asg`)

`-sp`
 	
oppure
 	
`--on-demand-above-base-capacity`
	La percentuale di istanze on demand come parte della capacità aggiuntiva che il gruppo Auto Scaling fornisce in aggiunta al numero di istanze specificato dall'opzione `--on-demand-base-capacity` .
	
	Questa opzione può essere specificata solo con l'opzione `--enable-spot` . Per ulteriori dettagli, consulta Gruppo Auto Scaling.
	
	Tipo: Number (integer)
	
	Impostazione predefinita: `0` per un ambiente a istanza singola; `70` per un ambiente con bilanciamento del carico
	
	Valori validi: da `0` a `100`

Nome	Descrizione
`-p` *`platform-version`* or `--platform` *`platform-version`*	La versione della piattaforma da utilizzare. Puoi specificare una piattaforma, una piattaforma e una versione della piattaforma, un nome dello stack di soluzioni o un ARN dello stack di soluzioni. Ad esempio: • `php`, `PHP`, `node.js`: la versione più recente per la piattaforma specificata • `php-7.2`, `"PHP 7.2"`: la versione della piattaforma PHP 7.2 consigliata (tipicamente la più recente) • `"PHP 7.2 running on 64bit Amazon Linux"`: la versione della piattaforma PHP consigliata (tipicamente la più recente) in questo ramo della piattaforma • `"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"`: la versione della piattaforma PHP specificata da questo nome dello stack di soluzioni • `"arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"` : la versione della piattaforma PHP specificata da questo ARN dello stack di soluzioni Utilizza `eb platform list` per ottenere un elenco delle configurazioni disponibili. Se si specifica l'opzione `--platform` , ignora il valore fornito durante `eb init`.

`-pr`
 	
oppure
 	
`--process`
	Pre-elabora e convalida l'ambiente il manifest e i file di configurazione dell'ambiente nel bundle di origine. La convalida dei file di configurazione è in grado di identificare i problemi prima di distribuire la versione dell'applicazione in un ambiente.

Nome	Descrizione
`-r` *`region`* or `--region` *`region`*	La AWS regione in cui desideri distribuire l'applicazione. Per l'elenco dei valori che è possibile specificare per questa opzione, consulta Endpoint e quote AWS Elastic Beanstalk nella sezione Riferimenti generali di AWS.
`--sample`	Distribuisce l'applicazione di esempio nel nuovo ambiente anziché il codice nel repository.
`--scale` *`number-of-instances`*	Esegue l'avvio con il numero di istanze specificato
`--service-role` *`servicerole`*	Assegna un ruolo di servizio non predefinito all'ambiente. **Note** Non inserire un ARN. Inserisci solo il nome del ruolo. Elastic Beanstalk aggiunge davanti al nome del ruolo i valori corretti per creare internamente gli ARN risultanti.

Nome	Descrizione
`-ls` *`load-balancer`* or `--shared-lb` *`load-balancer`*	Configurare l'ambiente per l'utilizzo di un sistema di bilanciamento del carico condiviso. Fornire il nome o l'ARN di un sistema di bilanciamento del carico condivisibile nell'account, un Application Load Balancer creato in modo esplicito, non uno creato da un altro ambiente Elastic Beanstalk. Per ulteriori informazioni, consulta Application Load Balancer condiviso. Esempi di parametri: • `FrontEndLB` : un nome del sistema di bilanciamento del carico. • `arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc` : un ARN di Application Load Balancer. È possibile specificare questa opzione solo con `--elb-type application` . Se si specifica tale opzione e non si specifica `--shared-lb` , Elastic Beanstalk crea un sistema di bilanciamento del carico dedicato per l'ambiente.

Nome	Descrizione
`-lp` *`port`* or `--shared-lb-port` *`port`*	La porta del listener predefinita del sistema di bilanciamento del carico condiviso per questo ambiente. Elastic Beanstalk aggiunge una regola del listener che instrada tutto il traffico da questo listener al processo ambiente predefinito. Per ulteriori informazioni, consulta Application Load Balancer condiviso. Tipo: Number (integer) Default: `80` Valori validi: qualsiasi numero intero che rappresenta una porta del listener del sistema di bilanciamento del carico condiviso.
`--single`	Crea l'ambiente con una singola EC2 istanza Amazon e senza un sistema di bilanciamento del carico. **Warning** Un ambiente con una singola istanza non è pronto per la produzione. Se l'istanza diviene instabile durante la distribuzione o Elastic Beanstalk termina e riavvia l'istanza durante un aggiornamento della configurazione, l'applicazione può risultare non disponibile per un certo periodo di tempo. Utilizza ambienti con una singola istanza per sviluppare, testare o gestire temporaneamente. Per la produzione, utilizza ambienti con bilanciamento del carico.

Nome	Descrizione

`-sm`
 	
oppure
 	
`--spot-max-price`
	Il prezzo massimo orario per unità che si è disposti a pagare per un'istanza Spot in dollari americani.
	
	Questa opzione può essere specificata solo con l'opzione `--enable-spot` . Per ulteriori dettagli, consulta Gruppo Auto Scaling.
	
	Tipo: Numero (float)
	
	Per impostazione predefinita: prezzo on demand per ogni tipo di istanza. Il valore dell'opzione in questo caso è `null`.
	
	Valori validi: da `0.001` a `20.0`
	
	Per consigli sulle opzioni di prezzo massimo per le istanze Spot, consulta la cronologia dei prezzi delle istanze Spot nella Amazon EC2 User Guide.

`--tags` *`key1=value1[,key2=va`*	Applica tag alle risorse nell'ambiente. I tag sono specifica ti come elenco di coppie `key=value` separate da virgole. Per ulteriori informazioni, consulta Etichettatura degli ambienti.

`-t worker`
 	
oppure
 	
`--tier worker`
	Crea un ambiente di ruoli di lavoro. Ometti questa opzione per creare un ambiente di server Web.

`--timeout` *`minutes`*	Imposta il numero di minuti prima del timeout del comando.

Nome	Descrizione
`--version` *`version_label`*	Specifica la versione dell'applicazione che deve essere distribuita nell'ambiente anziché il codice sorgente dell'applicazione nella directory di progetto locale. Tipo: string Valori validi: un'etichetta di versione dell'applicazione esistente
`--vpc`	Configura un VPC per l'ambiente. Quando includi questa opzione, la CLI EB richiede di immettere tutte le impostazi oni necessarie prima di avviare l'ambiente.
`--vpc.dbsubnets` *`subnet1,subnet2`*	Specifica le sottoreti per le istanze di database in un VPC. Opzione obbligatoria quando specifichi `--vpc.id`.
`--vpc.ec2subnets` *`subnet1,subnet2`*	Speciifica le sottoreti per le EC2 istanze Amazon in un VPC. Opzione obbligatoria quando specifichi `--vpc.id`.
`--vpc.elbpublic`	Avvia il sistema di bilanciamento del carico Elastic Load Balancing in una sottorete pubblica nel VPC. Non puoi specificare questa opzione con `--tier worker` o `--single`.
`--vpc.elbsubnets` *`subnet1,subnet2`*	Specifica le sottoreti per il sistema di bilanciamento del carico Elastic Load Balancing in un VPC. Non puoi specificare questa opzione con `--tier worker` o `--single`.
`--vpc.id` *`ID`*	Avvia l'ambiente nel VPC specificato.
`--vpc.publicip`	Avvia le tue EC2 istanze Amazon in una sottorete pubblica del tuo VPC. Non puoi specificare questa opzione con `--tier worker`.

Nome	Descrizione
`--vpc.securitygroups` *`securitygroup1,securitygroup2`*	Speciifica il gruppo di sicurezza. IDs Opzione obbligatoria quando specifichi `--vpc.id`.
Opzioni comuni	

Output

In caso di esito positivo, il comando pone alcune domande e restituisce lo stato dell'operazione di creazione. Se si sono verificati problemi durante l'avvio, puoi utilizzare l'operazione eb events per ottenere ulteriori dettagli.

Se hai abilitato il CodeBuild supporto nell'applicazione, eb create visualizza le informazioni relative alla creazione del codice. CodeBuild Per informazioni sul CodeBuild supporto in Elastic Beanstalk, consulta. Utilizzo dell'EB CLI con AWS CodeBuild

Esempi

L'esempio seguente crea un ambiente in modalità interattiva.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
Environment details for: tmp-dev
  Application name: tmp
  Region: us-east-2
  Deployed Version: app-141029_145448
  Environment ID: e-um3yfrzq22
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
```

```
  Updated: 2014-10-29 21:54:51.063000+00:00
Printing Status:
...
```

Anche l'esempio seguente crea un ambiente in modalità interattiva. In questo esempio la directory di progetto non include codice dell'applicazione. Il comando distribuisce un'applicazione di esempio e la scarica nella directory di progetto locale.

```
$ eb create
Enter Environment Name
(default is tmp-dev): ENTER
Enter DNS CNAME prefix
(default is tmp-dev): ENTER
Select a load balancer type
1) classic
2) application
3) network
(default is 2): ENTER
NOTE: The current directory does not contain any source code. Elastic Beanstalk is
 launching the sample application instead.
Do you want to download the sample application into the current directory?
(Y/n): ENTER
INFO: Downloading sample application to the current directory.
INFO: Download complete.
Environment details for: tmp-dev
  Application name: tmp
  Region: us-east-2
  Deployed Version: Sample Application
  Environment ID: e-um3yfrzq22
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
  Updated: 2017-11-08 21:54:51.063000+00:00
Printing Status:
...
```

Il comando seguente crea un ambiente senza visualizzare alcuna richiesta.

```
$ eb create dev-env
Creating application version archive "app-160312_014028".
Uploading test/app-160312_014028.zip to S3. This may take a while.
Upload Complete.
```

```
Application test has been created.
Environment details for: dev-env
  Application name: test
  Region: us-east-2
  Deployed Version: app-160312_014028
  Environment ID: e-6fgpkjxyyi
  Platform: 64bit Amazon Linux 2015.09 v2.0.8 running PHP 5.6
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-03-12 01:40:33.614000+00:00
Printing Status:
...
```

Il comando seguente crea un ambiente in un VPC personalizzato.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --
vpc.securitygroup sg-70cff265
Creating application version archive "app-160312_014309".
Uploading test/app-160312_014309.zip to S3. This may take a while.
Upload Complete.
Environment details for: dev-vpc
  Application name: test
  Region: us-east-2
  Deployed Version: app-160312_014309
  Environment ID: e-pqkcip3mns
  Platform: 64bit Amazon Linux 2015.09 v2.0.8 running Java 8
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-03-12 01:43:14.057000+00:00
Printing Status:
...
```

eb deploy

Description

Consente di distribuire il bundle di origine dell'applicazione dalla directory di progetto inizializzato all'applicazione in esecuzione.

Se git è installato, la CLI EB usa il comando `git archive` per creare un file `.zip` dal contenuto del comando `git commit` più recente.

Tuttavia, quando `.ebignore` è presente nella directory del progetto, la CLI EB non utilizza comandi e semantiche git per creare il bundle di origine. Questo significa che la CLI EB ignora i file specificati in `.ebignore` e include tutti gli altri file. In particolare, include i file sorgente non eseguiti.

> **Note**
>
> È possibile anche configurare la CLI EB per distribuire un artefatto del processo di build anziché creare un file ZIP della cartella di progetto. Per informazioni dettagliate, vedi Distribuzione di artifact invece della cartella del progetto.

Sintassi

eb deploy

eb deploy ***environment-name***

Opzioni

Nome	Descrizione
`-l` *`version_label`* or `--label` *`version_label`*	Specifica un'etichetta da utilizzare per la versione che la CLI EB crea. Se l'etichetta è già stata utilizzata, la CLI EB ridistribuisce la versione precedente con tale etichetta. Tipo: string
`--env-group-suffix` *`groupname`*	Nome del gruppo a cui aggiungere il nome dell'ambiente. Da utilizzare solo con ambienti Compose.

`-m "`*`version_description`*`"`
or
`--message "`*`version_description`*`"`
	La descrizione della versione dell'applicazione, racchiusa tra virgolette.
	Tipo: String

Nome	Descrizione
`--modules` *`component-a component-b`*	Elenco dei componenti da aggiornare. Da utilizzare solo con ambienti Compose.

`-p`
 	
oppure
 	
`--process`
	Pre-elabora e convalida l'ambiente il manifest e i file di configura zione dell'ambiente nel bundle di origine. La convalida dei file di configurazione è in grado di identificare i problemi prima di distribuire la versione dell'applicazione in un ambiente.

`--source codecommi t/` *`repository-name/branch-name`*	CodeCommit repository e branch.
`--staged`	Distribuisci i file per fasi nell'indice git anziché nel commit HEAD.
`--timeout` *`minutes`*	Il numero di minuti prima che il comando scada.
`--version` *`version_l abel`*	Una versione esistente dell'applicazione da distribuire. Tipo: string
Opzioni comuni	

Output

In caso di esito positivo, il comando restituisce lo stato dell'operazione `deploy`.

Se hai abilitato il CodeBuild supporto nell'applicazione, eb deploy visualizza le informazioni relative alla creazione del codice. CodeBuild Per informazioni sul CodeBuild supporto in Elastic Beanstalk, consulta. Utilizzo dell'EB CLI con AWS CodeBuild

Esempio

L'esempio seguente distribuisce l'applicazione corrente.

```
$ eb deploy
2018-07-11 21:05:22    INFO: Environment update is starting.
```

```
2018-07-11 21:05:27    INFO: Deploying new version to instance(s).
2018-07-11 21:05:53    INFO: New application version was deployed to running EC2
 instances.
2018-07-11 21:05:53    INFO: Environment update completed successfully.
```

eb events

Description

Restituisce gli eventi più recenti dell'ambiente.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando restituisce anche gli eventi più recenti dell'ambiente del builder.

Sintassi

eb events

eb events ***environment-name***

Opzioni

Nome	Descrizione

`-f`
oppure
`--follow`
	Streaming di eventi. Per annullare, premi CTRL+C.

Output

Se eseguito correttamente, il comando restituisce eventi recenti.

Esempio

L'esempio seguente restituisce gli eventi più recenti.

```
$ eb events
2014-10-29 21:55:39     INFO    createEnvironment is starting.
2014-10-29 21:55:40     INFO    Using elasticbeanstalk-us-west-2-111122223333 as Amazon
 S3 storage bucket for environment data.
2014-10-29 21:55:57     INFO    Created load balancer named: awseb-e-r-AWSEBLoa-
NSKUOK5X6Z9J
2014-10-29 21:56:16     INFO    Created security group named: awseb-e-rxgrhjr9bx-stack-
AWSEBSecurityGroup-1UUHU5LZ20ZY7
2014-10-29 21:57:18     INFO    Waiting for EC2 instances to launch. This may take a
 few minutes.
2014-10-29 21:57:18     INFO    Created Auto Scaling group named: awseb-e-rxgrhjr9bx-
stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD
2014-10-29 21:57:22     INFO    Created Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:2cced9e6-859b-421a-
be63-8ab34771155a:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleUpPolicy-1I2ZSNVU4APRY
2014-10-29 21:57:22     INFO    Created Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:1f08b863-
bf65-415a-b584-b7fa3a69a0d5:autoScalingGroupName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingGroup-1TE320ZCJ9RPD:policyName/awseb-e-rxgrhjr9bx-stack-
AWSEBAutoScalingScaleDownPolicy-1E3G7PZKZPSOG
2014-10-29 21:57:25     INFO    Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmLow-VF5EJ549FZBL
2014-10-29 21:57:25     INFO    Created CloudWatch alarm named: awseb-e-rxgrhjr9bx-
stack-AWSEBCloudwatchAlarmHigh-LA9YEW3O6WJO
2014-10-29 21:58:50     INFO    Added EC2 instance 'i-c7ee492d' to Auto ScalingGroup
 'awseb-e-rxgrhjr9bx-stack-AWSEBAutoScalingGroup-1TE320ZCJ9RPD'.
2014-10-29 21:58:53     INFO    Successfully launched environment: tmp-dev
2014-10-29 21:59:14     INFO    Environment health has been set to GREEN
2014-10-29 21:59:43     INFO    Adding instance 'i-c7ee492d' to your environment.
```

eb health

Description

Restituisce lo stato più recente dell'ambiente.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando restituisce anche i controlli dello stato più recenti per l'ambiente generatore.

Sintassi

eb health

eb health ***environment-name***

Opzioni

Nome	Descrizione

`-r`
oppure
`--refresh`
	Mostra informazioni sullo stato interattivamente e si aggiorna ogni dieci secondi quando viene segnalata una nuova informazione.

`--mono`	Non mostrare il colore nell'output.

Output

Se eseguito correttamente, il comando restituisce lo stato recente.

Esempio

L'esempio seguente restituisce le informazioni sullo stato più recenti per un ambiente Linux.

```
~/project $ eb health
 elasticBeanstalkExa-env                                    Ok
 2015-07-08 23:13:20
WebServer
 Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0

  instance-id   status     cause
                                          health
    Overall     Ok
  i-d581497d    Ok
  i-d481497c    Ok
  i-136e00c0    Ok
  i-126e00c1    Ok
```

```
 i-8b2cf575    Ok

 instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75     p50
  p10                                  requests
   Overall     671.8   100.0    0.0    0.0    0.0    0.003    0.002    0.001   0.001
0.000
 i-d581497d    143.0    1430      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-d481497c    128.8    1288      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-136e00c0    125.4    1254      0      0      0    0.004    0.002    0.001   0.001
0.000
 i-126e00c1    133.4    1334      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-8b2cf575    141.2    1412      0      0      0    0.003    0.002    0.001   0.001
0.000

 instance-id   type       az   running     load 1  load 5      user%  nice%  system%
idle%   iowait%                              cpu
 i-d581497d    t2.micro   1a   12 mins        0.0    0.04        6.2    0.0      1.0
92.5       0.1
 i-d481497c    t2.micro   1a   12 mins       0.01    0.09        5.9    0.0      1.6
92.4       0.1
 i-136e00c0    t2.micro   1b   12 mins       0.15    0.07        5.5    0.0      0.9
93.2       0.0
 i-126e00c1    t2.micro   1b   12 mins       0.17    0.14        5.7    0.0      1.4
92.7       0.1
 i-8b2cf575    t2.micro   1c   1 hour        0.19    0.08        6.5    0.0      1.2
92.1       0.1

 instance-id   status     id   version              ago
                                   deployments
 i-d581497d    Deployed   1    Sample Application   12 mins
 i-d481497c    Deployed   1    Sample Application   12 mins
 i-136e00c0    Deployed   1    Sample Application   12 mins
 i-126e00c1    Deployed   1    Sample Application   12 mins
 i-8b2cf575    Deployed   1    Sample Application   1 hour
```

eb init

Description

Imposta i valori predefiniti per le applicazioni Elastic Beanstalk create con la CLI EB ponendo una serie di domande.

> **Note**
>
> I valori impostati eb init si applicano alla directory e al repository correnti sul computer corrente.
> Il comando crea un'applicazione Elastic Beanstalk nel tuo account. Per creare un ambiente Elastic Beanstalk, esegui eb create dopo l'esecuzione di eb init.

Sintassi

eb init

eb init *application-name*

Opzioni

Se esegui eb init senza specificare l'opzione `--platform`, la CLI EB richiede di immettere un valore per ogni impostazione.

> **Note**
>
> Per utilizzare eb init per creare una nuova coppia di chiavi, `ssh-keygen` deve essere installato nel computer locale e disponibile dalla riga di comando.

Nome	Descrizione	
`-i` `--interactive`	Forza l'immissione di un valore per ogni opzione di comando eb init nella CLI EB.	

Nome	Descrizione	
	Note Il comando `init` richiede di specificare valori per le opzioni di comando eb init prive di valore (predefinito). Dopo la prima volta che esegui il comando eb init in una directory, la CLI EB potrebbe non richiedere di specificare alcuna opzione di comando. Di conseguenza, utilizza l'opzione `--interactive` quando desideri modificare un'impostazione precedentemente impostata.	
`-k` *`keyname`* `--keyname` *`keyname`*	Il nome della coppia di EC2 chiavi Amazon da utilizzare con il client Secure Shell (SSH) per accedere in modo sicuro alle EC2 istanze Amazon che eseguono l'applicazione Elastic Beanstalk.	
`--modules` *`folder-1 folder-2`*	Elenco di directory figlio da inizializzare. Da utilizzare solo con ambienti Compose.	

Nome	Descrizione	
`-p` *`platform-version`* `--platform` *`platform-version`*	La versione della piattaforma da utilizzare. Puoi specificare una piattaforma, una piattaforma e una versione della piattaforma, un nome dello stack di soluzioni o un ARN dello stack di soluzioni. Ad esempio: • `php`, `PHP`, `node.js`: la versione più recente per la piattaforma specificata • `php-7.2`, `"PHP 7.2"`: la versione della piattaforma PHP 7.2 consigliata (tipicamente la più recente) • `"PHP 7.2 running on 64bit Amazon Linux"`: la versione della piattaforma PHP consiglia ta (tipicamente la più recente) in questo ramo della piattaforma • `"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"`: la versione della piattafor ma PHP specificata da questo nome dello stack di soluzioni • `"arn:aws:elasticbeanstalk:us-east-2: :platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"` : la versione della piattaforma PHP specificata da questo ARN dello stack di soluzioni Utilizza `eb platform list` per ottenere un elenco delle configurazioni disponibili. Specifica l'opzione `--platform` per ignorare la configurazione interattiva. **Note** Quando si specifica questa opzione, la CLI EB non richiede i valori per le altre opzioni. Al	

Nome	Descrizione	
	contrario, desume i valori predefiniti per ogni opzione. Puoi specificare opzioni per qualsiasi operazione per la quale non desideri utilizzare valori predefiniti.	
`--source codecommit/ repository-name/branch-name`	CodeCommit repository e filiale.	
`--tags key1=value1`	Applica un tag all'applicazione. I tag sono specificati come elenco di coppie `key=value` separate da virgole. Per ulteriori dettagli, consultare Tagging delle applicazioni.	
Opzioni comuni		

CodeBuild supporto

Se esegui eb init in una cartella che contiene un file buildspec.yml, Elastic Beanstalk analizza il file per individuare una voce eb_codebuild_settings con le opzioni specifiche di Elastic Beanstalk. Per informazioni sul CodeBuild supporto in Elastic Beanstalk, consulta. Utilizzo dell'EB CLI con AWS CodeBuild

Output

Se ha esito positivo, il comando guida l'utente attraverso la configurazione di una nuova applicazione Elastic Beanstalk mediante una serie di richieste.

Esempio

L'esempio di richiesta seguente inizializza la CLI EB e richiede di immettere informazioni sull'applicazione. Sostituisci il *`placeholder`* testo con i tuoi valori.

```
$ eb init -i
```

```
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : Europe (Ireland)
5) eu-central-1 : Europe (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
...
(default is 3): 3

Select an application to use
1) HelloWorldApp
2) NewApp
3) [ Create new Application ]
(default is 3): 3

Enter Application Name
(default is "tmp"):
Application tmp has been created.

It appears you are using PHP. Is this correct?
(y/n): y

Select a platform branch.
1) PHP 7.2 running on 64bit Amazon Linux
2) PHP 7.1 running on 64bit Amazon Linux (Deprecated)
3) PHP 7.0 running on 64bit Amazon Linux (Deprecated)
4) PHP 5.6 running on 64bit Amazon Linux (Deprecated)
5) PHP 5.5 running on 64bit Amazon Linux (Deprecated)
6) PHP 5.4 running on 64bit Amazon Linux (Deprecated)
(default is 1): 1
Do you want to set up SSH for your instances?
(y/n): y

Select a keypair.
1) aws-eb
2) [ Create new KeyPair ]
(default is 2): 1
```

eb labs

Description

Sottocomandi di eb labs supporto work-in-progress o funzionalità sperimentali. Questi comandi potrebbero essere rimossi o rielaborati nelle versioni future della CLI EB e non è garantita la compatibilità.

Per un elenco dei sottocomandi disponibili e delle descrizioni, esegui eb labs --help.

eb list

Description

Elenca tutti gli ambienti nell'applicazione corrente o tutti gli ambienti in tutte le applicazioni, come specificato dall'opzione `--all`.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando elenca anche gli ambienti del generatore.

Sintassi

eb list

Opzioni

Nome	Descrizione

`-a`
oppure
`--all`
	Elenca tutti gli ambienti da tutte le applicazioni.

`-v`
oppure
`--verbose`
	Fornisce ulteriori informazioni dettagliate su tutti gli ambienti, incluse le istanze.

Nome	Descrizione
Opzioni comuni	

Output

Se riesce, il comando restituisce un elenco di nomi di ambiente in cui l'ambiente corrente è contrassegnato con un asterisco (*).

Esempio 1

L'esempio seguente elenca i tuoi ambienti e indica che tmp-dev è il tuo ambiente predefinito.

```
$ eb list
* tmp-dev
```

Esempio 2

L'esempio seguente elenca i tuoi ambienti con ulteriori dettagli.

```
$ eb list --verbose
Region: us-west-2
Application: tmp
    Environments: 1
        * tmp-dev : ['i-c7ee492d']
```

eb local

Description

Utilizza eb local run per eseguire i container della tua applicazione in locale in Docker. Verifica lo stato del container dell'applicazione con eb local status. Apri l'applicazione in un browser Web con eb local open. Recupera il percorso dei log dell'applicazione con eb local logs.

eb local setenv e eb local printenv consentono di impostare e visualizzare variabili di ambiente che vengono fornite ai contenitori Docker eseguiti in locale con eb local run.

È necessario eseguire tutti i comandi eb local nella directory del progetto di un'applicazione Docker inizializzata come repository CLI EB utilizzando eb init.

> **Note**
>
> Utilizza eb local su un computer locale con Linux o macOS in esecuzione. Il comando non supporta Windows.
> Prima di utilizzare il comando su macOS, installa Docker per Mac e assicurati che boot2docker non sia installato (o non si trovi nel percorso di esecuzione). Il comando eb local cerca di utilizzare boot2docker se è presente, ma non funziona su macOS.

Sintassi

eb local run

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Opzioni

eb local run

Nome	Descrizione
`--envvars` *`key1=value1,key2=value2`*	Imposta variabili di ambiente che la CLI EB passerà ai container Docker locali. Negli ambienti multicontainer, tutte le variabili vengono trasmesse a tutti i container.
`--port` *`hostport`*	Mappa una porta dell'host alla porta esposta sul container. Se non specifichi questa opzione, la CLI EB utilizza la stessa porta sull'host e sul container. Questa opzione funziona solo con le applicazioni della piattaforma Docker. Non si applica alla piattaforma Docker Multicontainer.

Nome	Descrizione
Opzioni comuni	

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Nome	Descrizione
Opzioni comuni	

Output

eb local run

Messaggi di stato da Docker. Rimane attivo fino a quando l'applicazione è in esecuzione. Premi Ctrl +C per arrestare l'applicazione.

eb local status

Lo stato di ogni container utilizzato dall'applicazione, in esecuzione o meno.

eb local open

Apre l'applicazione in un browser Web ed esce.

eb local logs

La posizione dei log generati nella directory del tuo progetto dalle applicazioni in esecuzione in locale in eb local run.

eb local setenv

Nessuna

eb local printenv

Il nome e i valori di variabili di ambiente impostate con eb local setenv.

Esempi

eb local run

```
~/project$ eb local run
Creating elasticbeanstalk_phpapp_1...
Creating elasticbeanstalk_nginxproxy_1...
Attaching to elasticbeanstalk_phpapp_1, elasticbeanstalk_nginxproxy_1
phpapp_1     | [23-Apr-2015 23:24:25] NOTICE: fpm is running, pid 1
phpapp_1     | [23-Apr-2015 23:24:25] NOTICE: ready to handle connections
```

eb local status

Visualizza lo stato dei tuoi container locali:

```
~/project$ eb local status
Platform: 64bit Amazon Linux 2014.09 v1.2.1 running Multi-container Docker 1.3.3
 (Generic)
Container name: elasticbeanstalk_nginxproxy_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): 80
Full local URL(s): 127.0.0.1:80

Container name: elasticbeanstalk_phpapp_1
Container ip: 127.0.0.1
Container running: True
Exposed host port(s): None
Full local URL(s): None
```

eb local logs

Visualizza il percorso dei log per il progetto corrente:

```
~/project$ eb local logs
Elastic Beanstalk will write logs locally to /home/user/project/.elasticbeanstalk/logs/
local.
```

```
Logs were most recently created 3 minutes ago and written to /home/user/
project/.elasticbeanstalk/logs/local/150420_234011665784.
```

eb local setenv

Imposta variabili di ambiente per l'uso con eb local run.

```
~/project$ eb local setenv PARAM1=value
```

Stampa le variabili di ambiente impostate con eb local setenv.

```
~/project$ eb local printenv
Environment Variables:
PARAM1=value
```

eb logs

Description

Il eb logs comando ha tre scopi distinti: abilitare o disabilitare lo streaming dei log su CloudWatch Logs, recuperare i log delle istanze o i log dei CloudWatch Logs e richiedere l'analisi basata sull'intelligenza artificiale dell'ambiente. Con l'opzione `--cloudwatch-logs` (`-cw`), il comando abilita o disabilita lo streaming dei log. Con l'opzione `--analyze` (`-ai`), il comando richiede un'analisi basata sull'intelligenza artificiale dei log, degli eventi e dello stato dell'istanza dell'ambiente. Senza nessuna di queste opzioni, recupera i log.

Quando recuperi i log, specifica l'opzione `--all`, `--zip` o `--stream` per recuperare i log completi. Se non specifichi alcuna di queste opzioni, Elastic Beanstalk recupera i log di coda.

Il comando elabora i log per l'ambiente specificato o predefinito. I log rilevanti possono variare a seconda del tipo di container. Se la directory root contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando elabora anche i log per l'ambiente generatore.

Per ulteriori informazioni, consulta the section called “CloudWatch Registri”. Per ulteriori informazioni sull'analisi basata sull'intelligenza artificiale, consulta. AI-powered analisi dell'ambiente

Sintassi

Per abilitare o disabilitare lo streaming dei log su Logs: CloudWatch

```
eb logs --cloudwatch-logs [enable | disable] [--cloudwatch-log-source instance |
 environment-health | all] [environment-name]
```

Per recuperare i log delle istanza:

```
eb logs [-all | --zip | --stream] [--cloudwatch-log-source instance] [--
instance instance-id] [--log-group log-group] [environment-name]
```

Per recuperare i log sullo stato dell'ambiente:

```
eb logs [-all | --zip | --stream] --cloudwatch-log-source environment-health
 [environment-name]
```

Per richiedere un'analisi basata sull'intelligenza artificiale:

```
eb logs --analyze [environment-name]
```

Opzioni

Nome	Descrizione

`-cw [enable | disable]`
 	
or
 	
`--cloudwatch-logs [enable | disable]`
	Abilita o disabilita lo streaming dei log su Logs. CloudWatch Se non viene specificato alcun argomento, lo streaming dei log è abilitato. Se l'opzione `--cloudwatch-log-source` (`-cls`) non è specificata come opzione aggiuntiva, lo streaming dei log delle istanze è abilitato o disabilitato.

`-cls instance | environment-health | all`
 	
or
 	
`--cloudwatch-log-source instance | environment-health | all`
	Specifica l'origine dei log quando si lavora con Logs. CloudWatch Con il modulo di attivazione o disattivazione del comando, questi sono i log per i quali abilitare o disabilitare lo streaming dei registri. CloudWatch Con il modulo di recupero del comando, questi sono i log da recuperare dai registri. CloudWatch
	
	Valori validi:
	
	• Con `--cloudwatch-logs` (enable o disable) – `instance | environment-health | all`

Nome	Descrizione
	• Senza `--cloudwatch-logs` (retrieve) – `instance` | `environment-health` Significato dei valori: • `instance` (predefinito): log delle istanze • `environment-health` : log sullo stato dell'ambiente (supportati solo quando nell'ambiente sono abilitate le informazioni sullo stato avanzate) • `all`: entrambe le origini di log

`-a`
 	
oppure
 	
`--all`
	Recupera i log completi e li salva nella directory `.elasticbeanstalk/logs` .

`-z`
 	
oppure
 	
`--zip`
	Recupera i log completi, li comprime in un file `.zip` e quindi salva il file nella directory `.elasticbeanstalk/logs` .

`--stream`	Esegue lo streaming dei log completi (restituendoli continuamente). Con questa opzione, il comando continua a essere eseguito finché non lo interrompi (premendo **`Ctrl+C`**).
`-i` *`instance-id`* oppure `--instance` *`instance-id`*	Recupera i log solo per l'istanza specificata.

Nome	Descrizione
`-g` *`log-group`* or `--log-group` *`log-group`*	Speciifica il gruppo di CloudWatch log Logs da cui recuperare i log. L'opzione è valida solo quando è abilitato lo streaming dei registri delle istanze su Logs. CloudWatch Se lo streaming dei log delle istanze è abilitato e non specifichi l'opzione `--log-group` , il gruppo di log predefinito è uno dei seguenti: • Amazon Linux 2 – `/aws/elasticbeanstalk/` *`environment-name`* `/var/log/eb-engine.log` • Piattaforme Window – `/aws/elasticbeanstalk/` *`environment-name`* `/EBDeploy-Log` • AMI Amazon Linux (AL1) — `/aws/elasticbeanstalk/` *`environment-name`* `/var/log/eb-activity.log` **Note** Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2. Per informazioni sul gruppo di log corrispondente a ogni file di log, consulta Come Elastic Beanstalk configura i log CloudWatch.

Nome	Descrizione

`-ai`
 	
or
 	
`--analyze`
	Richiede un'analisi basata sull'intelligenza artificiale dei log, degli eventi e dello stato delle istanze del tuo ambiente. L'analisi utilizza Amazon Bedrock per identificare le cause alla radice e consigliare soluzioni per problemi di salute ambientale. Il comando invia la richiesta, attende il completamento dell'analisi e quindi visualizza i risultati.
	
	Questa opzione non è compatibile con`--instance` ,`--all`, `--zip``--log-group` , o`--cloudwatch-logs` .
	
	Per i prerequisiti e le autorizzazioni richieste, vedere. AI-powered analisi dell'ambiente

Opzioni comuni	

Output

Per impostazione predefinita, visualizza i log direttamente nel terminale. Usa un programma di paging per visualizzare l'output. Premi **`Q`** o **`q`** per uscire.

Con `--stream`, mostra i log esistenti nel terminale e continua a essere eseguita. Premi **`Ctrl+C`** per uscire.

Con `--all` e `--zip`, salva i log in file locali e visualizza la posizione dei file.

Con`--analyze`, visualizza l'analisi generata dall'intelligenza artificiale direttamente nel terminale dopo il completamento dell'analisi.

Esempi

L'esempio seguente abilita lo streaming dei registri delle istanze su Logs. CloudWatch

```
$ eb logs -cw enable
Enabling instance log streaming to CloudWatch for your environment
After the environment is updated you can view your logs by following the link:
https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logs:prefix=/aws/
elasticbeanstalk/environment-name/
Printing Status:
```

```
2018-07-11 21:05:20    INFO: Environment update is starting.
2018-07-11 21:05:27    INFO: Updating environment environment-name's configuration
 settings.
2018-07-11 21:06:45    INFO: Successfully deployed new configuration to environment.
```

L'esempio seguente recupera i log delle istanze in un file `.zip`.

```
$ eb logs --zip
Retrieving logs...
Logs were saved to /home/workspace/environment/.elasticbeanstalk/logs/150622_173444.zip
```

L'esempio seguente richiede un'analisi dell'ambiente basata sull'intelligenza artificiale.

```
$ eb logs --analyze
Analyzing environment...
No critical issues detected. Environment appears healthy.

---
Note: This analysis was generated by AI.
```

eb migrate

Description

Migra siti e applicazioni Internet Information Services (IIS) da un server Windows a Elastic Beanstalk. Il comando impacchetta le applicazioni, ne preserva le configurazioni e le distribuisce in un nuovo ambiente Elastic Beanstalk.

Per ulteriori informazioni sulla migrazione dei siti e delle applicazioni IIS, consulta. Migrazione delle applicazioni IIS

> **Note**
>
> Prima di utilizzare questo comando, assicuratevi che il sistema soddisfi i seguenti requisiti:
>
> - Internet Information Services (IIS) versione 7.0 o successiva
> - Web Deploy 3.6 o versione successiva installata
> - Privilegi amministrativi sul server Windows
> - AWS credenziali configurate con le autorizzazioni appropriate

- Il server di origine dispone dell'accesso Internet in uscita ai servizi. AWS

I passaggi seguenti riassumono il processo di migrazione:

1. Scopri i siti IIS e le relative configurazioni.
2. Contenuto e configurazione dell'applicazione del Package.
3. Crea l'ambiente e l'applicazione Elastic Beanstalk.
4. Distribuisci l'applicazione con impostazioni preservate.

Il comando crea elementi di migrazione in una directory strutturata, come illustrato nell'elenco seguente:

```
C:\migration_workspace\
### .\migrations\latest\
    ### upload_target.zip
    ### upload_target\
        ### [SiteName].zip                      # One ZIP per default application of
 IIS site
        ### [SiteName-ApplicationName].zip      # One ZIP per additional application
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
            ### site_installer.ps1              # Site installation scripts
            ### permission_handler.ps1          # Permission management
            ### other helper scripts
```

Utilizzatelo eb migrate cleanup per gestire questi artefatti.

Sintassi

eb migrate ***[options]***

eb migrate explore ***[options]***

eb migrate cleanup ***[options]***

Se eseguito senza argomenti, eb migrate funziona in modalità non interattiva. Per eseguirlo in modalità interattiva, eb migrate --interactive esegui.

Il comando della modalità interattiva richiede le seguenti informazioni:

- Selezione dei siti IIS da migrare
- Nomi di ambienti e applicazioni
- Selezione della versione della piattaforma
- Tipo di istanza e altre opzioni di configurazione

Sottocomandi

esplorare

Il eb migrate explore sottocomando esamina il server IIS ed elenca i siti disponibili.

Utilizzare questo comando per visualizzare le seguenti informazioni:

- Visualizza tutti i siti IIS sul server
- Con`--verbose`, ispeziona la configurazione dettagliata, tra cui:
  - Collegamenti e porte del sito
  - Pool di applicazioni
  - Directory virtuali e relativi percorsi fisici
  - Impostazioni di autenticazione

```
PS C:\migrations_workspace >  eb migrate explore
Default Web Site
Site2
site3
router
```

```
PS C:\migrations_workspace >  eb migrate explore --verbose
1: Default Web Site:
  - Bindings:
    - *:8083:
  - Application '/':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\wwwroot
        - Logon Method: ClearText
```

```
  - Application '/dotnet-6-0':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\AspNetCoreWebApps\CoreWebApp-6-0
        - Logon Method: ClearText
  - Application '/dotnet-8-0':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\AspNetCoreWebApps\CoreWebApp-8-0
        - Logon Method: ClearText
2: Site2:
  - Bindings:
    - *:8081:
...
```

pulizia

Il eb migrate cleanup sottocomando gestisce gli artefatti della migrazione con le seguenti azioni:

- Conservazione della migrazione riuscita più recente in `./migrations/latest`
- Rimozione delle cartelle di migrazione più vecchie
- Manutenzione dei file di configurazione critici

```
PS C:\migrations_workspace >   eb migrate cleanup
Are you sure you would like to cleanup older artifacts within `./migrations/`? (y/N):
```

Consente `--force` di ignorare le richieste di conferma durante la pulizia.

```
PS C:\migrations_workspace >   eb migrate cleanup --force
```

Opzioni

Nessuna di queste opzioni è obbligatoria. Se si esegue eb migrate senza alcuna opzione, l'EB CLI verrà eseguita in modalità non interattiva. Coneb migrate --interactive, l'EB CLI richiede di inserire o selezionare un valore per le impostazioni richieste.

Nome	Descrizione
`-a` *`application-name`* or `--application-name` *`application-name`*	Nome per la nuova applicazione Elastic Beanstalk. Tipo: String Impostazione predefinita: App EBMigrated
`--archive` *`directory-or-zip`*	La directory o il file ZIP contenente il codice sorgente precedentemente generato daeb migrate --archive-only. Utilizzate questa opzione per distribuire un pacchetto di migrazione creato in precedenza. Esempio: `--archive .\migrations\latest\upload_target` o `--archive .\migrations\latest\upload_target.zip`

`-ao`
 	
or
 	
`--archive-only`
	Crea solo la directory di archivio di destinazione senza distribuzione.
	
	La directory risultante può essere distribuita manualmente utilizzando eb migrate l'`archive`opzione oeb deploy.

`-c` *`subdomain-name`* or `--cname` *`subdomain-name`*	Il nome del sottodominio da aggiungere come prefisso alla voce DNS CNAME per l'applicazione migrata. Tipo: String Impostazione predefinita: nome dell'ambiente

`-cf`
 	
or
 	
`--copy-firewall-config`
	Copia la configurazione del firewall del server di origine nella destinazione per tutte le porte HTTP con collegamenti attivi.
	
	Crea le regole del gruppo di sicurezza corrispondenti in. AWS

Nome	Descrizione
`-es` *`snapshot-id`* [*`snapshot-id`* ...] or `--ebs-snapshots` *`snapshot-id`* [*`snapshot-id`* ...]	Elenco separato da virgole di snapshot Amazon EBS da associare IDs all'ambiente. Ad esempio: `--ebs-snapshots snap-1234567890abcdef0, snap-0987654321fedcba1`
`--encrypt-ebs-volumes`	Applica la crittografia per tutti i nuovi volumi Amazon EBS. **Important** Si tratta di un'impostazione a livello di account che influisce su tutte le future creazioni di volumi Amazon EBS.
`-e` *`environment-name`* or `--environment-name` *`environment-name`*	Nome per il nuovo ambiente Elastic Beanstalk. Tipo: String Impostazione predefinita: Env EBMigrated Vincoli: deve avere una lunghezza compresa tra 4 e 40 caratteri. Può contenere solo lettere, numeri e trattini. Non può iniziare o terminare con un trattino (-).
`--force`	Ignora le richieste di conferma durante le operazioni. Se utilizzato con il cleanup sottocomando, rimuove gli elementi della migrazione senza conferma.

Nome	Descrizione
`-ip` *`profile-name`* or `--instance-profile` *`profile-name`*	Profilo dell'istanza da associare alle EC2 istanze Amazon dell'ambiente. Se non specificato, crea un profilo di istanza predefini to con le autorizzazioni per accedere alle risorse Elastic Beanstalk. Per ulteriori informazioni, consulta the section called “Profilo dell’istanza”.
`-i` *`instance-type`* or `--instance-type` *`instance-type`*	Il tipo di EC2 istanza Amazon per il tuo ambiente Elastic Beanstalk. Tipo: String Predefinito: c5.2xlarge Per i tipi di istanze disponibili, consulta i tipi di EC2 istanze Amazon nella Amazon EC2 User Guide.

`-in`
 	
or
 	
`--interactive`
	Forza la modalità interattiva per il processo di migrazione.
	
	Richiede i valori di configurazione anche quando sono disponibili quelli predefiniti.

`-k` *`key-name`* or `--keyname` *`key-name`*	Amazon EC2 key pair per abilitare l'accesso RDP alle istanze di ambiente. Utile per analizzare problemi a livello di istanza non visibili nei log. Valori validi: un nome di key pair esistente registrato con Amazon EC2

Nome	Descrizione
`-p` *`platform-version`* or `--platform` *`platform-version`*	Runtime della piattaforma Elastic Beanstalk per l'ambient e. Se non specificato, viene rilevato automaticamente dalla versione host di Windows Server. Ad esempio: `"64bit Windows Server 2016 v2.16.2 running IIS 10.0"` Per un elenco delle versioni disponibili della piattaforma, usa eb platform list.
`--remote`	Indica di eseguire la migrazione in modalità remota. Questa opzione consente l'esecuzione da un host bastion, che si connette al server di destinazione che contiene l'applicazione e le configurazioni da migrare su Elastic Beanstalk. Viene eseguito dal server bastion, `eb migrate` rileva le configurazioni, imposta la logica di migrazione sull'host bastion, quindi distribuisce l'applica zione in un nuovo ambiente Elastic Beanstalk. Questa opzione elimina la necessità di installare EB CLI e Python sul server Windows da migrare. Invece, installi Python e EB CLI su un host bastion, dove esegui il comando con l'eb migrateopzione. `--remote` Usa l'`--target-ip` opzione per specificare l'host con le configurazioni IIS da migrare. Deve essere usato con `--target-ip` `--username` , e. `--password`
`--target-ip` *`ip-address`*	Indirizzo IP pubblico del computer Windows remoto che contiene i server IIS da migrare. Richiesto durante l'utilizzo`--remote`. Può essere specificato solo durante l'uso`--remote`.

Nome	Descrizione
`--username` *`username`*	Nome utente del profilo utente per accedere alla macchina Windows remota che contiene i server IIS da migrare. Richiesto durante l'utilizzo`--remote`. Può essere specificato solo durante l'uso`--remote`.
`--password` *`password`*	Password del profilo utente per accedere alla macchina Windows remota che contiene i server IIS da migrare. Richiesto durante l'utilizzo`--remote`. Può essere specificato solo durante l'uso`--remote`.
`-sr` *`role-name`* or `--service-role` *`role-name`*	Ruolo di servizio IAM per Elastic Beanstalk per la gestione dei servizi correlati. AWS Se non specificato, crea un ruolo di servizio predefinito con le autorizzazioni necessarie. Per ulteriori informazi oni, consulta the section called “Ruolo del servizio”. **Note** Specificate solo il nome del ruolo, non l'ARN completo. Elastic Beanstalk crea automatic amente l'ARN completo.
`-s` *`site-names`* or `--sites` *`site-names`*	Elenco separato da virgole di siti IIS da migrare. Se non specificato, migra tutti i siti disponibili sul server. Ad esempio: `--sites "Default Web Site,Intr anet,API"`

Nome	Descrizione
`--ssl-certificates` *`certificate-arn`* [,*`certificate-arn`* ...]	Elenco separato da virgole di certificati SSL ACM da associare ARNs all'Application Load Balancer. Richiesto per la migrazione di siti con collegamenti HTTPS. Ad esempio: `--ssl-certificates arn:aws:acm:region:account:certificate/certificate-id`

`-t` *`key1`*`=`*`value1`*`[,`*`key2`*`=`*`value2`* `..`
 	
or
 	
`--tags` *`key1`*`=`*`value1`*`[,`*`key2`*`=`*`va`
	Elenco di `key=value` coppie separate da virgole per etichettare nuove risorse nell'ambiente: ambiente, applicazione Elastic Beanstalk, versione dell'applicazione.
	
	Per ulteriori informazioni, consulta Etichettatura degli ambienti.

`--verbose`	Mostra informazioni dettagliate durante il processo di migrazione. Se utilizzato con il explore sottocomando, visualizza dettagli completi sulla configurazione del sito.

Nome	Descrizione
`-vpc` *`config-file-or-string`* or `--vpc-config` *`config-file-or-string`*	Configurazione VPC per l'ambiente, specificata come percorso di file JSON o stringa JSON. La configurazione deve includere: `{ "id": "vpc-1234567890abcdef0", "publicip": "true|false", "elbscheme": "public|private", "ec2subnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"], "securitygroups": "sg-123456,sg-789012", "elbsubnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"] }` • `id`: (Obbligatorio) identificatore VPC • `publicip`: se assegnare il pubblico alle istanze IPs • `elbscheme` : schema di bilanciamento del carico (pubblico o privato) • `ec2subnets` : elenco di IDs sottoreti per le istanze EC2 • `securitygroups` : gruppo di sicurezza separato da virgole IDs • `elbsubnets` : elenco di sottoreti per il sistema di IDs bilanciamento del carico **Important** La migrazione ignorerà tutte le impostazioni VPC esistenti dall'ambiente di origine quando si specifica il *`--vpc-config`* parametro. Quando utilizzi questo parametro, la migrazione utilizzer à solo le impostazioni VPC specificate nel file di configurazione che stai trasmettendo. L'utilizzo di

Nome	Descrizione
	questo parametro sostituisce il comportamento predefinito di rilevamento della configurazione VPC dell'istanza di origine o dell'utilizzo del VPC predefinito.
Opzioni comuni	

Output

Il comando fornisce aggiornamenti sullo stato durante tutto il processo di migrazione:

1. Rilevamento della configurazione VPC (quando è in esecuzione su un' EC2 istanza)
2. Avanzamento della generazione dei bundle di sorgenti per ogni sito
3. Stato di creazione dell'ambiente
4. Avanzamento dell'implementazione

In caso di successo, visualizza i dettagli del nuovo ambiente, tra cui:

- Nome e ID dell'ambiente
- Application name (Nome applicazione)
- Region
- Versione della piattaforma
- CNAME dell'ambiente

Per problemi durante la migrazione, usa i eb healthcomandi eb eventsand per ottenere informazioni dettagliate.

Esempi

Utilizzo di base

Migrazione di base in modalità interattiva:

```
PS C:\migrations_workspace >  eb migrate
```

```
Identifying VPC configuration of this EC2 instance (i-0123456789abcdef0):
  id: vpc-1234567890abcdef0
  publicip: true
  elbscheme: public
  ec2subnets: subnet-123,subnet-456,subnet-789
  securitygroups: sg-123,sg-456
  elbsubnets: subnet-123,subnet-456,subnet-789

Using .\migrations\latest to contain artifacts for this migration run.
Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip

Creating application version
Creating environment

Environment details for: EBMigratedEnv
  Application name: EBMigratedApp
  Region: us-west-2
  Deployed Version: app-230320_153045
  Environment ID: e-abcdef1234
  Platform: 64bit Windows Server 2019 v2.7.0 running IIS 10.0
  Tier: WebServer-Standard-1.0
  CNAME: ebmigratedenv.us-west-2.elasticbeanstalk.com
  Updated: 2023-03-20 15:30:45
```

Migrazione di siti specifici con configurazione personalizzata:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "Default Web Site,InternalAPI" `
    --application-name "CorporateApp" `
    --environment-name "Production" `
    --instance-type "c5.xlarge" `
    --tags "Environment=Production,Team=WebOps" `
    --copy-firewall-config
```

Creazione di un archivio di migrazione senza distribuzione:

```
PS C:\migrations_workspace >  eb migrate --archive-only
Using .\migrations\latest to contain artifacts for this migration run.
Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip

Generated destination archive directory at .\migrations\latest\upload_target
```

```
You can execute `eb init` and `eb create` from this directory to deploy to EB.
```

Esempi di configurazione avanzata

Migrazione con configurazione VPC personalizzata utilizzando un file JSON:

```
PS C:\migrations_workspace >  cat vpc-config.json
{
    "id": "vpc-1234567890abcdef0",
    "publicip": "false",
    "elbscheme": "internal",
    "ec2subnets": [
        "subnet-private1",
        "subnet-private2"
    ],
    "securitygroups": [
        "sg-app",
        "sg-database",
        "sg-monitoring"
    ],
    "elbsubnets": [
        "subnet-private1",
        "subnet-private2"
    ]
}

PS C:\migrations_workspace eb migrate `
    --sites "InternalAPI" `
    --vpc-config vpc-config.json `
    --instance-type "r5.xlarge" `
    --tags "Environment=Internal,Security=High"
```

Migrazione di siti con certificati SSL e intestazioni host:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "SecurePortal" `
    --ssl-certificates "arn:aws:acm:region:account:certificate/
cert1,arn:aws:acm:region:account:certificate/cert2" `
    --verbose
INFO: Detected HTTPS bindings:
  - www.example.com:443
  - api.example.com:443
```

```
INFO: Configuring Application Load Balancer with SSL certificates
INFO: Creating host-based routing rules:
  - www.example.com -> target group 1
  - api.example.com -> target group 2
```

Migrazione con configurazione snapshot EBS:fo

```
PS C:\migrations_workspace >  eb migrate `
    --sites "Default Web Site" `
    --ebs-snapshots "snap-1234567890abcdef0" "snap-0987654321fedcba1" `
    --encrypt-ebs-volumes
Using .\migrations\latest to contain artifacts for this migration run.
INFO: Enabling EBS encryption for all new volumes in us-west-2
INFO: Configuring environment with specified EBS snapshots
```

Esempi di configurazione della sicurezza

Gestione di siti con regole firewall complesse:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "Default Web Site,ReportingService" `
    --copy-firewall-config `
    --verbose
INFO: Detected the following Windows Firewall rules:
  - Allow Web Traffic (TCP 80, 443)
  - Allow Reporting Traffic (TCP 8081)
INFO: Creating corresponding security group rules
```

Migrazione con ruoli IAM personalizzati:

```
PS C:\migrations_workspace >  eb migrate `
    --sites "SecureApp" `
    --instance-profile "CustomInstanceProfile" `
    --service-role "CustomServiceRole"
```

Esempi di esecuzione remota

Migrazione di applicazioni IIS da un server Windows remoto:

```
PS C:\migrations_workspace >  eb migrate `
    --remote `
```

```
    --target-ip "192.0.2.10" `
    --username "administrator" `
    --password "YourPassword123" `
    --application-name "RemoteApp" `
    --environment-name "RemoteEnv"
INFO: Establishing SSH connection to remote host 192.0.2.10...
INFO: Connection established
INFO: Discovering IIS sites on remote host...
INFO: Found 2 sites: Default Web Site, API
INFO: Extracting site configurations...
INFO: Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip
  API/ -> .\migrations\latest\upload_target\API.zip

Creating application version
Creating environment

Environment details for: RemoteEnv
  Application name: RemoteAppstage mi
  Region: us-west-2
  Deployed Version: app-230320_153045
  Environment ID: e-abcdef1234
  Platform: 64bit Windows Server 2019 v2.7.0 running IIS 10.0
  Tier: WebServer-Standard-1.0
  CNAME: remoteenv.us-west-2.elasticbeanstalk.com
  Updated: 2023-03-20 15:30:45
```

Migrazione remota con selezione specifica del sito:

```
PS C:\migrations_workspace >  eb migrate `
    --remote `
    --target-ip "192.0.2.10" `
    --username "administrator" `
    --password "YourPassword123" `
    --sites "API" `
    --instance-type "c5.large"
```

eb open

Description

Apre l'URL pubblico del tuo sito Web nel browser predefinito.

Sintassi

eb open

eb open ***environment-name***

Opzioni

Nome	Descrizione
Opzioni comuni	

Output

Il comando eb open non produce output. Al contrario, apre l'applicazione in una finestra del browser.

eb platform

Description

Questo comando supporta due diverse aree di lavoro:

Platform (Piattaforma)

Utilizza questo spazio di lavoro per gestire piattaforme personalizzate.

Ambiente

Utilizza questo spazio di lavoro per selezionare una piattaforma di default o mostrare informazioni sulla piattaforma attuale.

Elastic Beanstalk fornisce la scelta rapida ebp per eb platform.

> **Note**
>
> Windows PowerShell utilizza ebp come alias di comando. Se stai eseguendo l'EB CLI in Windows PowerShell, usa la forma lunga di questo comando:. eb platform

Utilizzo della piattaforma eb per piattaforme personalizzate

Elenca le versioni della piattaforma corrente e consente di gestire piattaforme personalizzate.

Sintassi

eb platform create [***version***] [***options***]

eb platform delete [***version***] [***options***]

eb platform events [***version***] [***options***]

eb platform init [***platform***] [***options***]

eb platform list [***options***]

eb platform logs [***version***] [***options***]

eb platform status [***version***] [***options***]

eb platform use [***platform***] [***options***]

Opzioni

Nome	Descrizione
`create [version] [options]`	Crea una nuova versione della piattaforma. Ulteriori informazioni.
`delete version [options]`	Elimina una versione della piattaforma. Ulteriori informazioni.
`events [version] [options]`	Visualizza gli eventi da una versione della piattaforma. Ulteriori informazioni.
`init [platform] [options]`	Inizializza un repository della piattaforma. Ulteriori informazioni.
`list [options]`	Elenca le versioni della piattaforma corrente. Ulteriori informazioni.

Nome	Descrizione
`logs [version] [options]`	Visualizza i log dell'ambiente generatore per una versione della piattaforma. Ulteriori informazioni.
`status [version] [options]`	Visualizza lo stato della versione di una piattaforma. Ulteriori informazioni.
`use [platform] [options]`	Seleziona una piattaforma diversa da cui vengono create nuove versioni. Ulteriori informazioni.
Opzioni comuni	

Opzioni comuni

Tutti i comandi eb platform includono le seguenti opzioni comuni.

Nome	Descrizione

`-h`
O
`--help`
	Mostra un messaggio di aiuto ed esce.

`--debug`	Mostra output di debug aggiuntivo.
`--quiet`	Elimina tutti gli output.

`-v`
O
`--verbose`
	Mostra output aggiuntivo.

`--profile PROFILE`	Utilizza le credenziali *PROFILE* specificate.
`-r REGION` O	Usa la regione. *REGION*

Nome	Descrizione
`--region REGION`	
`--no-verify-ssl`	Non verificate i certificati AWS SSL.

Eb platform create

Consente di creare una nuova versione della piattaforma e restituisce l'ARN per la nuova versione. Se non vi è un ambiente generatore in esecuzione nella regione attuale, questo comando consente di avviarne uno. Le opzioni *version* e increment (`-M-m`, `e-p`) si escludono a vicenda.

Opzioni

Nome	Description
version	Se *version* non è specificato, crea una nuova versione basata sulla piattaforma più recente con la versione della patch (N in n.N.n) incrementata.

`-M`
O
`--major-increment`
	Incrementa il numero di versione maggiore (N in N.n.n).

`-m`
O
`--minor-increment`
	Incrementa il numero di versione minore (N in n.N.n).

`-p`
O
`--patch-increment`
	Incrementa il numero della versione di patch (N in n.n.N).

`-i` *INSTANCE_TYPE* O	Usa *INSTANCE_TYPE* come tipo di istanza, ad esempio. **`t1.micro`**

Nome	Description
--instance-type *INSTANCE_TYPE*	
`-ip` *INSTANCE_PROFILE* O `--instance-profile` *INSTANCE_PROFILE*	*INSTANCE_PROFILE* Utilizzalo come profilo di istanza durante AMIs la creazione per una piattaforma personalizzata. Se l'opzione `-ip` non è specificata, crea il profilo dell'istanza `aws-elasticbeanstalk-custom-platforme-ec2-role` e la utilizza per la piattaforma personalizzata.
`--tags` *key1*=*value1*[,*ke*	I tag sono specificati come elenco di coppie `key=value` separate da virgole.
`--timeout` *minutes*	Imposta il numero di minuti prima del timeout del comando.
`--vpc.id` *VPC_ID*	ID del VPC in cui Packer sviluppa.
`--vpc.subnets` *VPC_SUBNETS*	Le sottoreti del VPC in cui Packer sviluppa.
`--vpc.publicip`	Associa il pubblico IPs alle EC2 istanze lanciate.

Eb platform delete

Elimina una versione della piattaforma. La versione non viene eliminata se un ambiente la sta utilizzando.

Opzioni

Nome	Descrizione
version	La versione da eliminare. Questo valore è obbligatorio.
`--cleanup`	Rimuovi tutte le versioni della piattaforma nello stato `Failed`.
`--all-platforms`	Se viene specificato `--cleanup` , rimuovi tutte le versioni della piattaforma nello stato `Failed` per tutte le piattaforme.

Nome	Descrizione
`--force`	Non richiedere la conferma quando elimini una versione.

Eb platform events

Visualizza gli eventi da una versione della piattaforma. Se *`version`* specificato, visualizza gli eventi di quella versione, altrimenti visualizza gli eventi della versione corrente.

Opzioni

Nome	Description
`version`	La versione per la quale vengono visualizzati gli eventi. Questo valore è obbligatorio.

`-f`
O
`--follow`
	Continua a visualizzare gli eventi mentre si verificano.

Eb platform init

Inizializza un repository della piattaforma.

Opzioni

Nome	Descrizione
`platform`	Nome della piattaforma da inizializzare. Questo valore è obbligatorio, a meno che non sia stata abilitata la `-i` (modalità interattiva).
`-i` O	Utilizza la modalità interattiva.

Nome	Descrizione
`--interactive`	
`-k` *`KEYNAME`* O `--keyname` *`KEYNAME`*	Il nome della EC2 chiave predefinito.

Puoi eseguire questo comando in una directory che è stata precedentemente inizializzata, anche se non puoi modificare il tipo di spazio di lavoro se eseguito in una directory che è stata precedentemente inizializzata.

Per inizializzare di nuovo con opzioni diverse, usa l'opzione `-i`.

Eb platform list

Elenca le versioni della piattaforma associate a un workspace (directory) o a una regione.

Il comando restituisce risultati diversi a seconda del tipo di workspace in esecuzione su di esso, come segue:

- In un workspace di piattaforma (una directory inizializzata da `eb platform init`), il comando restituisce un elenco di tutte le versioni della piattaforma personalizzata definite nel workspace. Aggiungi l'opzione `--all-platforms` o `--verbose` per ottenere un elenco di tutte le versioni di tutte le piattaforme personalizzate di cui il tuo account dispone nella regione associata al workspace.
- In un workspace di applicazione (una directory inizializzata da `eb init`), il comando restituisce un elenco di tutte le versioni di piattaforma, sia per le piattaforme gestite da Elastic Beanstalk, sia per quelle personalizzate del tuo account. L'elenco utilizza nomi di versione della piattaforma abbreviati e alcune varianti di versione della piattaforma potrebbero essere combinate. Aggiungi l'opzione `--verbose` per ottenere un elenco dei nomi completi e di tutte le varianti elencate separatamente.
- In una directory non inizializzata, il comando funziona solo con l'opzione `--region`. Il comando restituisce un elenco di tutte le versioni della piattaforma gestita da Elastic Beanstalk supportate nella regione. L'elenco utilizza nomi di versione della piattaforma abbreviati e alcune varianti di versione della piattaforma potrebbero essere combinate. Aggiungi l'opzione `--verbose` per ottenere un elenco dei nomi completi e di tutte le varianti elencate separatamente.

Opzioni

Nome	Descrizione

`-a`
O
`--all-platforms`
	Valido solo in un workspace inizializzato (una directory inizializzata da `eb platform init` o `eb init`). Elenca le versioni di tutte le piattaforme personalizzate associate all'account.

`-s` *`STATUS`* O `--status` *`STATUS`*	Elenca solo le piattaforme corrispondenti*`STATUS`*: • Pronto • Failed (Non riuscito) • Deleting (Eliminazione in corso) • Creazione

Eb platform logs

Visualizza i log dell'ambiente generatore per una versione della piattaforma.

Opzioni

Nome	Descrizione
`version`	La versione della piattaforma per la quale vengono visualizzati i log. Se assente, visualizza i log della versione attuale.
`--stream`	Trasmetti in streaming i log di distribuzione che sono stati configurati con CloudWatch.

Eb platform status

Visualizza lo stato della versione di una piattaforma.

Opzioni

Nome	Descrizione
version	La versione della piattaforma per cui è stato recuperato lo stato. Se assente, visualizza lo stato della versione attuale.

Eb platform use

Seleziona una piattaforma diversa da cui vengono create nuove versioni.

Opzioni

Nome	Description
platform	Specificato *platform* come versione attiva per questo spazio di lavoro. Questo valore è obbligatorio.

Utilizzo di piattaforme eb per gli ambienti

Elenca piattaforme supportate e consente di impostare la piattaforma di default e la versione della piattaforma da utilizzare all'avvio di un ambiente. Utilizza eb platform list per visualizzare un elenco di tutte le piattaforme supportate. Utilizza eb platform select per modificare la piattaforma per il tuo progetto. Utilizza eb platform show per visualizzare la piattaforma selezionata del progetto.

Sintassi

eb platform list

eb platform select

eb platform show

Opzioni

Nome	Descrizione
`list`	Elenca la versione della piattaforma corrente.

Nome	Descrizione
`select`	Seleziona la piattaforma di default.
`show`	Mostra informazioni sulla piattaforma corrente.

Esempio 1

Nell'esempio seguente sono elencati i nomi di tutte le configurazioni per tutte le piattaforme supportate da Elastic Beanstalk.

```
$ eb platform list
docker-1.5.0
glassfish-4.0-java-7-(preconfigured-docker)
glassfish-4.1-java-8-(preconfigured-docker)
go-1.3-(preconfigured-docker)
go-1.4-(preconfigured-docker)
iis-7.5
iis-8
iis-8.5
multi-container-docker-1.3.3-(generic)
node.js
php-5.3
php-5.4
php-5.5
python
python-2.7
python-3.4
python-3.4-(preconfigured-docker)
ruby-1.9.3
ruby-2.0-(passenger-standalone)
ruby-2.0-(puma)
ruby-2.1-(passenger-standalone)
ruby-2.1-(puma)
ruby-2.2-(passenger-standalone)
ruby-2.2-(puma)
tomcat-6
tomcat-7
tomcat-7-java-6
tomcat-7-java-7
tomcat-8-java-8
```

Esempio 2

L'esempio seguente ti richiede di scegliere da un elenco di piattaforme e di selezionare la versione che desideri distribuire per la piattaforma specificata.

```
$ eb platform select
Select a platform.
1) PHP
2) Node.js
3) IIS
4) Tomcat
5) Python
6) Ruby
7) Docker
8) Multi-container Docker
9) GlassFish
10) Go
(default is 1): 5

Select a platform version.
1) Python 2.7
2) Python
3) Python 3.4 (Preconfigured - Docker)
```

Esempio 3

L'esempio seguente mostra le informazioni sulla piattaforma predefinita attuale.

```
$ eb platform show
Current default platform: Python 2.7
New environments will be running:  64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

Platform info for environment "tmp-dev":
Current: 64bit Amazon Linux 2014.09 v1.2.0 running Python
Latest:  64bit Amazon Linux 2014.09 v1.2.0 running Python
```

eb printenv

Description

Stampa tutte le proprietà dell'ambiente nella finestra di comando.

Sintassi

eb printenv

eb printenv ***environment-name***

Opzioni

Nome	Descrizione
Opzioni comuni	

Output

In caso di esito positivo, il comando restituisce lo stato dell'operazione `printenv`.

Esempio

L'esempio seguente mostra le proprietà dell'ambiente per l'ambiente specificato.

```
$ eb printenv
Environment Variables:
     PARAM1 = Value1
```

eb restore

Description

Ricostruisci un ambiente terminato creando un nuovo ambiente con lo stesso nome, ID e configurazione. Devi quindi disporre del nome dell'ambiente, del nome di dominio e della versione dell'applicazione per riuscire a ricostruire l'ambiente.

Sintassi

eb restore

eb restore ***environment_id***

Opzioni

Nome	Descrizione
Opzioni comuni	

Output

La CLI EB visualizza un elenco di ambienti terminati che possono essere ripristinati.

Esempio

```
$ eb restore
Select a terminated environment to restore

  #   Name          ID             Application Version      Date Terminated        Ago

  3   gamma         e-s7mimej8e9   app-77e3-161213_211138   2016/12/14 20:32 PST   13
 mins
  2   beta          e-sj28uu2wia   app-77e3-161213_211125   2016/12/14 20:32 PST   13
 mins
  1   alpha         e-gia8mphu6q   app-77e3-161213_211109   2016/12/14 16:21 PST   4
 hours

 (Commands: Quit, Restore, # #)

Selected environment alpha
Application:    scorekeep
Description:    Environment created from the EB CLI using "eb create"
CNAME:          alpha.h23tbtbm92.us-east-2.elasticbeanstalk.com
Version:        app-77e3-161213_211109
Platform:       64bit Amazon Linux 2016.03 v2.1.6 running Java 8
Terminated:     2016/12/14 16:21 PST
Restore this environment? [y/n]: y

2018-07-11 21:04:20    INFO: restoreEnvironment is starting.
2018-07-11 21:04:39    INFO: Created security group named: sg-e2443f72
...
```

eb scale

Description

Scala l'ambiente per eseguire sempre un numero specifico di istanze, impostando il numero minimo e massimo di istanze sul valore specificato.

Sintassi

eb scale ***number-of-instances***

eb scale ***number-of-instances environment-name***

Opzioni

Nome	Descrizione
--timeout	Il numero di minuti prima che il comando scada.
Opzioni comuni	

Output

In caso di successo, il comando aggiorna il numero di istanze minimo e massimo per eseguire il valore specificato.

Esempio

L'esempio seguente imposta il numero di istanze su 2.

```
$ eb scale 2
2018-07-11 21:05:22    INFO: Environment update is starting.
2018-07-11 21:05:27    INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:08:53    INFO: Added EC2 instance 'i-5fce3d53' to Auto Scaling Group
 'awseb-e-2cpfjbra9a-stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E'.
2018-07-11 21:08:58    INFO: Successfully deployed new configuration to environment.
2018-07-11 21:08:59    INFO: Environment update completed successfully.
```

eb setenv

Description

Imposta le proprietà dell'ambiente per l'ambiente predefinito.

Sintassi

eb setenv ***key***=***value***

Puoi includere tutte le proprietà che desideri, ma la dimensione totale di tutte le proprietà non può superare 4096 byte. È possibile eliminare una variabile lasciando il valore vuoto. Per informazioni sui limiti, consulta Configurazione delle proprietà dell'ambiente (variabili di ambiente).

> **Note**
>
> Se il `value` contiene caratteri speciali, è necessario applicare un codice di escape al carattere inserendo prima del carattere stesso una \.

Opzioni

Nome	Descrizione
--timeout	Il numero di minuti prima che il comando scada.
Opzioni comuni	

Output

In caso di successo, il comando mostra che l'aggiornamento dell'ambiente è stato completato.

Esempio

Nell'esempio seguente viene impostata la variabile di ambiente ExampleVar.

```
$ eb setenv ExampleVar=ExampleValue
2018-07-11 21:05:25    INFO: Environment update is starting.
```

```
2018-07-11 21:05:29    INFO: Updating environment tmp-dev's configuration settings.
2018-07-11 21:06:50    INFO: Successfully deployed new configuration to environment.
2018-07-11 21:06:51    INFO: Environment update completed successfully.
```

Il comando seguente imposta diverse proprietà dell'ambiente. Aggiunge la proprietà dell'ambiente denominato `foo` e imposta il valore su `bar`, modificato il valore della proprietà `JDBC_CONNECTION_STRING` ed elimina le proprietà `PARAM4` e `PARAM5`.

```
$ eb setenv foo=bar JDBC_CONNECTION_STRING=hello PARAM4= PARAM5=
```

eb ssh

Description

> **Note**
>
> Questo comando non funziona negli ambienti che eseguono Windows Server Instances.

Connettiti a un' EC2 istanza Amazon Linux nel tuo ambiente utilizzando Secure Shell (SSH). Se un ambiente dispone di più istanze in esecuzione, la CLI EB ti chiede di specificare l'istanza a cui desideri connetterti. Per utilizzare questo comando, è necessario che sia installato SSH sul computer locale e che sia disponibile dalla riga di comando. I file delle chiavi private devono trovarsi in una cartella `.ssh` denominata nella directory utente e le EC2 istanze dell'ambiente devono avere indirizzi IP pubblici.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando si connette anche alle istanze nell'ambiente personalizzato.

> **Chiavi SSH**
>
> Se non hai già configurato SSH, è possibile utilizzare la CLI EB per creare una chiave quando è in esecuzione eb init. Se hai già eseguito eb init, eseguilo nuovamente con l'opzione `--interactive` e seleziona Yes (Sì) e Create New Keypair (Crea nuova coppia di chiavi) quando ti viene richiesto di configurare SSH. Le chiavi create durante il processo saranno salvate nella cartella corretta tramite la CLI EB.

Questo comando apre temporaneamente la porta 22 nel gruppo di sicurezza del tuo ambiente per il traffico in entrata da 0.0.0.0/0 (tutti gli indirizzi IP) se non sono già in vigore regole per la porta 22. Se hai configurato il gruppo di sicurezza del tuo ambiente per aprire la porta 22 a una gamma limitata di CIDR per aumentare la sicurezza, la CLI EB rispetterà tale impostazione e rinuncerà a tutte le modifiche al gruppo di sicurezza. Per sovrascrivere questo comportamento e forzare la CLI EB in modo che apra la porta 22 a tutto il traffico in entrata, utilizzare l'opzione `--force`.

Per ulteriori informazioni su come configurare il gruppo di sicurezza dell'ambiente, consulta EC2 gruppi di sicurezza per ulteriori informazioni su come configurare il gruppo di sicurezza dell'ambiente.

Sintassi

eb ssh

eb ssh ***environment-name***

Opzioni

Nome	Descrizione

`-i`
oppure
`--instance`
	Specifica l'ID istanza dell'istanza a cui connettersi. È consiglia bile utilizzare questa opzione.

`-n`
oppure
`--number`
	Specifica l'istanza a cui connettersi in base al numero.

`-o`
oppure
`--keep_open`
	Lascia la porta 22 aperta nel gruppo di sicurezza dopo la fine della sessione SSH.

`--command`	Esegui un comando shell sull'istanza specificata invece di avviare una sessione SSH.

Nome	Descrizione
`--custom`	Specifica un comando SSH da utilizzare invece di "ssh -i file_chiave". Non includere l'utente remoto e il nome host.
`--setup`	Cambia la coppia di chiavi assegnata alle istanze dell'ambiente (richiede le istanze da sostituire).
`--force`	Apri la porta 22 per il traffico in entrata da 0.0.0.0/0 nel gruppo di sicurezza dell'ambiente, anche se il gruppo di sicurezza è già configurato per SSH. Utilizza questa opzione se il gruppo di sicurezza dell'ambiente è configurato per aprire la porta 22 a un intervallo CIDR limitato che non include l'indirizzo IP da cui stai cercando di collegarti.
`--timeout` *`minutes`*	Imposta il numero di minuti prima del timeout del comando. Può essere utilizzato solo con l'argomento `--setup`.
Opzioni comuni	

Output

In caso di successo, il comando SSH apre una connessione all'istanza.

Esempio

L'esempio seguente esegue la connessione all'ambiente specificato.

```
$ eb ssh
Select an instance to ssh into
1) i-96133799
2) i-5931e053
(default is 1): 1
INFO: Attempting to open port 22.
INFO: SSH port 22 open.
The authenticity of host '54.191.45.125 (54.191.45.125)' can't be established.
RSA key fingerprint is ee:69:62:df:90:f7:63:af:52:7c:80:60:1b:3b:51:a9.
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added '54.191.45.125' (RSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-8-185 ~]$ ls
[ec2-user@ip-172-31-8-185 ~]$ exit
logout
Connection to 54.191.45.125 closed.
INFO: Closed port 22 on ec2 instance security group
```

eb status

Description

Fornisce informazioni sullo stato dell'ambiente.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando restituisce informazioni anche sull'ambiente generatore.

Sintassi

eb status

eb status ***environment-name***

Opzioni

Nome	Descrizione

`-v`
oppure
`--verbose`
	Fornisce ulteriori informazioni sulle singole istanze, ad esempio sul rispettivo stato con il sistema di bilanciamento del carico Elastic Load Balancing.

Nome	Descrizione
Opzioni comuni	

Output

In caso di esito positivo, il comando restituisce le informazioni seguenti sullo stato dell'ambiente:

- Nome ambiente
- Nome applicazione
- Versione dell'applicazione distribuita
- ID ambiente
- Piattaforma
- Livello di ambiente
- CNAME
- Ora dell'ultimo aggiornamento dell'ambiente
- Stato
- Integrità

Se utilizzi la modalità verbosa, EB CLI ti fornisce anche il numero di istanze Amazon EC2 in esecuzione.

Esempio

L'esempio seguente mostra lo stato dell'ambiente tmp-dev.

```
$ eb status
Environment details for: tmp-dev
  Application name: tmp
  Region: us-west-2
  Deployed Version: None
  Environment ID: e-2cpfjbra9a
  Platform: 64bit Amazon Linux 2014.09 v1.0.9 running PHP 5.5
  Tier: WebServer-Standard-1.0
  CNAME: tmp-dev.elasticbeanstalk.com
  Updated: 2014-10-29 21:37:19.050000+00:00
  Status: Launching
```

```
  Health: Grey
```

eb swap

Description

Scambia l'ambiente CNAME con il CNAME di un altro ambiente (ad esempio, per evitare tempi di inattività quando si aggiorna la versione dell'applicazione).

> **Note**
>
> Se sono presenti più di due ambienti, verrà richiesto di selezionare da un elenco di ambienti il nome dell'ambiente che attualmente utilizza il CNAME desiderato. Puoi evitarlo specificando il nome dell'ambiente da utilizzare includendo l'opzione `-n` quando esegui il comando.

Sintassi

eb swap

eb swap ***environment-name***

> **Note**
>
> *`environment-name`*È l'ambiente per il quale desideri un CNAME diverso. Se non si specifica *`environment-name`* come parametro della riga di comando durante l'esecuzioneeb swap, EB CLI aggiorna il CNAME dell'ambiente predefinito.

Opzioni

Nome	Descrizione

`-n`
or
`--destination_name`
	Specificate il nome dell'ambiente con cui desiderate effettuare lo scambio. CNAMEs Se esegui eb swap senza questa opzione, la CLI EB ti chiederà di scegliere da un elenco dei tuoi ambienti.

Nome	Descrizione
Opzioni comuni	

Output

In caso di esito positivo, il comando restituisce lo stato dell'operazione `swap`.

Esempi

L'esempio seguente scambia l'ambiente tmp-dev con live-env.

```
$ eb swap
Select an environment to swap with.
1) staging-dev
2) live-env
(default is 1): 2
2018-07-11 21:05:25    INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:05:26    INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:05:30    INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:05:30    INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
 'live-env'.
```

L'esempio seguente scambia l'ambiente tmp-dev con l'ambiente live-env ma non richiede di immettere o selezionare un valore per le impostazioni.

```
$ eb swap tmp-dev --destination_name live-env
2018-07-11 21:18:12    INFO: swapEnvironmentCNAMEs is starting.
2018-07-11 21:18:13    INFO: Swapping CNAMEs for environments 'tmp-dev' and 'live-env'.
2018-07-11 21:18:17    INFO: 'tmp-dev.elasticbeanstalk.com' now points to 'awseb-e-j-
AWSEBLoa-M7U21VXNLWHN-487871449.us-west-2.elb.amazonaws.com'.
2018-07-11 21:18:17    INFO: Completed swapping CNAMEs for environments 'tmp-dev' and
 'live-env'.
```

eb tags

Description

Aggiungi, elimina, aggiorna ed elenca tag di una risorsa Elastic Beanstalk.

Per ulteriori informazioni sul tagging delle risorse in Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Sintassi

eb tags [***environment-name***] [--resource ***ARN***] -l | --list

eb tags [***environment-name***] [--resource ***ARN***] -a | --add ***key1***=***value1***[,***key2***=***value2*** ...]

eb tags [***environment-name***] [--resource ***ARN***] -u | --update ***key1***=***value1***[,***key2***=***value2*** ...]

eb tags [***environment-name***] [--resource ***ARN***] -d | --delete ***key1***[,***key2*** ...]

È possibile abbinare le opzioni di sottocomando `--add`, `--update` e `--delete` in un unico comando. È richiesto almeno un sottocomando. Non è possibile abbinare una di questi tre opzioni di sottocomando a `--list`.

Senza argomenti aggiuntivi, tutti questi comandi elencano o modificano tag dell'ambiente predefinito nell'applicazione della directory corrente. Con un *`environment-name`* argomento, i comandi elencano o modificano i tag di quell'ambiente. Con l'opzione `--resource`, i comandi elencano o modificano i tag di qualsiasi risorsa Elastic Beanstalk: un'applicazione, un ambiente, una versione dell'applicazione, una configurazione salvata o una versione della piattaforma personalizzata. Specifica la risorsa tramite il suo Amazon Resource Name (ARN).

Opzioni

Nessuna di queste opzioni è obbligatoria. Se esegui eb create senza opzioni, ti viene richiesto di immettere o di selezionare un valore per ogni impostazione.

Nome	Descrizione

`-l`
oppure
`--list`
	Elenca tutti i tag attualmente applicati alla risorsa.

`-a` *`key1`*`=`*`value1`*`[,`*`key2`*`=`*`value2`* oppure	Applica nuovi tag alla risorsa. Specifica i tag come un elenco separato da virgole di coppie `key=value` . Non è possibile specificare chiavi di tag esistenti.

Nome	Descrizione
`--add key1=value1[,key2=val`	Valori validi: consulta Applicazione di tag alle risorse.

`-u key1=value1[,key2=value2`
 	
oppure
 	
`--updat e key1=value1[,key2=value2 .`
	Aggiorna i valori di tag delle risorse esistenti. Specifica i tag come un elenco separato da virgole di coppie `key=value` . Devi specificare le chiavi di tag esistenti.
	
	Valori validi: consulta Applicazione di tag alle risorse.

`-d key1[,key2 ...]`
 	
oppure
 	
`--delete key1[,key2 ...]`
	Elimina i tag delle risorse esistenti. Specifica i tag come un elenco separato da virgole di chiavi. Devi specificare le chiavi di tag esistenti.
	
	Valori validi: consulta Applicazione di tag alle risorse.

`-r region`
 	
or
 	
`--region region`
	Il Regione AWS luogo in cui esiste la tua risorsa.
	
	Impostazione predefinita: la regione di default configurata.
	
	Per l'elenco dei valori che è possibile specificare per questa opzione, consulta Endpoint e quote AWS Elastic Beanstalk nella sezione Riferimenti generali di AWS.

`--resource ARN`	L'ARN della risorsa per la quale il comando modifica o elenca i tag. Se non specificato, il comando fa riferimen to all'ambiente (predefinito o specificato) nell'applicazione della directory corrente. Valori validi: consulta uno degli argomenti secondari di Applicazione di tag alle risorse che è specifico per la risorsa che ti interessa. Questi argomenti mostrano come viene costruito l'ARN della risorsa e spiegano come ottenere un elenco delle risorse esistenti per l'applica zione o l'account. ARNs

Output

L'opzione di sottocomando `--list` visualizza un elenco dei tag della risorsa. L'output mostra i tag che Elastic Beanstalk applica per impostazione predefinita e i tag personalizzati.

```
$ eb tags --list
Showing tags for environment 'MyApp-env':

Key                                 Value

Name                                MyApp-env
elasticbeanstalk:environment-id     e-63cmxwjaut
elasticbeanstalk:environment-name   MyApp-env
mytag                               tagvalue
tag2                                2nd value
```

Le opzioni dei sottocomandi `--add`, `--update` e `--delete`, quando hanno esito positivo, non presentano alcun output. È possibile aggiungere l'opzione `--verbose` per visualizzare l'output dettagliato dell'attività del comando.

```
$ eb tags --verbose --update "mytag=tag value"
Updated Tags:

Key                                 Value

mytag                               tag value
```

Esempi

Il comando seguente aggiunge un tag con la chiave `tag1` e il valore `value1` all'ambiente predefinito dell'applicazione e allo stesso tempo elimina il tag `tag2`.

```
$ eb tags --add tag1=value1 --delete tag2
```

Il comando seguente aggiunge un tag a una configurazione salvata all'interno di un'applicazione.

```
$ eb tags --add tag1=value1 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Il comando seguente ha esito negativo perché cerca di aggiornare un tag non esistente.

```
$ eb tags --update tag3=newval
ERROR: Tags with the following keys can't be updated because they don't exist:

  tag3
```

Il comando seguente ha esito negativo perché cerca di aggiornare ed eliminare la stessa chiave.

```
$ eb tags --update mytag=newval --delete mytag
ERROR: A tag with the key 'mytag' is specified for both '--delete' and '--update'. Each
 tag can be either deleted or updated in a single operation.
```

eb terminate

Description

Termina l'ambiente in esecuzione in modo da non incorrere in addebiti per le risorse non utilizzate. AWS

L'opzione `--all` permette di eliminare l'applicazione su cui è stata inizializzata la directory corrente usando eb init. Il comando termina tutti gli ambienti dell'applicazione. Termina le versioni dell'applicazione e le configurazioni salvate per l'applicazione, quindi elimina l'applicazione.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando termina l'ambiente personalizzato in esecuzione.

> **Note**
>
> Puoi sempre avviare un nuovo ambiente utilizzando la stessa versione in un secondo momento.

Se disponi di dati da un ambiente che desideri conservare, imposta la policy di eliminazione del database su `Retain` prima di terminare l'ambiente. In questo modo, il database rimane operativo al di fuori di Elastic Beanstalk. Dopodiché, qualsiasi ambiente Elastic Beanstalk deve connettersi ad esso come database esterno. Se si desidera eseguire il backup dei dati senza mantenere operativo il database, impostare la policy di eliminazione per fare uno snapshot del database prima di terminare

l'ambiente. Per ulteriori informazioni, consultare Ciclo di vita del database nel capitolo Configurazione degli ambienti di questa guida.

> **Important**
>
> Se un ambiente viene terminato, è necessario anche eliminare eventuali mappature CNAME create, poiché altri clienti possono riutilizzare un nome host disponibile. Assicurati di eliminare i record DNS che puntano al tuo ambiente terminato per evitare una voce DNS inesatta. Una voce DNS inesatta può esporre il traffico Internet destinato al tuo dominio a vulnerabilità di sicurezza. Può presentare anche altri rischi.
> Per ulteriori informazioni, consulta l'argomento relativo alla Protezione dai record di delega inesatti in Route 53 nella Guida per sviluppatori di Amazon Route 53. Puoi anche saperne di più sulle voci DNS sospese in Enhanced Domain Protections for Amazon CloudFront Requests nel Security Blog.AWS

Sintassi

eb terminate

eb terminate ***environment-name***

Opzioni

Nome	Descrizione
`--all`	Termina tutti gli ambienti nell'applicazione, le versioni dell'applicazione e le configurazioni salvate, quindi elimina l'applicazione.
`--force`	Termina l'ambiente senza richiedere conferma.
`--ignore-links`	Termina l'ambiente, anche se non ci sono ambienti dipendenti collegati. Consulta Compose Environments.
`--timeout`	Il numero di minuti prima che il comando scada.

Output

In caso di esito positivo, il comando restituisce lo stato dell'operazione `terminate`.

Esempio

La seguente richiesta di esempio termina l'ambiente tmp-dev.

```
$ eb terminate
The environment "tmp-dev" and all associated instances will be terminated.
To confirm, type the environment name: tmp-dev
2018-07-11 21:05:25    INFO: terminateEnvironment is starting.
2018-07-11 21:05:40    INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-
AWSEBCloudwatchAlarmHigh-16V08YOF2KQ7U
2018-07-11 21:05:41    INFO: Deleted CloudWatch alarm named: awseb-e-2cpfjbra9a-stack-
AWSEBCloudwatchAlarmLow-6ZAWH9F20P7C
2018-07-11 21:06:42    INFO: Deleted Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:5d7d3e6b-
d59b-47c5-b102-3e11fe3047be:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleUpPolicy-1876U27JEC34J
2018-07-11 21:06:43    INFO: Deleted Auto Scaling group policy named:
 arn:aws:autoscaling:us-east-2:11122223333:scalingPolicy:29c6e7c7-7ac8-46fc-91f5-
cfabb65b985b:autoScalingGroupName/awseb-e-2cpfjbra9a-stack-
AWSEBAutoScalingGroup-7AXY7U13ZQ6E:policyName/awseb-e-2cpfjbra9a-stack-AWSEBAutoSca
lingScaleDownPolicy-SL4LHODMOMU
2018-07-11 21:06:48    INFO: Waiting for EC2 instances to terminate. This may take a
 few minutes.
2018-07-11 21:08:55    INFO: Deleted Auto Scaling group named: awseb-e-2cpfjbra9a-
stack-AWSEBAutoScalingGroup-7AXY7U13ZQ6E
2018-07-11 21:09:10    INFO: Deleted security group named: awseb-e-2cpfjbra9a-stack-
AWSEBSecurityGroup-XT4YYGFL7I99
2018-07-11 21:09:40    INFO: Deleted load balancer named: awseb-e-2-AWSEBLoa-
AK6RRYFQVV3S
2018-07-11 21:09:42    INFO: Deleting SNS topic for environment tmp-dev.
2018-07-11 21:09:52    INFO: terminateEnvironment completed successfully.
```

eb upgrade

Description

Aggiorna la piattaforma del tuo ambiente alla versione più recente della piattaforma su cui è attualmente in esecuzione.

Se la directory principale contiene un file `platform.yaml` che specifica una piattaforma personalizzata, questo comando esegue l'aggiornamento dell'ambiente alla versione più recente della piattaforma personalizzata su cui è attualmente in esecuzione.

Sintassi

eb upgrade

eb upgrade ***environment-name***

Opzioni

Nome	Descrizione
`--force`	Aggiorna senza la necessità di confermare il nome dell'ambiente prima di avviare il processo di aggiornamento.
`--noroll`	Aggiorna tutte le istanze senza utilizzare gli aggiornamenti in sequenza per mantenere alcune istanze attive durante l'aggiorn amento.
Opzioni comuni	

Output

Il comando mostra una panoramica delle modifiche e ti chiede di confermare l'aggiornamento digitando il nome dell'ambiente. Se l'operazione va a buon fine, l'ambiente viene aggiornato e quindi avviato con la versione più recente della piattaforma.

Esempio

Nell'esempio riportato qui sotto viene eseguito l'aggiornamento della versione attuale della piattaforma dell'ambiente specificato alla versione più recente disponibile.

```
$ eb upgrade
Current platform: 64bit Amazon Linux 2014.09 v1.0.9 running Python 2.7
Latest platform:  64bit Amazon Linux 2014.09 v1.2.0 running Python 2.7

WARNING: This operation replaces your instances with minimal or zero downtime. You may
 cancel the upgrade after it has started by typing "eb abort".
You can also change your platform version by typing "eb clone" and then "eb swap".

To continue, type the environment name:
```

eb use

Description

Imposta l'ambiente specificato come ambiente predefinito.

Quando usi Git, eb use imposta l'ambiente predefinito per l'attuale ramo. Esegui il comando una volta in ogni ramo che desideri distribuire per Elastic Beanstalk.

Sintassi

eb use ***environment-name***

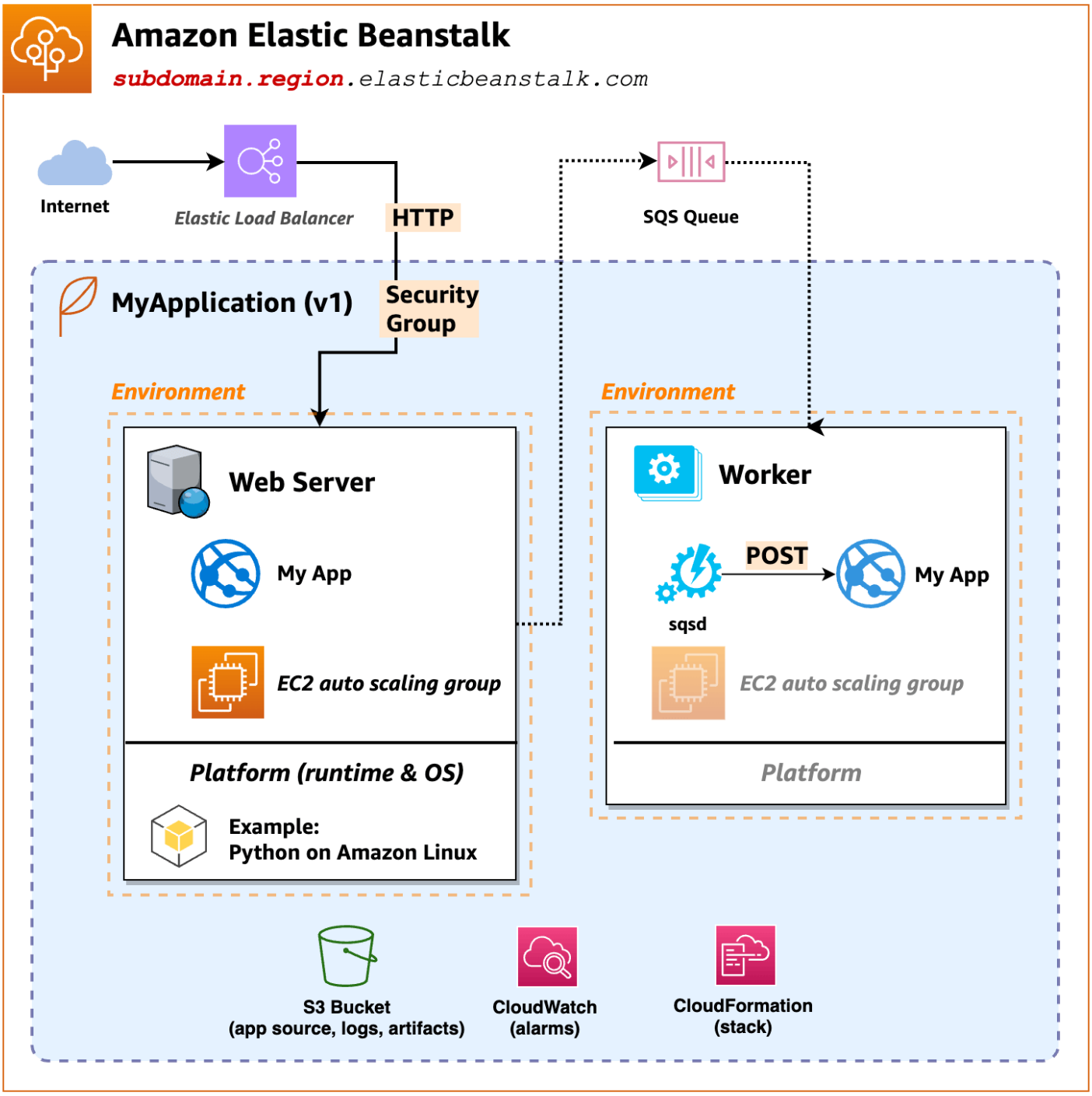
Opzioni

Nome	Description
`--source codecommit/` *`repository-name`*`/`*`branch-name`*	CodeCommit repository e branch.
`-r` *`region`* `--region` *`region`*	Cambia la regione in cui crei gli ambienti.

Nome	Description
Opzioni comuni	

Comprensione dei concetti di Elastic Beanstalk

Acquisire familiarità con i concetti e i termini ti aiuterà ad acquisire le conoscenze necessarie per distribuire le tue applicazioni con Elastic Beanstalk.



Applicazione

Un'applicazione Elastic Beanstalk è un contenitore per componenti Elastic Beanstalk, inclusi ambienti, versioni e configurazioni di ambiente. All'interno di un'applicazione Elastic Beanstalk, gestisci tutte le risorse relative all'esecuzione del codice.

Versione dell'applicazione

In Elastic Beanstalk, una versione dell'applicazione si riferisce a un’iterazione specifica con etichetta di codice distribuibile per un'applicazione Web. Una versione dell'applicazione punta a un oggetto Amazon Simple Storage Service (Amazon S3) contenente il codice distribuibile, ad esempio, un file WAR Java.

La versione dell’applicazione fa parte dell'applicazione. Le applicazioni possono avere molte versioni e ciascuna versione dell’applicazione è unica. In un ambiente in esecuzione, è possibile distribuire qualsiasi versione dell'applicazione già caricata nell’applicazione o è possibile caricare e distribuire immediatamente una nuova versione dell'applicazione. Ad esempio, puoi caricare più versioni dell'applicazione per verificare le differenze tra di esse.

Ambiente

Un ambiente è una raccolta di AWS risorse che eseguono una versione dell'applicazione. Ogni ambiente esegue una sola versione dell'applicazione alla volta. Tuttavia, puoi eseguire la stessa versione dell'applicazione o versioni dell'applicazione diverse in molti ambienti contemporaneamente. Quando crei un ambiente, Elastic Beanstalk fornisce le risorse necessarie AWS nel tuo account per eseguire la versione dell'applicazione specificata.

Livello di ambiente

Quando avvii un ambiente Elastic Beanstalk, devi scegliere prima di tutto un livello di ambiente. Il livello di ambiente indica il tipo di applicazione eseguita dall'ambiente e determina le risorse che Elastic Beanstalk mette a disposizione per supportarla. Un'applicazione che serve richieste HTTP viene eseguita in un livello ambiente di server Web. Un ambiente di back-end che preleva le attività da una coda Amazon Simple Queue Service (Amazon SQS) viene eseguito in un livello ambiente worker.

Configurazione dell'ambiente

La configurazione dell'ambiente identifica una raccolta di parametri e impostazioni che definiscono il comportamento dell'ambiente e delle relative risorse associate. Quando aggiorni le impostazioni di configurazione dell'ambiente, Elastic Beanstalk applica automaticamente le modifiche apportate alle risorse esistenti o elimina e distribuisce nuove risorse (a seconda del tipo di modifica).

Configurazione salvata

Una configurazione salvata è un modello che puoi utilizzare come un punto di partenza per la creazione di configurazioni specifiche di ambienti. Puoi creare e modificare configurazioni salvate e applicarle agli ambienti utilizzando la console Elastic Beanstalk, l'EB AWS CLI CLI o l'API. L'API e il AWS CLI si riferiscono alle configurazioni salvate come modelli di configurazione.

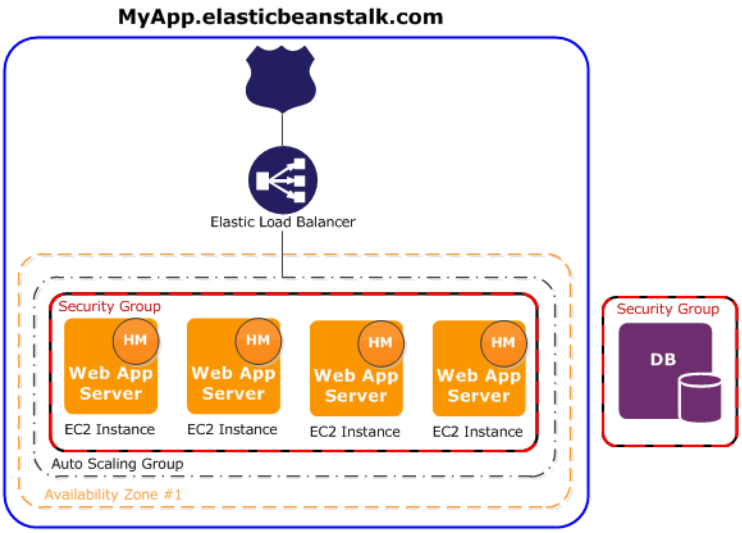
Platform (Piattaforma)

Una piattaforma è una combinazione di sistema operativo, runtime del linguaggio di programmazione, server Web, server applicazioni e componenti Elastic Beanstalk. Sarai tu a progettare e finalizzare l'applicazione Web per una piattaforma specifica. Elastic Beanstalk fornisce una serie di piattaforme su cui puoi creare le applicazioni.

Per informazioni dettagliate, vedi Piattaforme Elastic Beanstalk.

Ambienti server Web Elastic Beanstalk

Il seguente diagramma mostra un esempio di architettura Elastic Beanstalk per un livello di ambiente di server Web e illustra come i componenti in quel tipo di livello di ambiente lavorano insieme.



L'ambiente è il cuore dell'applicazione. Nel diagramma, l'ambiente viene visualizzato all'interno della linea continua di primo livello. Quando crei un ambiente, Elastic Beanstalk fornisce le risorse necessarie per eseguire l'applicazione. AWS le risorse create per un ambiente includono un sistema di bilanciamento del carico elastico (ELB nel diagramma), un gruppo Auto Scaling e una o più istanze Amazon Elastic Compute Cloud (Amazon). EC2

Ogni ambiente dispone di un CNAME (URL) che punta a un sistema di bilanciamento del carico. L'ambiente dispone di un URL, ad esempio `myapp.us-west-2.elasticbeanstalk.com`. Questo URL ha un alias in Amazon Route 53 per un URL di Elastic Load Balancing, simile a `abcdef-123456.us-west-2.elb.amazonaws.com`, utilizzando un record CNAME. Amazon Route 53 è un servizio Web DNS (Domain Name System) altamente scalabile e disponibile. Offre un routing sicuro e affidabile alla tua infrastruttura. Il nome di dominio registrato con il tuo fornitore DNS inoltrerà le richieste al CNAME.

Il sistema di bilanciamento del carico si trova davanti EC2 alle istanze Amazon, che fanno parte di un gruppo Auto Scaling. Amazon EC2 Auto Scaling avvia automaticamente EC2 istanze Amazon aggiuntive per far fronte all'aumento del carico sull'applicazione. Se il carico sull'applicazione diminuisce, Amazon EC2 Auto Scaling interrompe le istanze, ma lascia sempre almeno un'istanza in esecuzione.

Lo stack software in esecuzione sulle EC2 istanze Amazon dipende dal tipo di contenitore. Un tipo di container definisce la topologia dell'infrastruttura e lo stack del software da utilizzare per quell'ambiente. Ad esempio, un ambiente Elastic Beanstalk con un container Apache Tomcat utilizza il sistema operativo Amazon Linux, il server Web Apache e il software Apache Tomcat. Per un elenco di tipi di contenitori supportati, consulta Piattaforme supportate da Elastic Beanstalk. Ogni EC2

istanza Amazon che esegue la tua applicazione utilizza uno di questi tipi di contenitori. Inoltre, un componente software chiamato host manager (HM) viene eseguito su ogni EC2 istanza Amazon. L'host manager è responsabile per le operazioni seguenti:

- Distribuzione dell'applicazione
- Aggregazione di eventi e parametri per il recupero tramite la console, l'API o la riga di comando
- Generazione di eventi a livello di istanza
- Monitoraggio dei file di log dell'applicazione per errori critici
- Monitoraggio del server dell'applicazione
- Applicazione di patch ai componenti delle istanze
- Rotazione dei file di log dell'applicazione e loro pubblicazione su Amazon S3

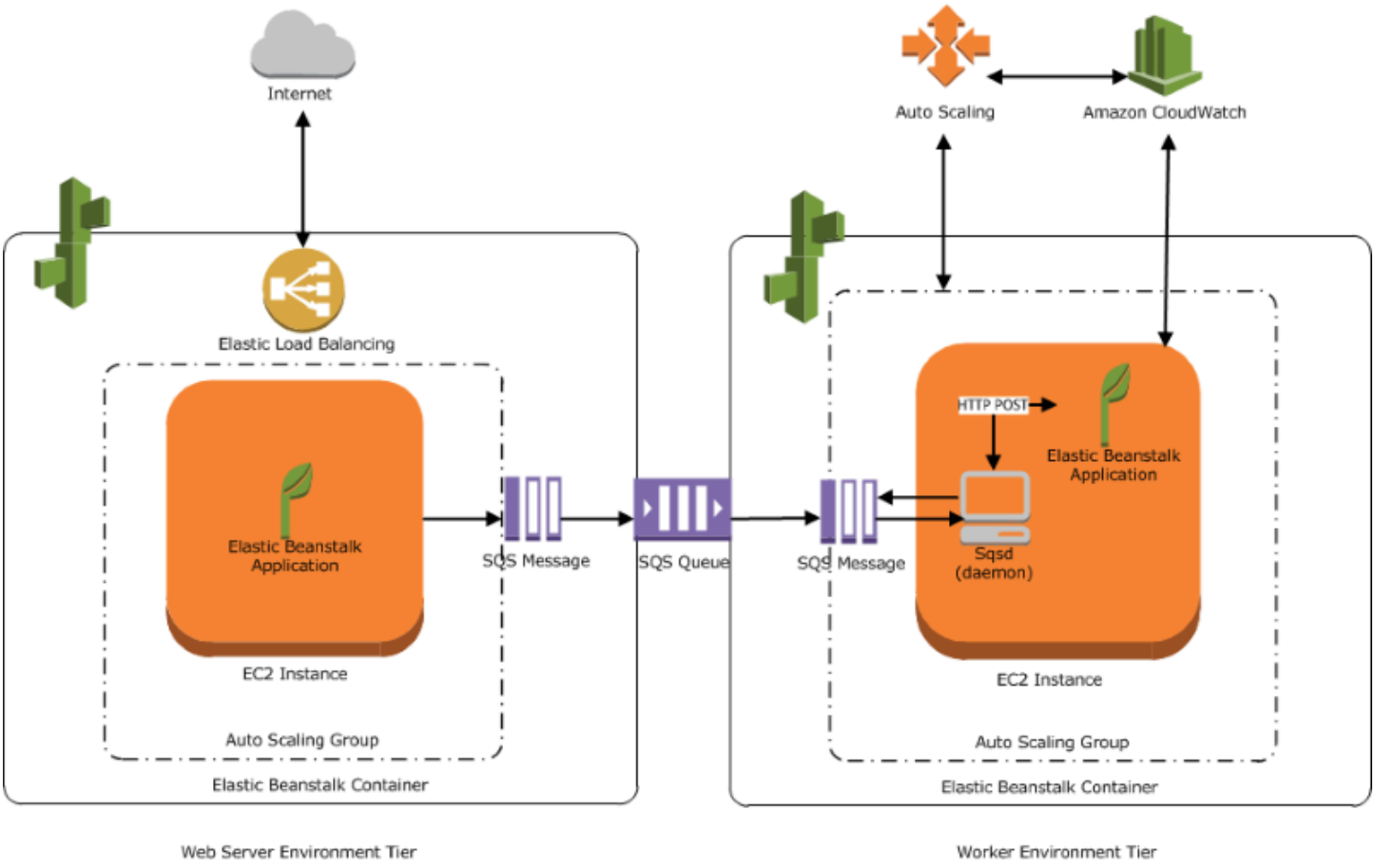
L'host manager riporta metriche, errori ed eventi e lo stato dell'istanza del server, disponibili tramite la console Elastic APIs Beanstalk, e. CLIs

Le EC2 istanze Amazon mostrate nel diagramma fanno parte di un gruppo di sicurezza. Un gruppo di sicurezza definisce le regole del firewall per le proprie istanze. Per impostazione predefinita, Elastic Beanstalk definisce un gruppo di sicurezza che consente a tutti di connettersi utilizzando la porta 80 (HTTP). Puoi definire più di un gruppo di sicurezza. Ad esempio, puoi definire un gruppo di sicurezza per il server di database. Per ulteriori informazioni sui gruppi di EC2 sicurezza Amazon e su come configurarli per l'applicazione Elastic Beanstalk, consulta. EC2 gruppi di sicurezza

Ambienti worker Elastic Beanstalk

AWS le risorse create per un livello di ambiente di lavoro includono un gruppo Auto Scaling, una o più EC2 istanze Amazon e un ruolo IAM. Per il livello di ambiente worker, Elastic Beanstalk crea e effettua il provisioning di una coda Amazon SQS se non ne hai già una. Quando avvii un ambiente di lavoro, Elastic Beanstalk installa i file di supporto necessari per il linguaggio di programmazione prescelto e un demone su EC2 ogni istanza del gruppo Auto Scaling. Il daemon legge i messaggi da una coda Amazon SQS. Il daemon invia i dati di ciascun messaggio che legge dall'applicazione Web in esecuzione nell'ambiente lavoratore per l'elaborazione. Se hai più istanze nel tuo ambiente worker, ogni istanza ha il proprio daemon, ma tutte leggono dalla stessa coda Amazon SQS.

Il diagramma seguente mostra i diversi componenti e le relative interazioni tra ambienti e servizi. AWS



Amazon CloudWatch viene utilizzato per allarmi e monitoraggio dello stato di salute. Per ulteriori informazioni, consulta Report dello stato di base.

Per informazioni dettagliate su come funziona il livello dell'ambiente worker, consulta Ambienti worker Elastic Beanstalk.

Considerazioni sulla progettazione per le tue applicazioni Elastic Beanstalk

Poiché le applicazioni distribuite utilizzando le risorse AWS Elastic Beanstalk vengono eseguite su Cloud AWS risorse, è necessario tenere presenti diversi fattori di configurazione per ottimizzare le applicazioni: scalabilità, sicurezza, archiviazione persistente, tolleranza agli errori, distribuzione dei contenuti, aggiornamenti e patch software e connettività. Ognuno di questi concetti è trattato separatamente in questo argomento. Per un elenco completo di AWS white paper tecnici, che trattano argomenti come l'architettura, la sicurezza e l'economia, consulta i white paper sul cloud computing.AWS

Scalabilità

Quando si lavora in un ambiente hardware fisico, a differenza di un ambiente cloud, è possibile avvicinarsi alla scalabilità in due modi diversi. È possibile dimensionare verso l'alto attraverso un dimensionamento verticale o aumentare orizzontalmente attraverso il dimensionamento orizzontale. L'approccio di dimensionamento verso l'alto richiede un investimento in hardware potente in grado di supportare le crescenti esigenze dell'azienda. L'approccio di aumento orizzontale richiede di seguire un modello di investimento distribuito. Pertanto, le acquisizioni di hardware e applicazioni possono essere più mirate, i set di dati sono federati e la progettazione è orientata ai servizi. L'approccio del dimensionamento verso l'alto può essere costoso ed espone comunque al rischio di non riuscire a soddisfare la domanda. A questo proposito, l'approccio di aumento orizzontale è generalmente più efficace. Tuttavia, quando lo si utilizza, è necessario essere in grado di prevedere la domanda a intervalli regolari e implementare l'infrastruttura in blocchi per soddisfare tale domanda. Questo approccio spesso porta a capacità inutilizzate e impone un attento monitoraggio.

Grazie alla migrazione al cloud, è possibile allineare bene l'infrastruttura alla domanda sfruttando l'elasticità del cloud. L'elasticità aiuta a semplificare l'acquisizione e il rilascio delle risorse. Con questo sistema, l'infrastruttura può essere ridotta e aumentata rapidamente in orizzontale con il variare della domanda. Per utilizzarlo, configurare le impostazioni di Auto Scaling per aumentare o ridurre le dimensioni in base ai parametri dalle risorse dell'ambiente. Ad esempio, puoi impostare metriche (come l'utilizzo del server o la retel/O. You can use Auto Scaling for compute capacity to be added automatically whenever usage rises and for it to be removed whenever usage drops. You can publish system metrics (for example, CPU, memory, disk I/O, and network I/O) su Amazon. CloudWatch Quindi, puoi utilizzarlo CloudWatch per configurare gli allarmi per attivare azioni di Auto Scaling o inviare notifiche in base a queste metriche. Per le istruzioni su come configurare Auto Scaling, consultare Auto Scaling delle istanze dell'ambiente Elastic Beanstalk.

Consigliamo anche di progettare tutte le applicazioni Elastic Beanstalk il più possibile stateless, grazie all'utilizzo di componenti con legami deboli e tolleranza ai guasti che possono essere dimensionati orizzontalmente in base alle esigenze. Per ulteriori informazioni sulla progettazione di architetture applicative scalabili per, AWS vedere Well-Architected Framework.AWS

Sicurezza

La sicurezza attiva è una responsabilità condivisa. AWS Amazon Web Services protegge le risorse fisiche nel proprio ambiente e garantisce che il cloud sia un luogo sicuro per l'esecuzione delle applicazioni. L'utente è responsabile della sicurezza dei dati in entrata e in uscita dell'ambiente Elastic Beanstalk e della sicurezza dell'applicazione.

Configurare SSL per proteggere il flusso di informazioni tra l'applicazione e i client. Per configurare SSL, è necessario un certificato gratuito da AWS Certificate Manager (ACM). Se si dispone già di un certificato di un'autorità di certificazione esterna (CA), è possibile utilizzare ACM per importarlo. Altrimenti, puoi importarlo utilizzando AWS CLI.

Se ACM non è disponibile nel tuo Regione AWS, puoi acquistare un certificato da una CA esterna, ad esempio VeriSign Entrust. Quindi, utilizza AWS Command Line Interface (AWS CLI) per caricare un certificato di terze parti o autofirmato e una chiave privata su AWS Identity and Access Management (IAM). La chiave pubblica del certificato consente di autenticare il server con il browser. Serve anche come base per la creazione di chiavi di sessione condivise che eseguono la crittografia dei dati in entrambe le direzioni. Per le istruzioni su come creare, caricare e assegnare un certificato SSL all'ambiente, consultare Configurazione del protocollo HTTPS per l'ambiente Elastic Beanstalk.

Quando si configura un certificato SSL per l'ambiente, i dati tra il client e il load balancer Elastic Load Balancing dell'ambiente vengono crittografati. Per impostazione predefinita, la crittografia viene interrotta sul sistema di bilanciamento del carico e il traffico tra il sistema di bilanciamento del carico e le EC2 istanze Amazon non è crittografato.

Storage persistente

Le applicazioni Elastic Beanstalk vengono eseguite EC2 su istanze Amazon prive di storage locale persistente. Quando le EC2 istanze Amazon terminano, il file system locale non viene salvato. Le nuove EC2 istanze Amazon iniziano con un file system predefinito. Consigliamo di configurare l'applicazione in modo da archiviare i dati in un'origine dati persistente. AWS offre una serie di servizi di archiviazione persistente che è possibile utilizzare per l'applicazione. Tali versioni sono elencate nella tabella seguente.

Servizio di storage	Documentazione per il servizio	Integrazione di Elastic Beanstalk
Amazon S3	Documentazione su Amazon Simple Storage Service	Utilizzo di Elastic Beanstalk con Amazon S3
Amazon Elastic File System	Documentazione su Amazon Elastic File System	Utilizzo di Elastic Beanstalk con Amazon Elastic File System
Amazon Elastic Block Store	Amazon Elastic Block Store Feature Guide: Elastic Block Store	

Servizio di storage	Documentazione per il servizio	Integrazione di Elastic Beanstalk
Amazon DynamoDB	Documentazione di Amazon DynamoDB	Utilizzo di Elastic Beanstalk con Amazon DynamoDB
Amazon Relational Database Service (RDS)	Documentazione su Amazon Relational Database Service	Utilizzo di Elastic Beanstalk con Amazon RDS

> **Note**
>
> Elastic Beanstalk crea un utente webapp da configurare come proprietario delle directory delle applicazioni sulle istanze. EC2 Per le versioni della piattaforma Amazon Linux 2 rilasciate il o dopo il 3 febbraio 2022, Elastic Beanstalk assegna all'utente webapp un valore uid (ID utente) e gid (ID gruppo) pari a 900 per i nuovi ambienti. Lo stesso vale per gli ambienti esistenti dopo un aggiornamento della versione della piattaforma. Questo approccio mantiene coerente l'autorizzazione di accesso per l'utente webapp nell'archiviazione del file system permanente.
>
> Nell'improbabile situazione in cui un altro utente o processo stia già utilizzando il valore 900, il sistema operativo utilizzerà un altro valore di uid e gid dell'utente webapp. Esegui il comando Linux id webapp sulle tue EC2 istanze per verificare i valori uid e gid assegnati all'utente della webapp.

Tolleranza ai guasti

In linea generale, devi essere pessimista quando progetti l'architettura per il cloud. Sfruttare l'elasticità che offre. Progetta, implementa e distribuisci sempre includendo il ripristino automatizzato in seguito a errori. Utilizza più zone di disponibilità per le tue EC2 istanze Amazon e per Amazon RDS. Le zone di disponibilità sono concettualmente simili ai data center logici. Usa Amazon CloudWatch per ottenere maggiore visibilità sullo stato della tua applicazione Elastic Beanstalk e intraprendere le azioni appropriate in caso di guasto hardware o peggioramento delle prestazioni. Configura le impostazioni di Auto Scaling per mantenere la tua flotta di EC2 istanze Amazon a una dimensione fissa in modo che le EC2 istanze Amazon non integre vengano sostituite da nuove. Se si utilizza Amazon RDS, configurare il periodo di conservazione per i backup in modo che Amazon RDS possa eseguire i backup automatici.

Distribuzione di contenuti

Quando gli utenti si connettono al tuo sito Web, le loro richieste potrebbero essere instradate attraverso una serie di reti individuali. In questo modo gli utenti possono riscontrare una riduzione delle prestazioni a causa di una latenza elevata. Amazon CloudFront può aiutarti a ridurre i problemi di latenza distribuendo i tuoi contenuti web, come immagini e video, su una rete di edge location in tutto il mondo. Le richieste degli utenti vengono indirizzate alla edge location più vicina, in modo che i contenuti vengano forniti con le migliori prestazioni possibili. CloudFront funziona perfettamente con Amazon S3, che archivia in modo duraturo le versioni originali e definitive dei tuoi file. Per ulteriori informazioni su Amazon CloudFront, consulta l'Amazon CloudFront Developer Guide.

Applicazione di patch e aggiornamenti software

AWS Elastic Beanstalk rilascia regolarmente aggiornamenti della piattaforma per fornire correzioni, aggiornamenti software e nuove funzionalità. Elastic Beanstalk offre diverse opzioni per gestire gli aggiornamenti della piattaforma. Gli aggiornamenti gestiti della piattaforma aggiornano automaticamente l'ambiente alla versione più recente della piattaforma durante una finestra di manutenzione pianificata mentre l'applicazione rimane operativa. Per gli ambienti creati il 25 novembre 2019 o successivamente tramite la console Elastic Beanstalk, gli aggiornamenti gestiti sono abilitati per impostazione predefinita (quando possibile). È inoltre possibile avviare manualmente gli aggiornamenti utilizzando la console di Elastic Beanstalk o la CLI EB.

Connettività

Elastic Beanstalk deve essere in grado di connettersi alle istanze del tuo ambiente per completare le distribuzioni. Quando distribuisci un'applicazione Elastic Beanstalk all'interno di un Amazon VPC, la configurazione necessaria per abilitare la connettività dipende dal tipo di ambiente Amazon VPC che crei:

- Per gli ambienti con una singola istanza non è richiesta alcuna configurazione aggiuntiva. Questo perché, con questi ambienti, Elastic Beanstalk assegna a EC2 ogni istanza Amazon un indirizzo IP elastico pubblico che consente all'istanza di comunicare direttamente con Internet.
- Per gli ambienti dimensionabili e con bilanciamento del carico in un Amazon VPC con sottoreti sia pubbliche che private, devi completare le attività seguenti:
  - Crea un sistema di bilanciamento del carico nella sottorete pubblica per indirizzare il traffico in entrata da Internet alle istanze Amazon. EC2
  - Crea un dispositivo NAT (Network Address Translation) per indirizzare il traffico in uscita dalle EC2 istanze Amazon in sottoreti private verso Internet.

  - Crea regole di routing in entrata e in uscita per le EC2 istanze Amazon all'interno della sottorete privata.
  - Se utilizzi un'istanza NAT, configura i gruppi di sicurezza per l'istanza NAT e EC2 le istanze Amazon per abilitare la comunicazione Internet.
- Per un ambiente scalabile con bilanciamento del carico in un Amazon VPC con una sottorete pubblica, non è richiesta alcuna configurazione aggiuntiva. Questo perché, con questo ambiente, le EC2 istanze Amazon sono configurate con un indirizzo IP pubblico che consente alle istanze di comunicare con Internet.

Per ulteriori informazioni sull'utilizzo di Elastic Beanstalk con Amazon VPC, consulta Utilizzo di Elastic Beanstalk con Amazon VPC.

Gestione delle applicazioni Elastic Beanstalk

Questo capitolo descrive come gestire e configurare le applicazioni Elastic Beanstalk. Il primo passaggio per l'utilizzo AWS Elastic Beanstalk consiste nel creare un'applicazione che rappresenti la tua applicazione web in. AWS In Elastic Beanstalk un'applicazione funge da container per gli ambienti che eseguono l'app Web e per le versioni del codice sorgente dell'app Web, le configurazioni salvate, i log e altri artefatti che crei durante l'utilizzo di Elastic Beanstalk.

Come creare un’applicazione

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegli Applications (Applicazioni), quindi Create application (Crea applicazione).
3. Utilizzare il modulo sullo schermo per specificare il nome di un'applicazione.
4. (Facoltativo) Fornisci una descrizione e aggiungi le chiavi e i valori dei tag.
5. Scegli Create (Crea).

Dopo la creazione di una nuova applicazione, la console chiede di creare un ambiente. Per informazioni dettagliate su tutte le opzioni disponibili, consulta Creazione di un ambiente Elastic Beanstalk.

Se un'applicazione non è più necessaria, puoi eliminarla.

> **Warning**
>
> L'eliminazione di un'applicazione termina tutti gli ambienti associati ed elimina tutte le versioni dell'applicazione nonché le configurazioni salvate che appartengono all'applicazione.

Come eliminare un’applicazione

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applications (Applicazioni), quindi selezionare l'applicazione nell'elenco.
3. Scegliere Actions (Operazioni), quindi Delete application (Elimina applicazione).

Argomenti

- Console di gestione delle applicazioni Elastic Beanstalk
- Gestione delle versioni delle applicazioni
- Crea un bundle di sorgenti dell'applicazione Elastic Beanstalk
- Utilizzo dell'EB CLI con AWS CodeBuild
- Tagging delle applicazioni
- Tagging delle risorse dell'applicazione Elastic Beanstalk

Console di gestione delle applicazioni Elastic Beanstalk

Questo argomento spiega come utilizzare la AWS Elastic Beanstalk console per gestire applicazioni, versioni delle applicazioni e configurazioni salvate.

Per accedere alla console di gestione delle applicazioni

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.

   La pagina panoramica dell'applicazione mostra un elenco con una panoramica di tutti gli ambienti associati all'applicazione.
3. È possibile continuare in modi diversi:

   a. Dal menu a discesa Azioni, puoi scegliere una delle azioni di gestione dell'applicazione: Crea ambiente, Elimina applicazione, Visualizza le versioni dell'applicazione, Visualizza le configurazioni salvate, Ripristina l'ambiente terminato.

      Per avviare un ambiente in questa applicazione, puoi scegliere direttamente Crea ambiente. Per informazioni dettagliate, vedi the section called “Creazione di ambienti”.

   b. La pagina elenca il nome dell'ambiente accanto alle applicazioni distribuite in un ambiente. Scegliere il nome di un ambiente per accedere alla console di gestione dell'ambiente specifico per configurarlo, monitorarlo o gestirlo.

   c. Quando si seleziona un'applicazione dall'elenco, il riquadro di navigazione a sinistra elenca l'applicazione.

- Scegliere Application versions (Versioni applicazione) seguendo il nome dell'applicazione nel riquadro di navigazione per visualizzare e gestire le versioni dell'applicazione per l'applicazione in uso.

  Una versione dell'applicazione è una versione caricata del codice dell'applicazione. Puoi caricare nuove versioni, distribuire una versione esistente in qualsiasi ambiente dell'applicazione o eliminare le versioni precedenti. Per ulteriori informazioni, consulta Gestione delle versioni delle applicazioni.
- Scegliere Saved configurations (Configurazioni salvate) seguendo il nome dell'applicazione nel riquadro di navigazione per visualizzare e gestire le configurazioni salvate dagli ambienti in esecuzione.

  Una configurazione salvata è una raccolta di impostazioni che consente di ripristinare lo stato precedente delle impostazioni di un ambiente o creare un ambiente con le stesse impostazioni. Per ulteriori informazioni, consulta . Utilizzo delle configurazioni salvate di Elastic Beanstalk.

Gestione delle versioni delle applicazioni

Questo argomento spiega le versioni delle applicazioni e come crearle e gestirle.

Elastic Beanstalk crea una versione dell'applicazione ogni volta che carichi il codice sorgente. Questo si verifica in genere quando crei un ambiente oppure carichi e distribuisci il codice usando la console di gestione dell'ambiente o l'interfaccia a riga di comando EB. Elastic Beanstalk elimina queste versioni dell'applicazione in base alla policy per il ciclo di vita dell'applicazione e quando elimini l'applicazione. Per ulteriori informazioni sulla policy per il ciclo di vita dell'applicazione, consulta Configurazione delle impostazioni del ciclo di vita delle versioni delle applicazioni.

Puoi anche caricare un bundle fonte senza implementarlo dalla console di gestione dell'applicazione o con il comando CLI EB.eb appversion. Elastic Beanstalk archivia i bundle di origine in Amazon Simple Storage Service (Amazon S3) e non li elimina automaticamente.

Puoi applicare tag a una versione dell'applicazione al momento della sua creazione e modificare i tag di versioni dell'applicazione esistenti. Per informazioni dettagliate, vedi Tagging delle versioni delle applicazioni.

Creazione di versioni delle applicazioni

È inoltre possibile creare una nuova versione dell'applicazione utilizzando la CLI EB. Per ulteriori informazioni, consulta eb appversion il capitolo Comandi CLI EB.

> **Note**
>
> Nel tempo, l'applicazione può accumulare un numero elevato di versioni dell'applicazione. Per risparmiare spazio di archiviazione ed evitare di raggiungere la quota della versione dell'applicazione, è consigliabile eliminare le versioni dell'applicazione che non sono più necessarie.

Il file specificato nella procedura seguente è associato all'applicazione. Puoi distribuire la versione dell'applicazione in un ambiente nuovo o esistente.

Come creare una nuova versione dell'applicazione

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Application versions (Versioni applicazione).
4. Scegli Carica. Utilizza il modulo su schermo per caricare il bundle di origine dell'applicazione.

   > **Note**
   >
   > Il limite per le dimensioni del file del bundle di origine è di 500 MB.

5. Facoltativamente, fornire una breve descrizione e aggiungere chiavi e valori dei tag.
6. Scegli Carica.

Eliminazione delle versioni delle applicazioni

È inoltre possibile eliminare una versione dell'applicazione utilizzando la CLI EB. Per ulteriori informazioni, consulta eb appversion il capitolo Comandi CLI EB.

> **Note**
>
> L'eliminazione di una versione dell'applicazione non ha alcun effetto sugli ambienti che eseguono tale versione.

Inoltre puoi configurare Elastic Beanstalk per eliminare automaticamente le vecchie versioni configurando le impostazioni del ciclo di vita della versione dell'applicazione. Se configurate, queste impostazioni del ciclo di vita dell'applicazione vengono applicate quando crei nuove versioni dell'applicazione. Ad esempio, se configuri un massimo di 25 versioni dell'applicazione Elastic Beanstalk elimina la versione più vecchia nel momento in cui carichi una ventiseiesima versione. Se imposti una durata massima di 90 giorni, le eventuali versioni più vecchie di 90 giorni vengono eliminate quando carichi una nuova versione. Per informazioni dettagliate, vedi the section called “Ciclo di vita delle versioni”.

Come eliminare una versione di un’applicazione

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Application versions (Versioni applicazione).
4. Selezionare una o più versioni dell'applicazione che si desidera eliminare.
5. Selezionare Actions (Operazioni), quindi selezionare Delete (Elimina).
6. (Facoltativo) Per lasciare il bundle di origine dell'applicazione per queste versioni dell'applicazione nel bucket Amazon Simple Storage Service (Amazon S3) deseleziona la casella Delete versions from Amazon S3 (Elimina versioni da Amazon S3).
7. Scegli Elimina.

Se non scegli di eliminare il bundle di origine da Amazon S3, Elastic Beanstalk elimina comunque la versione dai suoi record. Tuttavia, il bundle di origine viene lasciato nel bucket di storage Elastic Beanstalk. La quota della versione dell'applicazione si applica solo alle versioni monitorate da Elastic Beanstalk. Puoi quindi eliminare versioni per rimanere entro la quota, mantenendo tutti i bundle di origine in Amazon S3.

> **Note**
>
> La quota relativa alle versioni dell'applicazione non si applica ai bundle di origine, ma potrebbero comunque venire addebitati costi per Amazon S3 e le informazioni personali potrebbero venire conservate più a lungo del necessario. Elastic Beanstalk non elimina mai automaticamente i bundle di origine. È consigliabile eliminare i bundle di origine quando non sono più necessari.

Configurazione delle impostazioni del ciclo di vita delle versioni delle applicazioni

Questo argomento spiega le politiche e le quote che Elastic Beanstalk applica alle versioni dell'applicazione in un determinato ambiente, incluso per quanto tempo una versione dell'applicazione rimane in un ambiente.

Ogni volta che carichi una nuova versione dell'applicazione con la console Elastic Beanstalk o l'EB CLI, Elastic Beanstalk crea una versione dell'applicazione. Se non elimini le versioni che non usi più, finirai per raggiungere la quota di versioni dell'applicazione e non sarai in grado di creare nuove versioni dell'applicazione stessa.

Puoi evitare di raggiungere la quota applicando una policy del ciclo di vita delle versioni dell'applicazione alle tue applicazioni. Una policy del ciclo di vita indica a Elastic Beanstalk di eliminare le versioni delle applicazioni obsolete o di eliminare le versioni delle applicazioni quando il numero complessivo delle versioni per un'applicazione supera un valore specificato.

Elastic Beanstalk applica la policy del ciclo di vita di un'applicazione ogni volta che crei una nuova versione dell'applicazione ed elimina fino a 100 versioni ogni volta che viene applicata la policy del ciclo di vita. Elastic Beanstalk elimina le versioni precedenti dopo aver creato la nuova versione e non conteggia la nuova versione ai fini del numero massimo di versioni definito nella policy.

Elastic Beanstalk non elimina le versioni delle applicazioni attualmente utilizzate da un ambiente o versioni di applicazioni distribuite in ambienti terminati meno di dieci settimane prima che la policy sia stata attivata.

La quota delle versioni dell'applicazione è valida per tutte le applicazioni di una regione. Se disponi di numerose applicazioni, configura ognuna di esse con una policy del ciclo di vita appropriata per evitare di raggiungere la quota. Ad esempio, se in una regione hai 10 applicazioni e la quota è pari a 1.000 versioni dell'applicazione, valuta la possibilità di stabilire una policy del ciclo di vita con una

quota di 99 versioni per tutte le applicazioni. In alternativa, puoi impostare altri valori in ciascuna applicazione, purché il totale sia inferiore a 1.000 versioni dell'applicazione. Elastic Beanstalk applica la policy solo se la creazione della versione dell'applicazione va a buon fine; pertanto se hai già raggiunto la quota è necessario eliminare alcune versioni manualmente prima di creare una nuova versione.

Per impostazione predefinita, Elastic Beanstalk lascia il bundle di origine della versione dell'applicazione in Amazon S3 per evitare la perdita di dati. Puoi eliminare il bundle di origine per recuperare spazio.

Puoi configurare le impostazioni del ciclo di vita tramite l'Elastic Beanstalk CLI e. APIs Vedi CreateApplication(utilizzando eb appversion il `ResourceLifecycleConfig` parametro) e per i dettagli. UpdateApplicationResourceLifecycle

Specifica delle impostazioni del ciclo di vita delle applicazioni nella console

Puoi specificare le impostazioni del ciclo di vita nella console Elastic Beanstalk.

Per specificare le impostazioni del ciclo di vita delle applicazioni.

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Application versions (Versioni applicazione).
4. Seleziona Impostazioni.
5. Utilizza il modulo sullo schermo per configurare le impostazioni del ciclo di vita dell'applicazione.
6. Scegli Save (Salva).

Nella pagina delle impostazioni, puoi eseguire quanto segue.

- Configura le impostazioni del ciclo di vita in base al numero totale delle versioni dell'applicazione o all'età delle versioni dell'applicazione.
- Specifica se eliminare il bundle di origine da S3 quando la versione dell'applicazione viene eliminata.
- Specifica il ruolo del servizio con cui la versione dell'applicazione viene eliminata. Per includere tutte le autorizzazioni necessarie per l'eliminazione della versione, scegli il ruolo di servizio Elastic

Beanstalk predefinito denominato `aws-elasticbeanstalk-service-role` o un altro ruolo di servizio utilizzando le policy del servizio gestite Elastic Beanstalk. Per ulteriori informazioni, consulta Gestione dei ruoli del servizio Elastic Beanstalk.

Tagging delle versioni delle applicazioni

Questo argomento spiega i vantaggi dell'etichettatura delle versioni dell'applicazione Elastic Beanstalk e come gestire i tag.

Puoi applicare tag alle versioni delle tue applicazioni. AWS Elastic Beanstalk I tag sono coppie chiave-valore associate AWS alle risorse. Per informazioni sul tagging delle risorse, sui casi d'uso, sui vincoli delle chiavi e dei valori di tag e sui tipi di risorse supportati di Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Puoi specificare i tag al momento della creazione di una versione dell'applicazione. In una versione dell'applicazione esistente, puoi aggiungere o rimuovere i tag e aggiornare i valori dei tag esistenti. Puoi aggiungere fino a 50 tag per ogni versione dell'applicazione.

Aggiunta di tag durante la creazione della versione dell'applicazione

Quando utilizzi la console Elastic Beanstalk per creare un ambiente e scegli di caricare una versione del codice dell'applicazione, puoi specificare chiavi e valori dei tag da associare alla nuova versione dell'applicazione.

Puoi anche utilizzare la console Elastic Beanstalk per caricare una versione dell'applicazione senza utilizzarla immediatamente in un ambiente. È possibile specificare chiavi e valori dei tag quando si carica una versione dell'applicazione.

Con AWS CLI o con altri client basati su API, aggiungi tag utilizzando il `--tags` parametro sul comando. create-application-version

```
$ aws elasticbeanstalk create-application-version \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --version-label v1
```

Quando utilizza la CLI EB per creare o aggiornare un ambiente, una versione dell'applicazione viene creata dal codice distribuito. Non esiste un modo diretto per applicare un tag a una versione dell'applicazione durante la sua creazione tramite la CLI EB. Consulta la sezione seguente per informazioni sull'aggiunta di tag a una versione dell'applicazione esistente.

Gestione dei tag di una versione dell'applicazione esistente

Puoi aggiungere, aggiornare ed eliminare i tag in una versione dell'applicazione Elastic Beanstalk esistente.

Per gestire i tag di una versione dell'applicazione utilizzando la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Application versions (Versioni applicazione).
4. Selezionare la versione dell'applicazione che si desidera gestire.
5. Scegliere Actions (Operazioni), quindi Manage tags (Gestisci tag).
6. Utilizzare il modulo su schermo per aggiungere, aggiornare o eliminare tag.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Se utilizzi la CLI EB per aggiornare la versione dell'applicazione, utilizza eb tags per aggiungere, aggiornare, eliminare o elencare i tag.

Ad esempio, il comando seguente elenca i tag in una versione dell'applicazione.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Il comando seguente aggiorna il tag `mytag1` ed elimina il tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Per un elenco completo delle opzioni e per altri esempi, consulta `eb tags`.

Con il AWS CLI o altri client basati su API, usa il list-tags-for-resource comando per elencare i tag di una versione dell'applicazione.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-
version"
```

Utilizza il comando update-tags-for-resource per aggiungere, aggiornare o eliminare i tag in una versione dell'applicazione.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:applicationversion/my-app/my-version"
```

Specifica i tag da aggiungere e i tag da aggiornare nel parametro `--tags-to-add` di update-tags-for-resource. È stato aggiunto un tag non esistente e il valore di un tag esistente è stato aggiornato.

> **Note**
>
> Per utilizzare alcuni AWS CLI comandi e CLI EB con una versione dell'applicazione Elastic Beanstalk, è necessario l'ARN della versione dell'applicazione. È possibile recuperare l'ARN utilizzando il seguente comando.
>
> ```
> $ aws elasticbeanstalk describe-application-versions --application-name my-app
> --version-label my-version
> ```

Crea un bundle di sorgenti dell'applicazione Elastic Beanstalk

Questo argomento spiega come caricare i file sorgente dell'applicazione su Elastic Beanstalk in un pacchetto sorgente. Spiega i requisiti di un pacchetto sorgente, la struttura e gli approcci per crearne uno.

Quando si utilizza la AWS Elastic Beanstalk console per distribuire una nuova applicazione o una versione dell'applicazione, è necessario caricare i file dell'applicazione in un pacchetto sorgente. Il tuo bundle di origine deve soddisfare i seguenti requisiti:

- Deve essere costituito da un singolo file `ZIP` o dal file `WAR` (puoi includere più file `WAR` all'interno del file `ZIP`)
- Non superare i 500 MB

- Non includere una cartella padre o una directory di livello superiore (le sottodirectory vanno bene)

Se desideri distribuire un'applicazione lavoratore che elabora attività periodiche in background, il tuo bundle di origine dell'applicazione deve anche includere un file `cron.yaml`. Per ulteriori informazioni, consulta Attività periodiche.

Se stai distribuendo l'applicazione con Elastic Beanstalk Command Line Interface (EB CLI), AWS Toolkit for Eclipse o AWS Toolkit for Visual Studio, il file ZIP o WAR verrà strutturato automaticamente correttamente. Per ulteriori informazioni, consultare Configurazione dell'interfaccia a riga di comando EB (EB CLI) per gestire Elastic Beanstalk, Distribuzione di applicazioni Java con Elastic Beanstalk e La AWS Toolkit for Visual Studio.

Sections

- Creazione di un bundle di origine dalla riga di comando
- Creazione di un bundle di origine con Git
- Compressione dei file in Mac OS X Finder o in Windows Explorer
- Creazione di un bundle di origine per un'applicazione .NET
- Test del bundle di origine

Creazione di un bundle di origine dalla riga di comando

Creazione di un bundle di origine tramite il comando `zip`. Per includere file e cartelle nascosti, utilizza un modello come il seguente.

```
~/myapp$ zip ../myapp.zip -r * .[^.]*
  adding: app.js (deflated 63%)
  adding: index.js (deflated 44%)
  adding: manual.js (deflated 64%)
  adding: package.json (deflated 40%)
  adding: restify.js (deflated 85%)
  adding: .ebextensions/ (stored 0%)
  adding: .ebextensions/xray.config (stored 0%)
```

Così facendo ci si assicura che i file di configurazione di Elastic Beanstalk e altri file e cartelle che iniziano con un punto siano inclusi nell'archivio.

Per le applicazioni Web Tomcat, utilizza `jar` per creare un archivio Web.

```
~/myapp$ jar -cvf myapp.war .
```

I comandi precedenti includono i file nascosti che potrebbe aumentare la dimensione del tuo bundle di origine inutilmente. Per un maggiore controllo, utilizza un modello di file più dettagliato, oppure crea il tuo bundle di origine con Git.

Creazione di un bundle di origine con Git

Se usi Git per gestire il codice sorgente della tua applicazione, utilizza il comando `git archive` per creare il bundle di origine.

```
$ git archive -v -o myapp.zip --format=zip HEAD
```

`git archive` include solo i file che sono archiviati in git ed esclude i file ignorati e i file git. Questo ti aiuta a mantenere il tuo bundle di origine il più piccolo possibile. per ulteriori informazioni visita la pagina del manuale git-archive.

Compressione dei file in Mac OS X Finder o in Windows Explorer

Quando crei un file `ZIP` in Mac OS X Finder o Windows Explorer, assicurati di comprimere i file e le sottocartelle, invece di comprimere la cartella padre.

> **Note**
>
> L'interfaccia utente grafica (GUI) su Mac OS X e sui sistemi operativi basati su Linux non consente di visualizzare i file e le cartelle con nomi che iniziano con un punto (.). Utilizza la riga di comando invece della GUI per comprimere la tua applicazione se il file `ZIP` deve includere una cartella nascosta, ad esempio `.ebextensions`. Per le procedure a riga di comando per creare un file `ZIP` su Mac OS X o su un sistema operativo basato su Linux, vedi Creazione di un bundle di origine dalla riga di comando.

Example

Supponiamo che tu abbia una cartella del progetto Python denominata `myapp`, che comprende i seguenti file e sottocartelle:

```
myapplication.py
```

```
README.md
static/
static/css
static/css/styles.css
static/img
static/img/favicon.ico
static/img/logo.png
templates/
templates/base.html
templates/index.html
```

Come indicato nell'elenco dei requisiti qui sopra, il tuo bundle di origine deve essere compresso senza una cartella padre, in modo che la sua struttura decompressa non includa una directory con un ulteriore livello superiore. In questo esempio, non deve essere creata una cartella `myapp` quando i file vengono decompressi (o, alla riga di comando, nessun segmento `myapp` deve essere aggiunto ai percorsi dei file).

Questo esempio di struttura di file viene utilizzato in questo argomento per illustrare come comprimere i file.

Creazione di un bundle di origine per un'applicazione .NET

Se usi Visual Studio, puoi utilizzare lo strumento di distribuzione incluso in AWS Toolkit for Visual Studio per distribuire l'applicazione.NET su Elastic Beanstalk. Per ulteriori informazioni, consulta Distribuzione di applicazioni Elastic Beanstalk in.NET utilizzando strumenti di distribuzione AWS.

Se devi creare manualmente un bundle di origine per la tua applicazione .NET, non puoi semplicemente creare un file `ZIP` che contiene la directory del progetto. È necessario creare un pacchetto di distribuzione Web per il tuo progetto idoneo alla distribuzione su Elastic Beanstalk. Esistono diversi metodi per creare un pacchetto di distribuzione:

- Crea il pacchetto di distribuzione utilizzando la procedura guidata Publish Web (Pubblicazione Web) in Visual Studio. Per ulteriori informazioni, andare su Come creare un pacchetto di distribuzione Web in Visual Studio.

> **Important**
>
> Quando crei il pacchetto di distribuzione Web, è necessario iniziare il Site name (Nome del sito) con `Default Web Site`.

- Se hai un progetto .NET, puoi creare il pacchetto di distribuzione utilizzando il comando msbuild come mostrato nel seguente esempio.

> **Important**
>
> Il parametro `DeployIisAppPath` deve iniziare con `Default Web Site`.

```
C:/> msbuild <web_app>.csproj /t:Package /p:DeployIisAppPath="Default Web Site"
```

- Se disponi di un progetto di un sito Web, è possibile utilizzare lo strumento IIS Web Deploy per creare il pacchetto di distribuzione. Per ulteriori informazioni, consulta la pagina relativa a creazione del pacchetto e ripristino di un sito Web.

> **Important**
>
> Il parametro `apphostconfig` deve iniziare con `Default Web Site`.

Se distribuisci più applicazioni o un'applicazione ASP.NET Core, posiziona la cartella `.ebextensions` nella radice del bundle di origine, accanto ai bundle dell'applicazione bundle e al file manifest:

```
~/workspace/source-bundle/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- AspNetCore101HelloWorld.zip
|-- AspNetCoreHelloWorld.zip
|-- aws-windows-deployment-manifest.json
`-- VS2015AspNetWebApiApp.zip
```

Test del bundle di origine

Può essere opportuno testare il tuo bundle di origine in locale prima di caricarlo su Elastic Beanstalk. Poiché Elastic Beanstalk utilizza essenzialmente la riga di comando per estrarre i file, si consiglia di eseguire i test dalla riga di comando anziché con uno strumento GUI.

Verifica che i file decompressi appaiano nella stessa cartella dell'archivio, invece che in una nuova cartella o directory di primo livello.

Utilizzo dell'EB CLI con AWS CodeBuild

AWS CodeBuild compila il codice sorgente, esegue unit test e prepara artefatti pronti per essere distribuiti. Puoi utilizzare CodeBuild con l'interfaccia a riga di comando di EB per automatizzare la creazione dell'applicazione dal codice sorgente. La creazione dell'ambiente e ogni implementazione successiva inizia con una fase di compilazione e quindi di distribuzione dell'applicazione risultante.

> **Note**
>
> Alcune regioni non offrono CodeBuild. L'integrazione tra Elastic CodeBuild Beanstalk e non funziona in queste aree.
> Per informazioni sui AWS servizi offerti in ogni regione, consulta la Tabella delle regioni.

Creazione di un’applicazione

Per creare un'applicazione Elastic Beanstalk che utilizza CodeBuild

1. Includi un file con le specifiche della CodeBuild build nella cartella dell'applicazione. `buildspec.yml`
2. Aggiungi al file una voce `eb_codebuild_settings` con opzioni specifiche per Elastic Beanstalk.
3. Eseguire eb init nella cartella.

> **Note**
>
> Non utilizzare i caratteri punto (`.`) o spazio ( ) nel nome dell'applicazione quando si utilizza l'EB CLI con. CodeBuild

Elastic Beanstalk CodeBuild estende il formato del file delle specifiche di build per includere le seguenti impostazioni aggiuntive:

```
eb_codebuild_settings:
```

```
CodeBuildServiceRole: role-name
ComputeType: size
Image: image
Timeout: minutes
```

`CodeBuildServiceRole`

L'ARN o il nome del ruolo del servizio AWS Identity and Access Management (IAM) che CodeBuild puoi utilizzare per interagire con i AWS servizi dipendenti per tuo conto. Questo valore è obbligatorio. Se viene omesso, qualsiasi comando eb create o eb deploy eseguito successivamente non riesce.

Per ulteriori informazioni sulla creazione di un ruolo di servizio per CodeBuild, consulta Creare un ruolo CodeBuild di servizio nella Guida per l'AWS CodeBuild utente.

> **Note**
>
> Sono inoltre necessarie le autorizzazioni per eseguire azioni CodeBuild autonomamente. La politica utente gestita di AdministratorAccessElastic AWSElastic Beanstalk - Beanstalk include tutte le autorizzazioni di azione richieste. CodeBuild Se non usi la policy gestita, assicurati di consentire le seguenti autorizzazioni nella tua policy utente.
>
> ```
> "codebuild:CreateProject",
> "codebuild:DeleteProject",
> "codebuild:BatchGetBuilds",
> "codebuild:StartBuild"
> ```
>
> Per informazioni dettagliate, vedi Gestione delle policy utente Elastic Beanstalk.

`ComputeType`

La quantità di risorse utilizzate dal contenitore Docker nell'ambiente di compilazione. CodeBuild I valori validi sono BUILD_ GENERAL1 _SMALL, BUILD_ _MEDIUM e BUILD_ GENERAL1 _LARGE. GENERAL1

`Image`

Il nome dell'hub Docker o dell'immagine Amazon ECR CodeBuild utilizzata per l'ambiente di compilazione. Questa immagine Docker deve contenere tutti gli strumenti e le librerie di

runtime necessari per creare il codice e deve corrispondere alla piattaforma di destinazione dell'applicazione. CodeBuild gestisce e mantiene un set di immagini specificamente pensate per essere utilizzate con Elastic Beanstalk. Ti consigliamo di utilizzare una di queste immagini. Per i dettagli, consulta Docker Images fornito da CodeBuild nella Guida per l'utente.AWS CodeBuild

Il valore `Image` è facoltativo. Se viene omesso, il comando eb init tenta di scegliere l'immagine che meglio corrisponde alla piattaforma di destinazione. Inoltre, se esegui eb init in modalità interattiva e l'immagine non viene scelta automaticamente, ti verrà richiesto di sceglierne una. Alla fine di una corretta inizializzazione, eb init scrive l'immagine scelta nel file `buildspec.yml`.

`Timeout`

La durata, in minuti, dell'esecuzione della CodeBuild build prima del timeout. Questo valore è facoltativo. Per informazioni dettagliate sui valori validi e predefiniti, consulta Creare un progetto di compilazione in CodeBuild.

> **Note**
>
> Questo timeout controlla la durata massima di un' CodeBuild esecuzione e l'EB CLI lo rispetta anche come parte del primo passaggio per creare una versione dell'applicazione. È diverso dal valore che puoi specificare con l'opzione `--timeout` del comando eb create o eb deploy. L'ultimo valore controlla la durata massima per l'interfaccia a riga di comando di EB in attesa della creazione o dell'aggiornamento dell'ambiente.

Compilazione e distribuzione del codice dell'applicazione

Ogni volta che è necessario distribuire il codice dell'applicazione, l'EB CLI CodeBuild esegue una build, quindi distribuisce gli artefatti di build risultanti nell'ambiente. Ciò accade quando crei un ambiente Elastic Beanstalk per l'applicazione usando il comando eb create e ogni volta che distribuisci successivamente le modifiche del codice nell'ambiente usando il comando eb deploy.

Se il CodeBuild passaggio fallisce, la creazione o la distribuzione dell'ambiente non viene avviata.

Tagging delle applicazioni

Questo argomento spiega i vantaggi dell'etichettatura delle applicazioni Elastic Beanstalk. Fornisce inoltre istruzioni per creare e gestire i tag delle applicazioni. I tag sono coppie chiave-valore associate

AWS alle risorse. Per informazioni sul tagging delle risorse, sui casi d'uso, sui vincoli delle chiavi e dei valori di tag e sui tipi di risorse supportati di Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Puoi specificare i tag durante la creazione di un'applicazione. In un'applicazione esistente, puoi aggiungere o rimuovere tag e aggiornare i valori di tag esistenti. Puoi aggiungere fino a 50 tag per ogni applicazione.

Aggiunta di tag durante la creazione di applicazioni

Quando utilizzi la console Elastic Beanstalk per creare un'applicazione, puoi specificare chiavi e valori dei tag nella finestra di dialogo Create New Application (Crea nuova applicazione).

Se utilizzi la CLI EB per creare un'applicazione, utilizza l'opzione `--tags` con eb init per aggiungere i tag.

```
~/workspace/my-app$ eb init --tags mytag1=value1,mytag2=value2
```

Con AWS CLI o con altri client basati su API, aggiungi tag utilizzando il `--tags` parametro sul comando. create-application

```
$ aws elasticbeanstalk create-application \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --version-label v1
```

Gestione dei tag di un'applicazione esistente

Puoi aggiungere, aggiornare ed eliminare i tag in un'applicazione Elastic Beanstalk esistente.

Per gestire i tag di un'applicazione nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Scegliere Actions (Operazioni), quindi Manage tags (Gestisci tag).
4. Utilizzare il modulo su schermo per aggiungere, aggiornare o eliminare tag.
5. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Se utilizzi la CLI EB per aggiornare l'applicazione, utilizza eb tags per aggiungere, aggiornare, eliminare o elencare i tag.

Ad esempio, il comando seguente consente di elencare i tag in un'applicazione.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Il comando seguente aggiorna il tag `mytag1` ed elimina il tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Per un elenco completo delle opzioni e per altri esempi, consulta `eb tags`.

Con il AWS CLI o altri client basati su API, usa il list-tags-for-resource comando per elencare i tag di un'applicazione.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Utilizza il comando update-tags-for-resource per aggiungere, aggiornare o eliminare i tag in un'applicazione.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Specifica i tag da aggiungere e i tag da aggiornare nel parametro `--tags-to-add` di update-tags-for-resource. È stato aggiunto un tag non esistente e il valore di un tag esistente è stato aggiornato.

> **Note**
>
> Per utilizzare alcuni AWS CLI comandi e CLI di EB con un'applicazione Elastic Beanstalk, è necessario l'ARN dell'applicazione. È possibile recuperare l'ARN utilizzando il seguente comando.
>
> ```
> $ aws elasticbeanstalk describe-applications --application-names my-app
> ```

Tagging delle risorse dell'applicazione Elastic Beanstalk

Questo argomento spiega i vantaggi dell'utilizzo dei tag con le risorse dell'applicazione Elastic Beanstalk e i relativi vincoli. Spiega inoltre come creare e gestire i tag per le risorse dell'applicazione.

È possibile applicare tag alle risorse delle AWS Elastic Beanstalk applicazioni. I tag sono coppie chiave-valore associate AWS alle risorse. che possono aiutare a suddividere le risorse in categorie. Sono particolarmente utili se gestisci molte risorse come parte di più AWS applicazioni.

Di seguito sono elencati alcuni modi per utilizzare il tagging con risorse Elastic Beanstalk:

- Fasi di distribuzione: identifica le risorse associate a diverse fasi dell'applicazione, ad esempio sviluppo, beta e produzione.
- Allocazione dei costi: utilizza i report sull'allocazione dei costi per tenere traccia dell'utilizzo delle AWS risorse associate a vari conti di spesa. I report includono risorse con tag e risorse prive di tag; i costi vengono aggregati in base ai tag. Per informazioni su come usare i tag per i report di allocazione dei costi, consultare Utilizzo di tag di allocazione costi per i report di fatturazione personalizzati nella Guida per l'utente su fatturazione e gestione costi di AWS .
- Controllo degli accessi: utilizza i tag per gestire le autorizzazioni a richieste e risorse. Ad esempio, un utente che può solo creare e gestire ambienti beta deve disporre solo dell'accesso a risorse in fase beta. Per informazioni dettagliate, vedi Utilizzo dei tag per controllare l'accesso alle risorse Elastic Beanstalk.

Puoi aggiungere fino a 50 tag per ciascuna risorsa . Gli ambienti sono leggermente diversi: Elastic Beanstalk aggiunge tre tag di sistema predefiniti agli ambienti e non è possibile modificare né eliminare questi tag. Oltre ai tag predefiniti, puoi aggiungere fino a 47 tag aggiuntivi a ogni ambiente.

I seguenti vincoli si applicano a chiavi e valori dei tag:

- Chiavi e valori possono contenere solo lettere, numeri, spazi e i seguenti simboli: `_ . : / = + - @`
- Le chiavi possono contenere fino a 127 caratteri. I valori possono contenere fino a 255 caratteri.

> **Note**
>
> Questi limiti di lunghezza sono relativi a caratteri Unicode in formato UTF-8. Per altre codifiche multibyte, i limiti potrebbero essere inferiori.

- Le chiavi fanno distinzione tra maiuscole e minuscole.
- Le chiavi non possono iniziare con `aws:` o `elasticbeanstalk:`.

Risorse cui è possibile associare tag

Di seguito sono elencati i tipi di risorse Elastic Beanstalk cui è possibile applicare i tag e i collegamenti ad argomenti specifici relativi alla gestione dei tag per ciascuno di essi:

- Applicazioni
- Ambienti
- Versioni delle applicazioni
- Configurazioni salvate

Propagazione di tag ai modelli di avvio

Elastic Beanstalk offre un'opzione per abilitare la propagazione di tag dell'ambiente ai modelli di avvio. Questa opzione offre un supporto continuo per il controllo degli accessi basato su tag (TBAC) con i modelli di avvio.

> **Note**
>
> Le configurazioni di avvio vengono gradualmente eliminate e sostituite da modelli di avvio. Per ulteriori informazioni, consulta Launch configurations nella Amazon EC2 Auto Scaling User Guide.

Per evitare i tempi di inattività delle EC2 istanze in esecuzione, i tag CloudFormation non vengono propagati ai modelli di avvio esistenti. Per un caso d'uso che richiede tag per le risorse dell'ambiente, puoi abilitare Elastic Beanstalk per creare modelli di avvio con tag per queste risorse. A tale scopo, imposta l'opzione `LaunchTemplateTagPropagationEnabled` nello spazio dei nomi aws:autoscaling:launchconfiguration su `true`. Il valore predefinito è `false`.

Il seguente file di configurazione di esempio consente la propagazione di tag ai modelli di avvio.

```
option_settings:
  aws:autoscaling:launchconfiguration:
```

```
    LaunchTemplateTagPropagationEnabled: true
```

Elastic Beanstalk è in grado di propagare i tag ai modelli di avvio solo per le risorse seguenti:

- Volumi EBS
- EC2 istanze
- EC2 interfacce di rete
- CloudFormation modelli di avvio che definiscono una risorsa

Questo vincolo esiste perché consente CloudFormation solo l'uso di tag per la creazione di modelli per risorse specifiche. Per ulteriori informazioni, consulta la TagSpecificationGuida per l'AWS CloudFormation utente.

> **Important**
>
> - La modifica del valore di questa opzione da `false` a `true` per un ambiente esistente può costituire una modifica rivoluzionaria per i tag esistenti.
> - Quando questa funzionalità è abilitata, la propagazione dei tag richiederà la EC2 sostituzione, il che può causare tempi di inattività. Per applicare le modifiche alla configurazione in batch e prevenire i tempi di inattività durante il processo di aggiornamento, puoi abilitare gli aggiornamenti in sequenza. Per ulteriori informazioni, consulta Modifiche di configurazione.

Per ulteriori informazioni sui modelli di avvio, consulta le risorse seguenti:

- Avvia i modelli nella Amazon EC2 Auto Scaling User Guide
- Utilizzo dei modelli nella Guida per l'utente di AWS CloudFormation
- Frammenti di modello Elastic Beanstalk nella Guida per l'utente di AWS CloudFormation

Gestione degli ambienti Elastic Beanstalk

Questo capitolo descrive come creare e gestire gli ambienti Elastic Beanstalk. Questa pagina introduttiva fornisce una panoramica degli aggiornamenti, della manutenzione e delle configurazioni che applicherai nel tempo man mano che l'applicazione e l'ambiente si evolvono.

Funzioni ambientali

È possibile creare e gestire diversi ambienti di sviluppo, test e di utilizzo in un ambiente di produzione ed è possibile distribuire una versione qualsiasi dell'applicazione in qualsiasi ambiente. Gli ambienti possono essere di lunga esecuzione o temporanei. Quando termini un ambiente, è possibile salvare la configurazione per ricrearla in un secondo momento.

Implementazioni delle applicazioni

Mentre sviluppi l'applicazione, la distribuirai spesso, probabilmente in ambienti differenti per scopi diversi. Elastic Beanstalk ti consente di configurare la modalità di esecuzione delle distribuzioni. È possibile distribuire su tutte le istanze nell'ambiente simultaneamente, oppure dividere una distribuzione in batch con le distribuzioni in sequenza.

Modifiche di configurazione

Le modifiche di configurazione vengono elaborate separatamente dalle distribuzioni e hanno un proprio ambito di applicazione. Ad esempio, se modifichi il tipo di istanze EC2 che eseguono la tua applicazione, tutte le istanze devono essere sostituite. Se, invece, modifichi la configurazione del sistema di bilanciamento del carico dell'ambiente, tale modifica può essere eseguita in sede senza interrompere il servizio o diminuire la capacità. È inoltre possibile applicare le modifiche di configurazione che modificano le istanze nell'ambiente in batch con gli aggiornamenti di configurazione in sequenza.

> **Note**
>
> Modifica le risorse nell'ambiente utilizzando solo Elastic Beanstalk. Se modifichi le risorse utilizzando la console di un altro servizio, i comandi CLI o SDKs Elastic Beanstalk non sarà in grado di monitorare con precisione lo stato di tali risorse e non sarai in grado di salvare la configurazione o ricreare l'ambiente in modo affidabile. Le modifiche fuori banda possono anche causare problemi durante l'aggiornamento o la terminazione di un ambiente.

Aggiornamenti della piattaforma

Quando avvii un ambiente, scegli una versione della piattaforma. Aggiorniamo periodicamente le piattaforme con nuove versioni della piattaforma per fornire miglioramenti in termini di prestazioni e nuove caratteristiche. Puoi aggiornare il tuo ambiente alla versione della piattaforma più recente in qualsiasi momento. Consulta la guida alle AWS Elastic Beanstalk piattaforme per un elenco delle piattaforme supportate e una cronologia delle versioni della piattaforma che include gli intervalli di date in cui erano correnti.

Opzioni di architettura

Poiché la tua applicazione cresce in termini di complessità, è possibile suddividerla in più componenti, ognuno in esecuzione in un ambiente separato. Per carichi di lavoro con esecuzione a lungo termine, puoi avviare ambienti worker che elaborano i processi da una coda Amazon Simple Queue Service (Amazon SQS).

Argomenti

- Utilizzo della console di gestione dell'ambiente Elastic Beanstalk
- Creazione di un ambiente Elastic Beanstalk
- Gestione di più ambienti elastici Elastic Beanstalk come gruppo con la CLI EB
- Distribuzione di applicazioni in ambienti Elastic Beanstalk
- Modifiche di configurazione
- Aggiornamento della versione della piattaforma dell'ambiente Elastic Beanstalk
- Annullamento degli aggiornamenti della configurazione dell'ambiente e delle distribuzioni delle applicazioni
- Ricompilazione degli ambienti Elastic Beanstalk
- Tipi di ambiente
- Ambienti worker Elastic Beanstalk
- Creazione di collegamenti tra gli ambienti Elastic Beanstalk
- Ripristino dell'ambiente Elastic Beanstalk da uno stato non valido

Utilizzo della console di gestione dell'ambiente Elastic Beanstalk

Questa sezione descrive come gestire l'ambiente Elastic Beanstalk utilizzando la console di gestione dell'ambiente. La console offre la capacità di gestire la configurazione dell'ambiente ed eseguire

azioni comuni. Queste azioni includono il riavvio dei server Web in esecuzione nell'ambiente, la clonazione dell'ambiente o la ricostruzione dell'ambiente da zero.

Argomenti

- Accesso alla console di gestione dell'ambiente
- Pannello di panoramica dell'ambiente
- Dettagli sull'ambiente
- Operazioni ambiente

Accesso alla console di gestione dell'ambiente

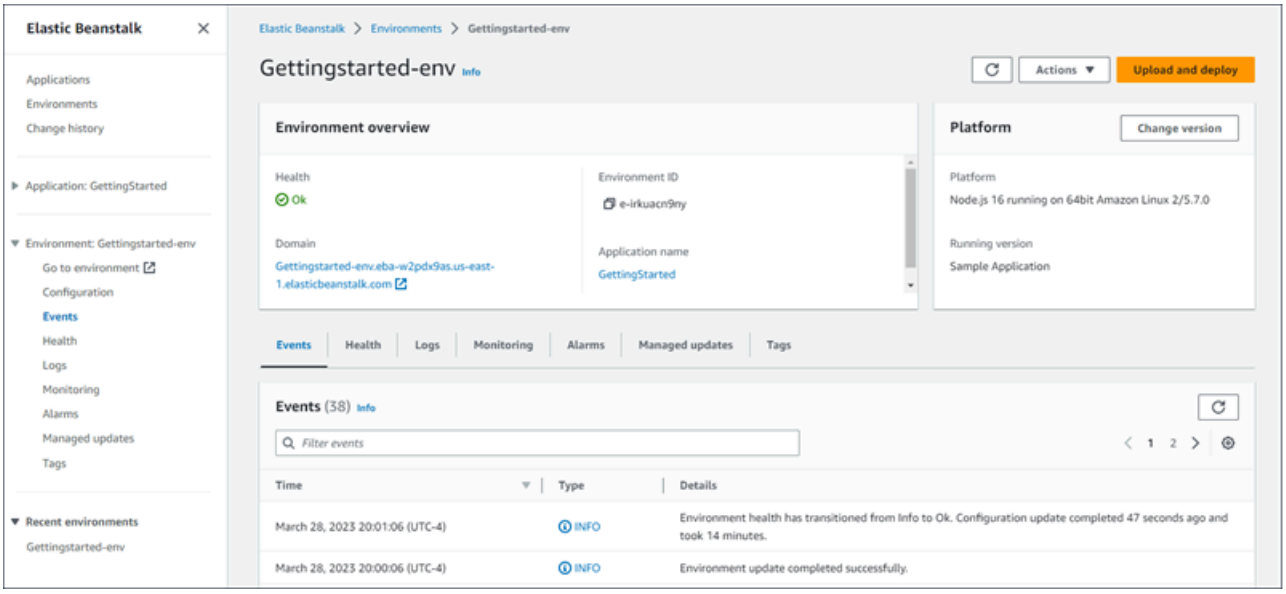
La procedura seguente fornisce i passaggi per avviare la console di gestione dell'ambiente.

Se hai già effettuato l'accesso alla console Elastic Beanstalk, puoi anche avviare la pagina di gestione dell'ambiente da. Console di gestione delle applicazioni Seleziona un ambiente dall'elenco per visualizzare i dettagli della console di gestione per l'ambiente selezionato.

Per accedere alla console di gestione dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

L'immagine seguente illustra la console di gestione dell'ambiente.



Il riquadro superiore è la pagina di panoramica dell'ambiente. Mostra informazioni di primo livello sull'ambiente in uso.

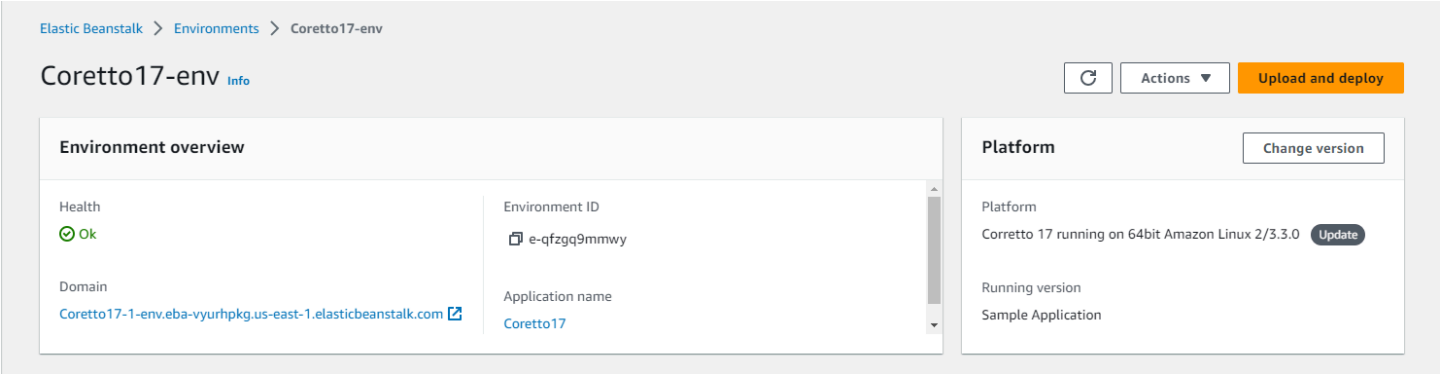
Nella parte inferiore della pagina sono visualizzate schede che forniscono informazioni più dettagliate. La scheda Eventi viene visualizzata per impostazione predefinita. Le pagine collegate alle schede sono elencate anche nel riquadro di navigazione a sinistra sotto l'ambiente.

Il riquadro di navigazione della console mostra il nome dell'applicazione distribuita nell'ambiente, con le relative pagine di gestione delle applicazioni. Il nome dell'ambiente viene visualizzato anche nella pagina di navigazione, seguito dalle pagine di gestione dell'ambiente. I collegamenti elencati sotto il nome dell'ambiente includono anche Vai all'ambiente e Configurazione, oltre alle pagine a schede menzionate in precedenza.

Pannello di panoramica dell'ambiente

Questo argomento descrive le informazioni fornite dal riquadro di panoramica dell'ambiente. Mostra informazioni di primo livello sull'ambiente e si trova nella parte superiore della console di gestione dell'ambiente.

L'immagine seguente mostra il riquadro di panoramica dell'ambiente.



Integrità

Lo stato generale dell'ambiente. Se lo stato dell'ambiente peggiora, accanto allo stato di integrità dell'ambiente sarà visualizzato il collegamento Visualizza cause. Seleziona questo link per visualizzare la scheda Integrità con maggiori dettagli.

Dominio

Il dominio, o URL, dell'ambiente si trova nella parte superiore della pagina Environment overview (Panoramica dell'ambiente), sotto la voce Health (Integrità) dell'ambiente. Si tratta dell'URL dell'applicazione Web eseguita dall'ambiente. È possibile avviare l'applicazione selezionando l'URL.

ID ambiente

L'ID dell'ambiente. Si tratta di un ID interno che viene generato al momento della creazione dell'ambiente.

Application name (Nome applicazione)

Il nome dell'applicazione implementata e in esecuzione nell'ambiente.

Versione in esecuzione

Nome della versione dell'applicazione distribuita e in esecuzione nell'ambiente. Selezionare Upload and Deploy (Carica e distribuisci) per caricare un bundle di origine e distribuirlo al tuo ambiente. Questa opzione crea una nuova versione dell'applicazione.

Platform (Piattaforma)

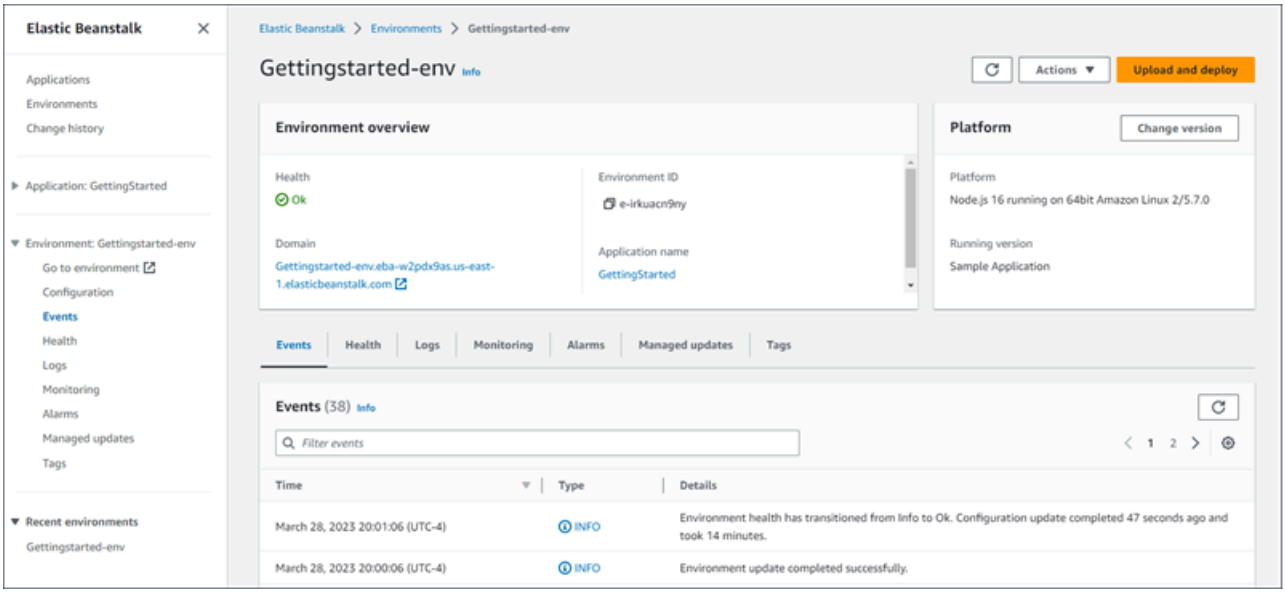
Il nome della versione della piattaforma in esecuzione nel proprio ambiente. In genere, questo comprende l'architettura, il sistema operativo (OS), il linguaggio e il server delle applicazioni (noti collettivamente come ramo della piattaforma), con un numero specifico di versione della piattaforma.

Se la versione della piattaforma non è la più recente disponibile, accanto ad essa viene visualizzata un'etichetta di stato nella sezione Piattaforma. L'etichetta Aggiorna indica che, sebbene la versione della piattaforma sia supportata, è disponibile una versione più recente. La versione della piattaforma può anche essere etichettata come obsoleta o ritirata. Seleziona Cambia versione per aggiornare la ramificazione della piattaforma a una versione più recente. Per ulteriori informazioni sugli stati di una versione della piattaforma, consulta la sezione Ramificazione della piattaforma nel Glossario delle piattaforme Elastic Beanstalk. L'immagine precedente in questa pagina illustra l'etichetta di stato dell'aggiornamento per la piattaforma specificata.

Dettagli sull'ambiente

Questo argomento descrive le informazioni aggiuntive fornite dalla console di gestione dell'ambiente dal riquadro di navigazione a sinistra e dalle pagine a schede.

L'immagine seguente illustra la console di gestione dell'ambiente.



Nella parte inferiore della console di gestione dell'ambiente sono elencate le schede che forniscono informazioni più dettagliate e varie sull'ambiente. È possibile selezionare la scheda o l'etichetta della pagina dal riquadro di navigazione a sinistra.

Nel riquadro di navigazione a sinistra della console, sotto il nome dell'ambiente, sono disponibili due scelte che non si trovano nelle pagine a schede. Queste sono Vai all'ambiente e alla configurazione.

> **Note**
>
> Seleziona Vai all'ambiente per avviare l'applicazione.

Configurazione

Utilizza la pagina Configurazione nel riquadro di navigazione a sinistra per visualizzare e aggiornare le impostazioni di configurazione correnti per l'ambiente e le relative risorse. Ciò include la configurazione di rete, la configurazione del database, il bilanciamento del carico, le notifiche, le impostazioni di monitoraggio dello stato, la configurazione degli aggiornamenti della piattaforma gestita, la configurazione della distribuzione, lo streaming dei log delle istanze, CloudWatch l'integrazione AWS X-Ray, le impostazioni del server proxy, le proprietà dell'ambiente e le opzioni specifiche della piattaforma. Utilizza le impostazioni in questa pagina per personalizzare il comportamento del tuo ambiente durante le distribuzioni, abilitare le caratteristiche aggiuntive e modificare il tipo di istanza e altre impostazioni scelte durante la creazione dell'ambiente.

Per ulteriori informazioni, consulta Configurazione degli ambienti Elastic Beanstalk.

Eventi

Nella pagina Events (Eventi) viene mostrato il flusso degli eventi per il tuo ambiente. Elastic Beanstalk invia messaggi di evento ogni volta che interagisci con l'ambiente e quando, in seguito a tale situazione, una qualunque risorsa del tuo ambiente viene creata o modificata.

Per ulteriori informazioni, consulta Visualizzazione del flusso di eventi di un ambiente Elastic Beanstalk.

Integrità

Se è abilitato il monitoraggio avanzato dello stato, questa pagina elenca le istanze EC2 presenti nell'ambiente e le informazioni sullo stato in tempo reale per ciascuna istanza.

La pagina sullo stato generale mostra i dati sullo stato di salute come media per tutte le istanze dell'ambiente combinate.

Il riquadro Enhanced Instance Health mostra informazioni in tempo reale sullo stato di ogni singola istanza EC2 presente nell'ambiente. Il monitoraggio avanzato dello stato consente a Elastic Beanstalk di monitorare con attenzione le risorse nel tuo ambiente in modo da valutare lo stato della tua applicazione in modo più accurato.

Quando il monitoraggio avanzato dello stato è abilitato, in questa pagina vengono mostrate informazioni sulle richieste inoltrate dalle istanze nel tuo ambiente e sui parametri provenienti dal sistema operativo, inclusi latenza, caricamento e utilizzo della CPU.

Per ulteriori informazioni, consulta Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk.

Log

La pagina Logs (Log) consente di recuperare i log dalle istanze EC2 nel proprio ambiente. Quando richiedi i log, Elastic Beanstalk invia un comando alle istanze che quindi caricano i log nel tuo bucket di storage Elastic Beanstalk in Amazon S3. Quando richiedi i log in questa pagina, Elastic Beanstalk li elimina automaticamente da Amazon S3 dopo 15 minuti.

Puoi inoltre configurare le istanze del tuo ambiente per caricare i log in Amazon S3 per uno storage permanente dopo che sono stati ruotati localmente.

Per ulteriori informazioni, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Monitoraggio

La pagina Monitoring (Monitoraggio) mostra una panoramica dello stato dell'ambiente. Sono inclusi il set predefinito di parametri fornito da Elastic Load Balancing e Amazon EC2 e i grafici che mostrano il modo in cui lo stato dell'ambiente è cambiato nel corso del tempo.

Per ulteriori informazioni, consulta Monitoraggio dello stato dell'ambiente nella console AWS di gestione.

Allarmi

Nella pagina Existing Alarms (Allarmi esistenti) vengono mostrate le informazioni su eventuali allarmi che hai configurato per il tuo ambiente. Puoi utilizzare le opzioni in questa pagina per creare o eliminare gli allarmi.

Per ulteriori informazioni, consulta Gestione degli allarmi.

Aggiornamenti gestiti

La pagina Aggiornamenti gestiti mostra informazioni sugli aggiornamenti imminenti e completati della piattaforma gestita e sulla sostituzione delle istanze.

La funzione di aggiornamento gestito consente di configurare il tuo ambiente per l'aggiornamento automatico all'ultima versione della piattaforma durante una finestra di manutenzione settimanale a tua scelta. Nel periodo che intercorre tra due versioni della piattaforma, puoi scegliere che l'ambiente sostituisca tutte le istanze Amazon EC2 durante il periodo di manutenzione. Ciò può alleviare i problemi che si verificano quando l'applicazione viene eseguita per lunghi periodi di tempo.

Per ulteriori informazioni, consulta Aggiornamenti gestiti della piattaforma.

Tag

Nella pagina Tags (Tag) vengono mostrati i tag che Elastic Beanstalk ha applicato all'ambiente al momento della sua creazione. È possibile aggiungere, modificare ed eliminare tag personalizzati. Non è possibile modificare o eliminare i tag applicati da Elastic Beanstalk.

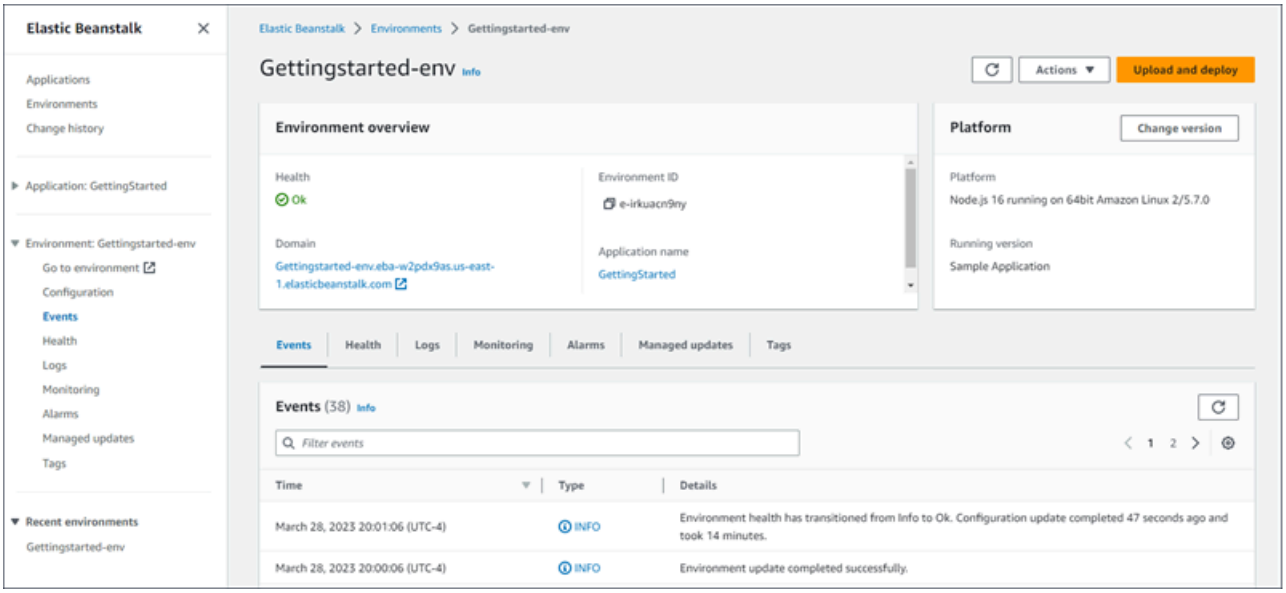
Questi tag vengono applicati a ogni risorsa creata da Elastic Beanstalk per supportare l'applicazione.

Per ulteriori informazioni, consulta Tagging delle risorse negli ambienti Elastic Beanstalk.

Operazioni ambiente

Questo argomento descrive le operazioni comuni che è possibile scegliere di eseguire sull'ambiente dal menu a discesa Azioni nella console di gestione dell'ambiente.

L'immagine seguente illustra la console di gestione dell'ambiente. Il menu a discesa Azioni si trova sul lato destro dell'intestazione che mostra il nome dell'ambiente, accanto al pulsante Aggiorna.



> **Note**
>
> Alcune operazioni sono disponibili solo in determinate condizioni, restando disabilitate fino a quando non vengono soddisfatte le giuste condizioni.

Load Configuration (Carica configurazione)

Consente di caricare una configurazione salvata in precedenza. Le configurazioni vengono salvate nella tua applicazione e possono essere caricate da qualsiasi ambiente associato. Se hai apportato modifiche alla configurazione del tuo ambiente, puoi caricare una configurazione salvata per annullare tali modifiche. Puoi anche caricare una configurazione salvata in precedenza da un altro ambiente che esegue la stessa applicazione per propagare le modifiche di configurazione tra loro.

Salva la configurazione

Consente di salvare l'attuale configurazione dell'ambiente nella tua applicazione. Prima di apportare modifiche alla configurazione dell'ambiente, salva la configurazione attuale in modo che sia possibile

eseguire il rollback in un secondo momento, se necessario. Inoltre, puoi applicare una configurazione salvata all'avvio di un nuovo ambiente.

Scambia domini dell'ambiente (URL)

Scambia il CNAME dell'ambiente attuale con un nuovo ambiente. Dopo uno scambio CNAME, tutto il traffico verso l'applicazione che utilizza l'URL dell'ambiente accederà al nuovo ambiente. Quando sei pronto per distribuire una nuova versione della tua applicazione, puoi avviare un ambiente diverso nell'ambito della nuova versione. Quando il nuovo ambiente è pronto per iniziare ad accettare richieste, esegui uno scambio CNAME per iniziare a instradare il traffico verso il nuovo ambiente. In questo modo non interrompe i tuoi servizi. Per ulteriori informazioni, consulta Distribuzioni blu/verde con Elastic Beanstalk.

Ambiente di clonazione

Avvia un nuovo ambiente con la stessa configurazione dell'ambiente attualmente in esecuzione.

Clone with Latest Platform (Clona con la piattaforma più recente)

Clona il tuo attuale ambiente con la versione più recente della piattaforma Elastic Beanstalk in uso. Questa opzione è disponibile solo quando una versione più recente della piattaforma dell'ambiente attuale è disponibile per l'uso.

Abort Current Operation (Interrompi operazione corrente)

Interrompe un aggiornamento dell'ambiente in corso. L'interruzione di un'operazione può far sì che alcune delle istanze nel tuo ambiente si trovino in uno stato diverso rispetto ad altre, a seconda dell'avanzamento dell'operazione. Questa opzione è disponibile solo quando il tuo ambiente viene aggiornato.

Restart app servers (Riavvia server app)

Riavvia il server Web in esecuzione sulle istanze del tuo ambiente. Questa opzione non interrompe o riavvia alcuna AWS risorsa. Se il tuo ambiente mostra uno strano comportamento in risposta ad alcune richieste non valide, il riavvio del server dell'applicazione ti consente di ripristinare temporaneamente la funzionalità mentre cerchi di comprendere e risolvere la causa principale del problema.

Rebuild environment (Ricrea ambiente)

Termina tutte le risorse nell'ambiente in esecuzione e crea un nuovo ambiente con le stesse impostazioni. Questa operazione richiede diversi minuti, approssimativamente quanto il tempo necessaria per la distribuzione di un nuovo ambiente da zero. Qualsiasi istanza Amazon RDS in esecuzione al livello dati del tuo ambiente viene eliminata durante una ricompilazione. Se hai bisogno dei dati, crea uno snapshot. Puoi creare uno snapshot manualmente nella console RDS o configurare la policy di eliminazione del livello dati per creare uno snapshot automaticamente prima di eliminare l'istanza. Questa è l'impostazione predefinita quando crei un livello dati.

Terminate environment (Termina ambiente)

Termina tutte le risorse nell'ambiente in esecuzione e rimuovi l'ambiente dall'applicazione. Se disponi di un'istanza RDS in esecuzione in un livello dati e devi conservarne i dati, assicurati che la policy di eliminazione del database è impostata sia su `Snapshot` che su `Retain`. Per ulteriori informazioni, consultare Ciclo di vita del database nel capitolo Configurazione degli ambienti di questa guida.

Creazione di un ambiente Elastic Beanstalk

La procedura seguente avvia un nuovo ambiente che esegue l'applicazione predefinita. La procedura è semplificata per consentirti di avere un ambiente immediatamente operativo utilizzando i valori delle opzioni predefinite.

> **Nota sulle autorizzazioni**
>
> Per la creazione di un ambiente sono necessarie le autorizzazioni della policy gestita di accesso completo di Elastic Beanstalk. Per informazioni dettagliate, consulta Policy utente Elastic Beanstalk.

Per avviare un ambiente con un'applicazione (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applications (Applicazioni). Seleziona un'applicazione esistente nell'elenco. Puoi anche scegliere di crearne una seguendo le istruzioni riportate inGestione delle applicazioni .
3. Nella pagina di panoramica dell'applicazione, scegli Crea ambiente.

Verrà avviata la procedura guidata Create environment (Creazione dell'ambiente). La procedura guidata fornisce una procedura che consente di creare un nuovo ambiente.

4. Per il livello Ambiente, scegli il livello Ambiente server Webo Ambiente di lavoro. Non è possibile modificare un piano di un ambiente dopo la creazione.

> **Note**
>
> La piattaforma .NET su Windows Server non supporta il piano di ambiente operatore.

I campi di informazioni sull'applicazione sono predefiniti, in base all'applicazione scelta in precedenza.

Nel gruppo di informazioni sull'ambiente, il valore predefinito del nome dell'ambiente è basato sul nome dell'applicazione. Se preferisci un nome di ambiente diverso, puoi inserire un altro valore nel campo. Facoltativamente, puoi inserire un nome di dominio; in caso contrario Elastic Beanstalk genera automaticamente un valore. Facoltativamente, puoi anche inserire una descrizione dell'ambiente.

5. Per Piattaforma, selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.

> **Note**
>
> Elastic Beanstalk supporta più versioni per la maggior parte delle piattaforme elencate. Per impostazione predefinita, la console seleziona la versione consigliata per il ramo della piattaforma e per la piattaforma di scelta. Se l'applicazione richiede una versione diversa, puoi selezionarla qui. Per informazioni sulle versioni della piattaforma supportate, consulta the section called “Piattaforme supportate”.

6. Per il codice dell'applicazione, sono disponibili diverse scelte per procedere.
   - Per avviare l'applicazione di esempio predefinita senza fornire il codice sorgente, scegliete Applicazione di esempio. Questa azione sceglie l'applicazione a pagina singola fornita da Elastic Beanstalk per la piattaforma selezionata in precedenza.
   - Se hai scaricato un'applicazione di esempio da questa guida o disponi del tuo codice sorgente per un'applicazione, procedi nel seguente modo.

   a. Seleziona Carica il tuo codice.
   b. Quindi scegli File locale, quindi in Carica applicazione, seleziona Scegli file.
   c. Il sistema operativo del computer client ti presenterà un'interfaccia per selezionare il file locale che hai scaricato. Seleziona il file del pacchetto sorgente e continua.

7. La scelta delle impostazioni predefinite dipende dallo scopo dell'ambiente.
   - Se stai creando un ambiente di esempio per conoscere Elastic Beanstalk o un ambiente di sviluppo, scegli Istanza singola (idoneo al piano gratuito).
   - Se stai creando un ambiente di produzione o un ambiente per saperne di più sul bilanciamento del carico, scegli una delle opzioni di alta disponibilità.
8. Scegli Next (Successivo).

Per configurare l'accesso al servizio

Successivamente, sono necessari due ruoli. Un ruolo di servizio consente a Elastic Beanstalk di monitorare le istanze EC2 e aggiornare la piattaforma dell'ambiente. Un ruolo di profilo di istanza EC2 consente attività come la scrittura di log e l'interazione con altri servizi.

Per creare o selezionare il ruolo Service

1. Se hai già creato un ruolo di servizio e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa Ruolo di servizio e salta il resto di questi passaggi per creare un ruolo di servizio.
2. Se non vedi alcun valore elencato per il ruolo di servizio o desideri crearne uno nuovo, continua con i passaggi successivi.
3. Per Ruolo di servizio, scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Environment.
6. Scegli Next (Successivo).
7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`
8. Scegli Crea ruolo.

9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il ruolo di servizio appena creato.

Per creare o selezionare un profilo di istanza EC2

1. Se hai già creato un profilo di istanza EC2 e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa del profilo dell'istanza EC2 e salta il resto di questi passaggi per creare un profilo di istanza EC2.
2. Se non vedi alcun valore elencato per il profilo dell'istanza EC2 o desideri crearne uno nuovo, continua con i passaggi successivi.
3. Scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Compute.
6. Scegli Next (Successivo).
7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
8. Scegli Crea ruolo.
9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il profilo di istanza EC2 appena creato.

Per completare la configurazione e la creazione dell'applicazione

1. (Facoltativo) Se in precedenza hai creato una coppia di chiavi EC2, puoi selezionarla dal menu a discesa del campo delle coppie di chiavi EC2. Puoi utilizzarlo per accedere in modo sicuro all'istanza Amazon EC2 fornita da Elastic Beanstalk per la tua applicazione. Se salti questo passaggio, puoi sempre creare e assegnare una coppia di key pair EC2 dopo la creazione dell'ambiente. Per ulteriori informazioni, consulta Coppia di chiavi EC2.
2. Scegli Skip to Review (Passa alla revisione) nella pagina Configure service access (Configura l'accesso al servizio).
3. La pagina Review (Revisione) mostra un riepilogo di tutte le tue scelte.

Per personalizzare ulteriormente l'ambiente, scegli Edit (Modifica) accanto al passaggio che include tutti gli elementi che desideri configurare. Le seguenti opzioni possono essere impostate solo durante la creazione dell'ambiente:

- Nome ambiente
- Nome dominio
- Versione della piattaforma
- Processore
- Nuovo tipo di load balancer
- Livello

Le seguenti impostazioni possono essere modificate dopo la creazione dell'ambiente, ma richiedono nuove istanze o altre risorse per il provisioning e i tempi di applicazione possono essere lunghi:

- Tipo di istanza, volume root, key pair e ruolo AWS Identity and Access Management (IAM)
- Database interno Amazon RDS
- VPC

Per ulteriori dettagli sulle impostazioni disponibili, consulta Procedura guidata per la creazione del nuovo ambiente.

4. Scegli Submit (Invia) nella parte inferiore della pagina per inizializzare la creazione del nuovo ambiente.

Mentre Elastic Beanstalk crea l'ambiente, vieni reindirizzato alla console Elastic Beanstalk. Quando lo stato dell'ambiente diventa verde, selezionare l'URL accanto al nome dell'ambiente per visualizzare l'applicazione in esecuzione. Questo URL è in genere accessibile da Internet, a meno che non configuri l'ambiente per l'utilizzo di un VPC personalizzato con un load balancer interno.

Argomenti

- Procedura guidata per la creazione del nuovo ambiente
- Clonazione di un ambiente Elastic Beanstalk
- Terminazione di un ambiente Elastic Beanstalk

- Creazione di ambienti Elastic Beanstalk con la CLI AWS
- Creazione di ambienti Elastic Beanstalk con l'API
- Creazione di un URL Launch Now
- Creazione e aggiornamento di gruppi di ambienti Elastic Beanstalk

Procedura guidata per la creazione del nuovo ambiente

Questo argomento descrive la procedura guidata Create environment (Creazione ambiente) e tutti i modi in cui è possibile usarla per configurare l'ambiente desiderato.

> **Note**
>
> Qui Creazione di un ambiente Elastic Beanstalk mostriamo come avviare la procedura guidata Crea ambiente e creare rapidamente un ambiente con valori predefiniti e impostazioni consigliate. Questo argomento attuale ti illustrerà tutte le opzioni.

Pagina della procedura guidata

La procedura guidata Create environment (Crea ambiente) fornisce una serie di passaggi che consentono di creare un nuovo ambiente.

Step 1
Configure environment

Step 2
Configure service access

Step 3 - *optional*
Configure instance traffic and scaling

Step 4 - *optional*
Set up networking, database, and tags

Step 5 - *optional*
Configure updates, monitoring, and logging

Step 6
Review

Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- (●) Web server environment
  Run a website, web application, or web API that serves HTTP requests. Learn more
- ( ) Worker environment
  Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more

Application information Info

Application name

GettingStarted

Maximum length of 100 characters.

▶ Application tags (optional)

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

GettingStarted-env

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain name

Leave blank for autogenerated value .us-east-1.elasticbeanstalk.com Check availability

Environment description

Platform Info

Platform type

- (●) Managed platform
  Platforms published and maintained by Amazon Elastic Beanstalk. Learn more
- ( ) Custom platform
  Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Choose a platform ▼

Platform branch

Choose a platform branch ▼

Platform version

Choose a platform version ▼

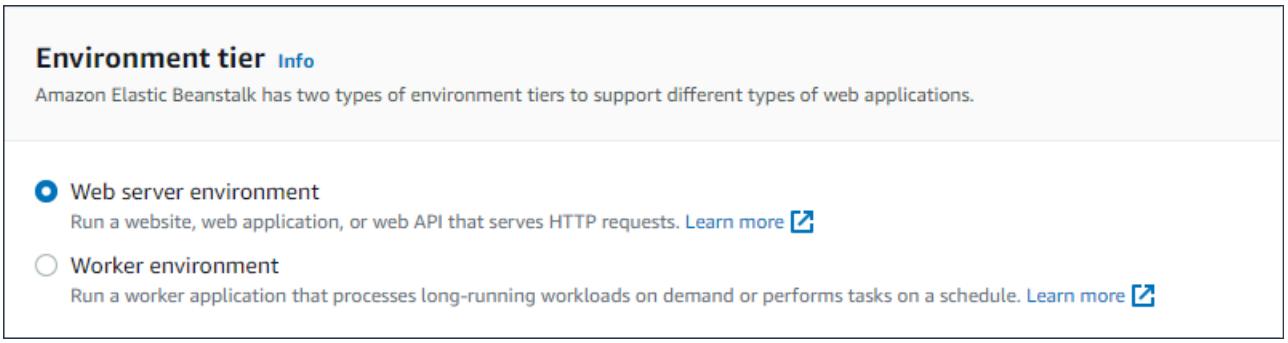
Application code Info

- (●) Sample application
- ( ) Existing version
  Choose an application version that you have uploaded.

  Sample Application ▼
- ( ) Upload your code
  Upload a source bundle from your computer or copy one from Amazon S3.

Livello ambientale

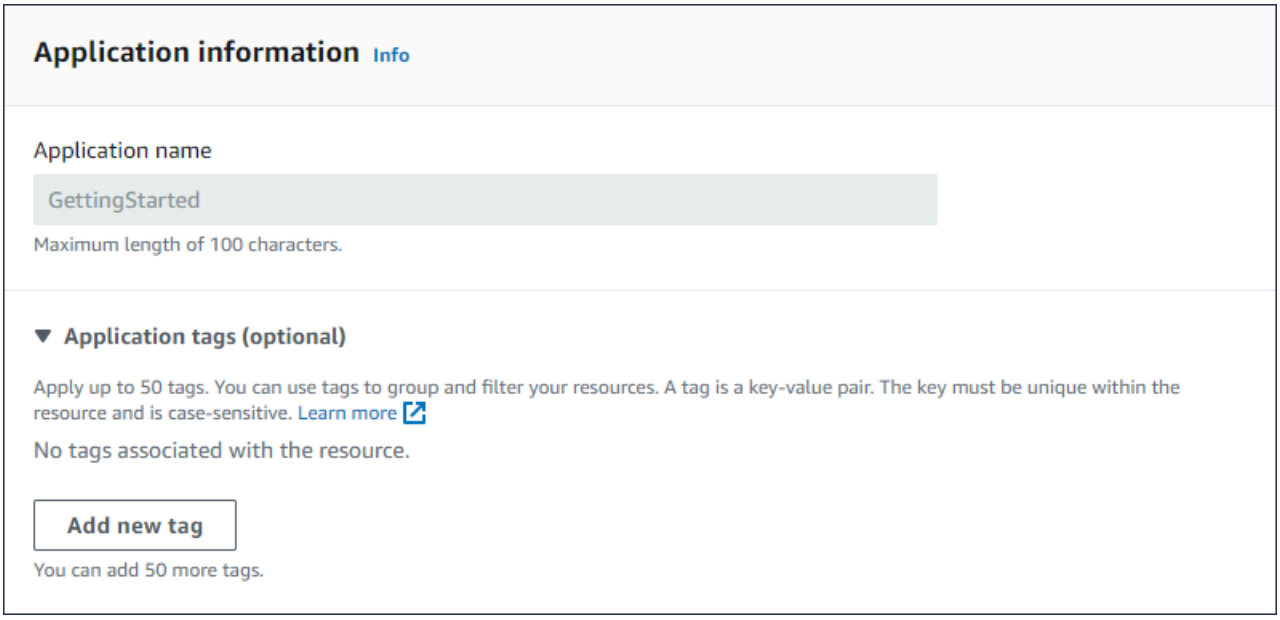
Per Environment tier (Livello ambiente), scegli il livello di ambiente Web server environment (Ambiente server Web) o Worker environment (Ambiente worker). Non è possibile modificare un piano di un ambiente dopo la creazione.



> **Note**
>
> La piattaforma .NET su Windows Server non supporta il piano di ambiente operatore.

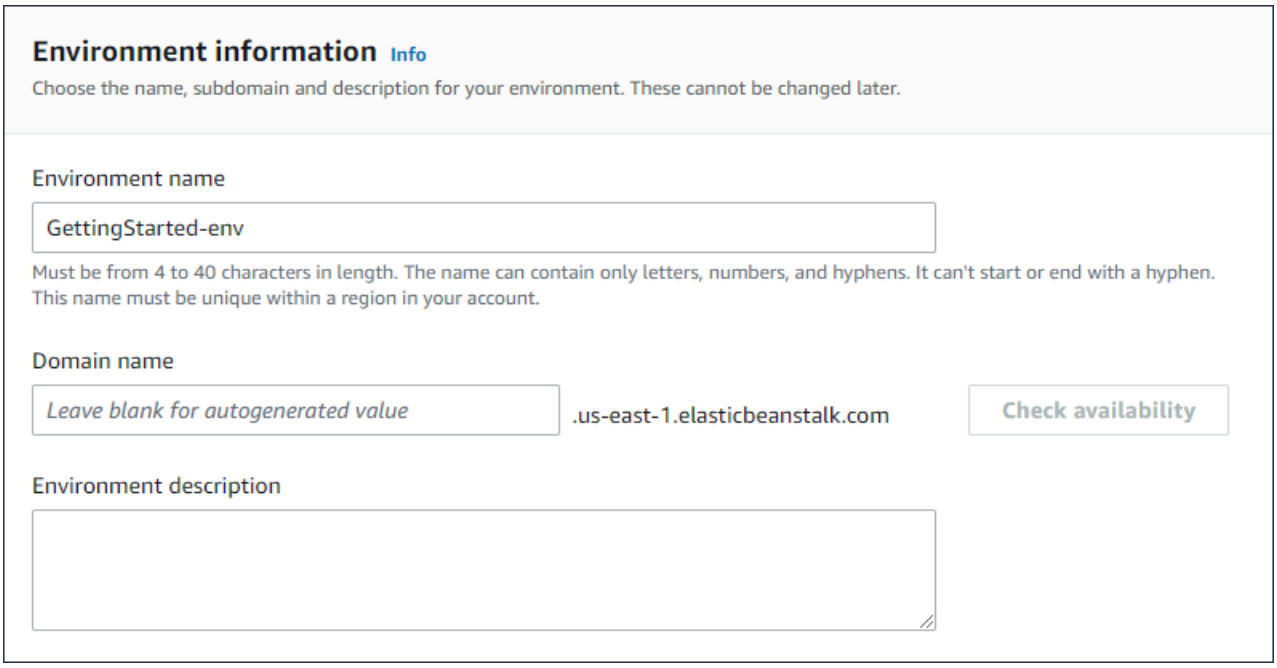
Informazioni sull'applicazione

Se hai avviato la procedura guidata selezionando Create new environment (Crea nuovo ambiente) dalla pagina di Application overview (Panoramica dell'applicazione), il campo Application name è pre-compilato. Altrimenti, inserisci il nome dell'applicazione. Facoltativamente aggiungi tag dell'applicazione.



Informazioni sull'ambiente

Imposta il nome e il dominio dell'ambiente, quindi crea una descrizione. Ricorda che non è possibile modificare queste impostazioni una volta creato l'ambiente.



- Name (Nome): immetti un nome per l'ambiente. Il modulo fornisce un nome generato.
- Domain (Dominio): (ambienti server Web) immetti un nome di dominio univoco per l'ambiente. Il nome predefinito è il nome dell'ambiente. ma puoi sceglierne uno diverso. Elastic Beanstalk utilizza questo nome per creare un CNAME univoco per l'ambiente. Per verificare se il nome di dominio è disponibile, scegli Check Availability (Controlla disponibilità).
- Description (Descrizione): immetti una descrizione per l'ambiente.

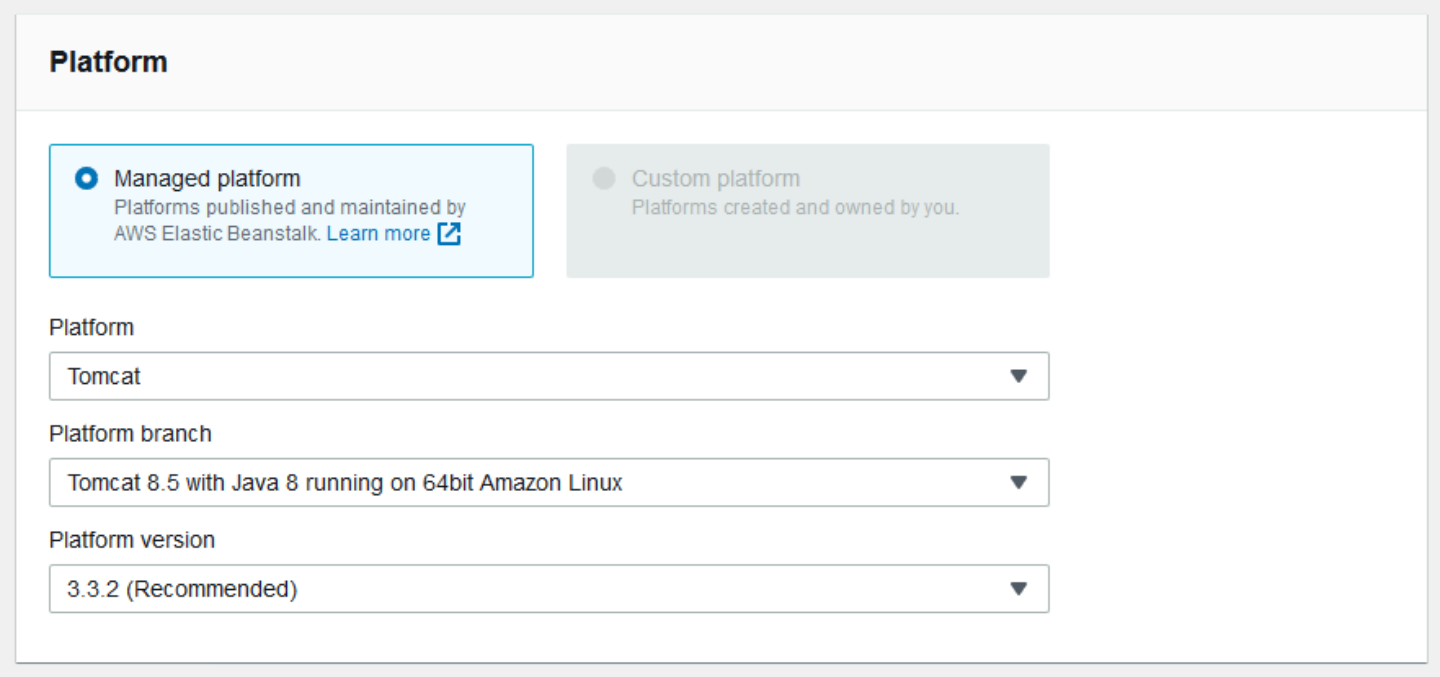
Selezione di una piattaforma per il nuovo ambiente

Puoi creare un nuovo ambiente in base a due tipi di piattaforme:

- Piattaforma gestita
- Piattaforma personalizzata

Piattaforma gestita

Nella maggior parte dei casi utilizzi una piattaforma gestita Elastic Beanstalk per il nuovo ambiente. All'avvio della procedura guidata del nuovo ambiente, seleziona l'opzione Managed platform (Piattaforma gestita) per impostazione predefinita.



Selezionare una piattaforma, un ramo di piattaforma all'interno di tale piattaforma e una versione della piattaforma specifica nel ramo. Quando si seleziona un ramo di piattaforma, la versione consigliata all'interno del ramo viene selezionata per impostazione predefinita. Inoltre, è possibile selezionare qualsiasi versione della piattaforma utilizzata in precedenza.

> **Note**
>
> Per un ambiente di produzione, si consiglia di scegliere una versione della piattaforma in un ramo di piattaforma supportato. Per informazioni dettagliate sugli stati dei rami della piattaforma, consulta la definizione di ramo della piattaforma nel the section called “Glossario delle piattaforme”.

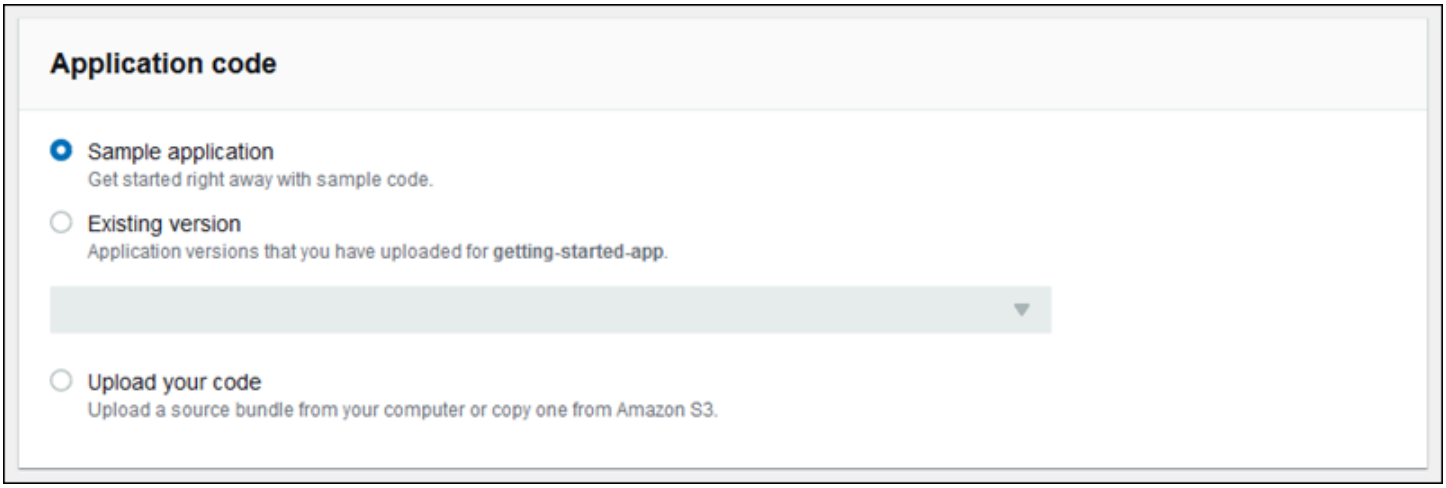
Piattaforma personalizzata

Se una piattaforma preconfigurata non risponde alle tue esigenze, puoi creare un nuovo ambiente a partire da una piattaforma personalizzata. Per specificare una piattaforma personalizzata, scegliere l'opzione Custom platform (Piattaforma personalizzata), quindi selezionare una delle piattaforme

personalizzate disponibili. Se non sono disponibili piattaforme personalizzate, questa opzione non è selezionabile.

Fornisci il codice dell'applicazione

Dopo aver selezionato la piattaforma da utilizzare, è necessario fornire il codice dell'applicazione.

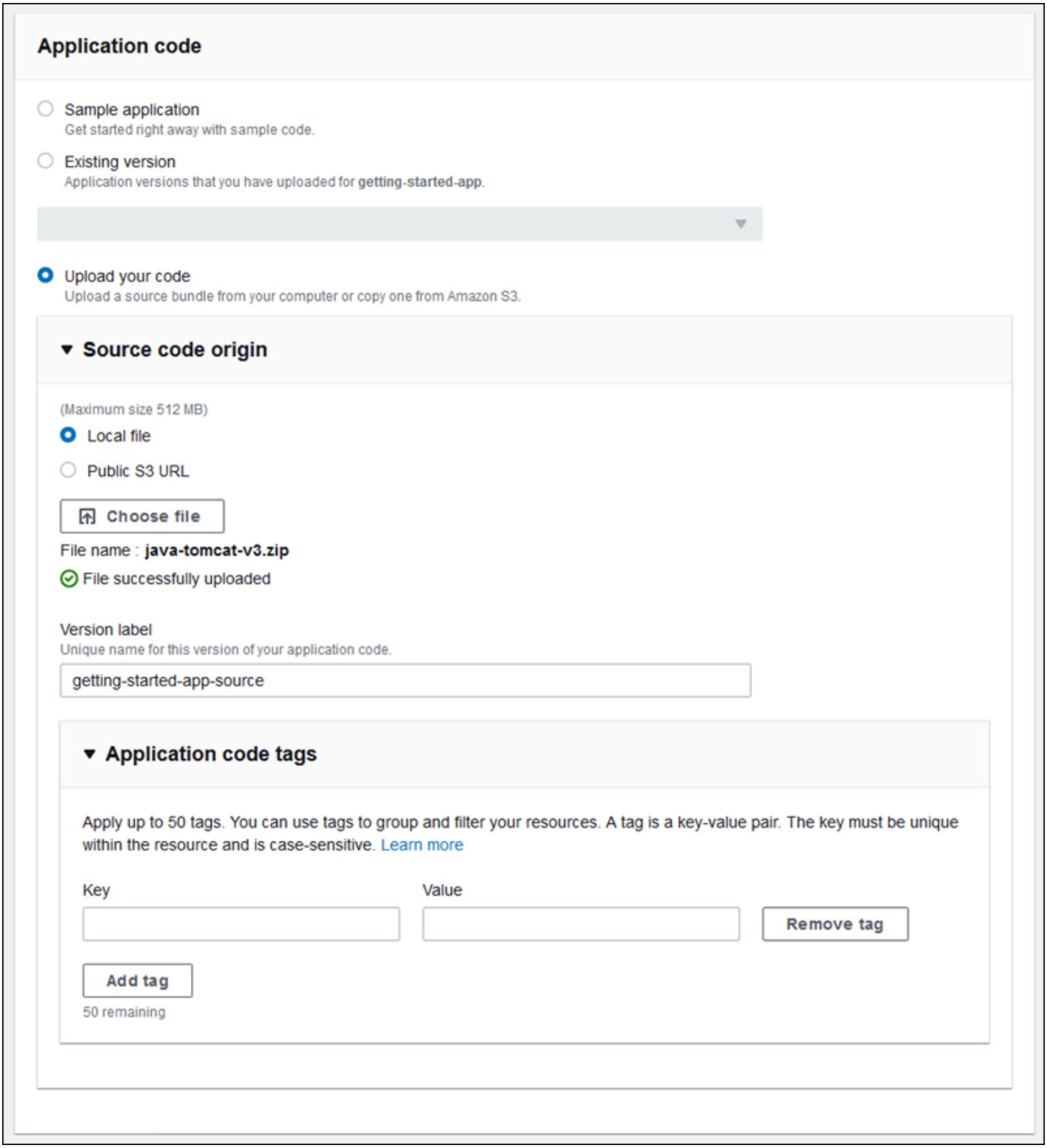


A questo scopo, sono disponibili numerose opzioni:

- Puoi utilizzare l'applicazione di esempio fornita da Elastic Beanstalk per ogni piattaforma.
- Puoi utilizzare il codice che hai già distribuito su Elastic Beanstalk. Scegli Existing version (Versione esistente) e l'applicazione nella sezione Application code (Codice applicazione).
- Puoi caricare un nuovo codice. Scegli Upload your code (Carica il tuo codice), quindi Upload (Carica). Puoi caricare il nuovo codice dell'applicazione da un file locale oppure specificare l'URL del bucket Amazon S3 contenente il codice.

> **Note**
>
> A seconda della versione della piattaforma selezionata, puoi caricare la tua applicazione come un bundle di origine in formato ZIP, un file WAR o una configurazione Docker in testo normale. Il limite della dimensione del file è 500 MB.

Quando scegli di caricare nuovo codice, puoi anche fornire i tag da associare a esso. Per ulteriori informazioni sull'assegnazione di tag di una versione dell'applicazione, consulta the section called “Tagging delle versioni delle applicazioni”.


Application code
Sample application
Get started right away with sample code.
Existing version
Application versions that you have uploaded for getting-started-app.
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.
Source code origin
(Maximum size 512 MB)
Local file
Public S3 URL
Choose file
File name : java-tomcat-v3.zip
File successfully uploaded
Version label
Unique name for this version of your application code.
getting-started-app-source
Application code tags
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more
Key
Value
Remove tag
Add tag
50 remaining

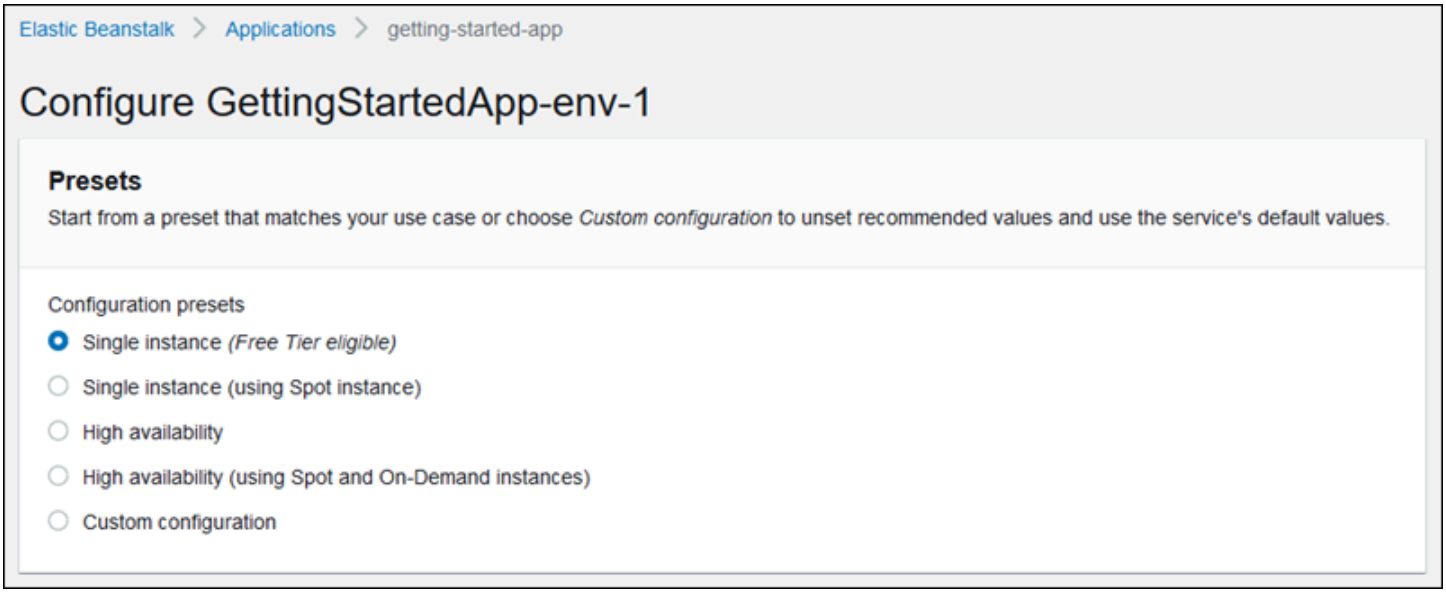
Per la creazione rapida dell'ambiente utilizzando le opzioni di configurazione predefinite, è ora possibile scegliere Create environment (Crea ambiente). Scegli Configure more options (Configura altre opzioni) per apportare altre modifiche di configurazione, come descritto nelle sezioni seguenti.

Pagina di configurazione guidata

Quando si sceglie Configure more options (Configura altre opzioni), la procedura guidata mostra la pagina Configure (Configura). In questa pagina è possibile selezionare una configurazione preimpostata, modificare la versione della piattaforma che si desidera far utilizzare all'ambiente o effettuare scelte di configurazione specifiche per il nuovo ambiente.

Scegliere una configurazione preimpostata

Nella sezione Presets (Impostazioni predefinite) della pagina, Elastic Beanstalk offre diverse impostazioni di configurazione predefinite per i vari casi d'uso. Ogni set include i valori consigliati per varie opzioni di configurazione.



I preset ad alta disponibilità includono un bilanciamento del carico e sono consigliati per gli ambienti di produzione. Scegli questa opzione se desideri un ambiente con carico bilanciato in grado di eseguire più istanze per elevata disponibilità e di evolvere in funzione del carico. I preset a istanza singola sono consigliati principalmente per lo sviluppo. Due dei preset abilitano le richieste di istanza Spot. Per informazioni dettagliate sulla configurazione della capacità di Elastic Beanstalk, consulta Gruppo Auto Scaling.

Il terzo set di impostazioni predefinite, ovvero Custom configuration (Configurazione personalizzata), rimuove tutti i valori consigliati, eccetto le impostazioni relative al ruolo, e utilizza le API predefinite.

Scegli questa opzione se distribuisci un bundle di origine con file di configurazione che definiscono le opzioni di configurazione. L'opzione Custom configuration (Configurazione personalizzata) viene anche selezionata automaticamente se modifichi il set di impostazioni di configurazione predefinite Low cost (Basso costo) or High availability (Alta disponibilità).

Personalizzazione della configurazione

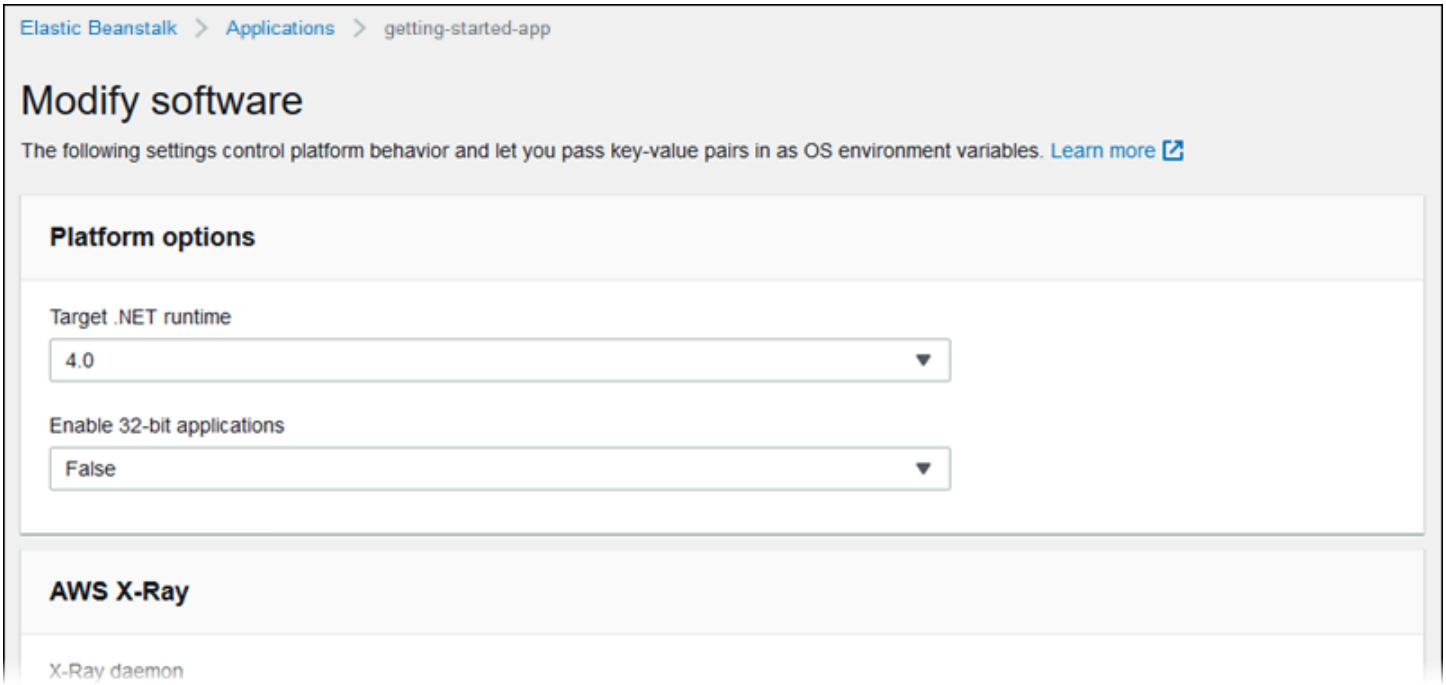
Oltre a (o invece di) scegliere una configurazione preimpostata, puoi perfezionare le opzioni di configurazione dell'ambiente. La procedura guidata Configure (Configura) mostra diverse categorie di configurazione. ognuna delle quali riporta un riepilogo dei valori per un gruppo di impostazioni di configurazione. Scegliere Edit (Modifica) per modificare il gruppo di impostazioni.

Categorie di configurazione

- Impostazioni software
- Istanze
- Capacity
- Sistema di bilanciamento del carico
- Aggiornamenti e distribuzioni continui
- Sicurezza
- Monitoraggio
- Aggiornamenti gestiti
- Notifications
- Rete
- Database
- Tag
- Ambienti worker

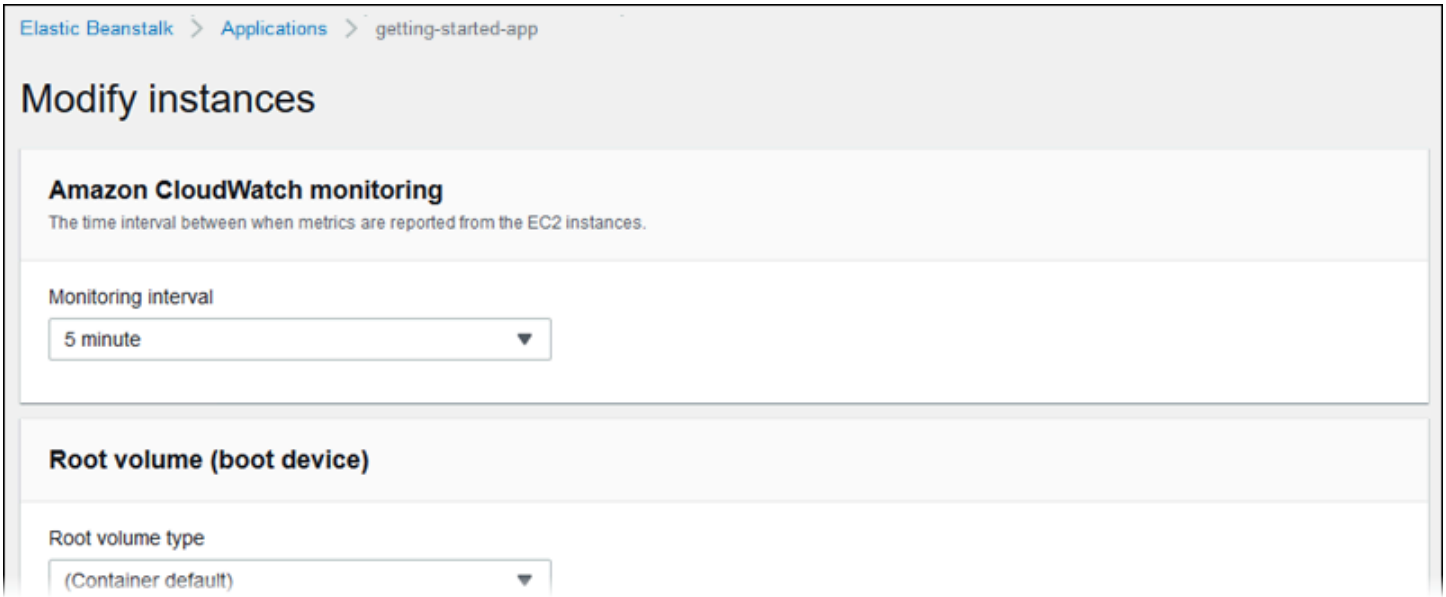
Impostazioni software

Utilizza la pagina di configurazione Modify software (Modifica software) per configurare il software sulle istanze Amazon Elastic Compute Cloud (Amazon EC2) che eseguono l'applicazione. È possibile configurare le proprietà dell'ambiente, il AWS X-Ray debug, l'archiviazione e lo streaming dei log delle istanze e le impostazioni specifiche della piattaforma. Per informazioni dettagliate, vedi the section called “Variabili di ambiente e impostazioni del software”.



Istanze

Utilizza la pagina di configurazione Modify instances (Modifica istanze) per configurare le istanze Amazon EC2 che eseguono l'applicazione. Per informazioni dettagliate, consulta the section called " EC2 Istanze Amazon".



Capacity

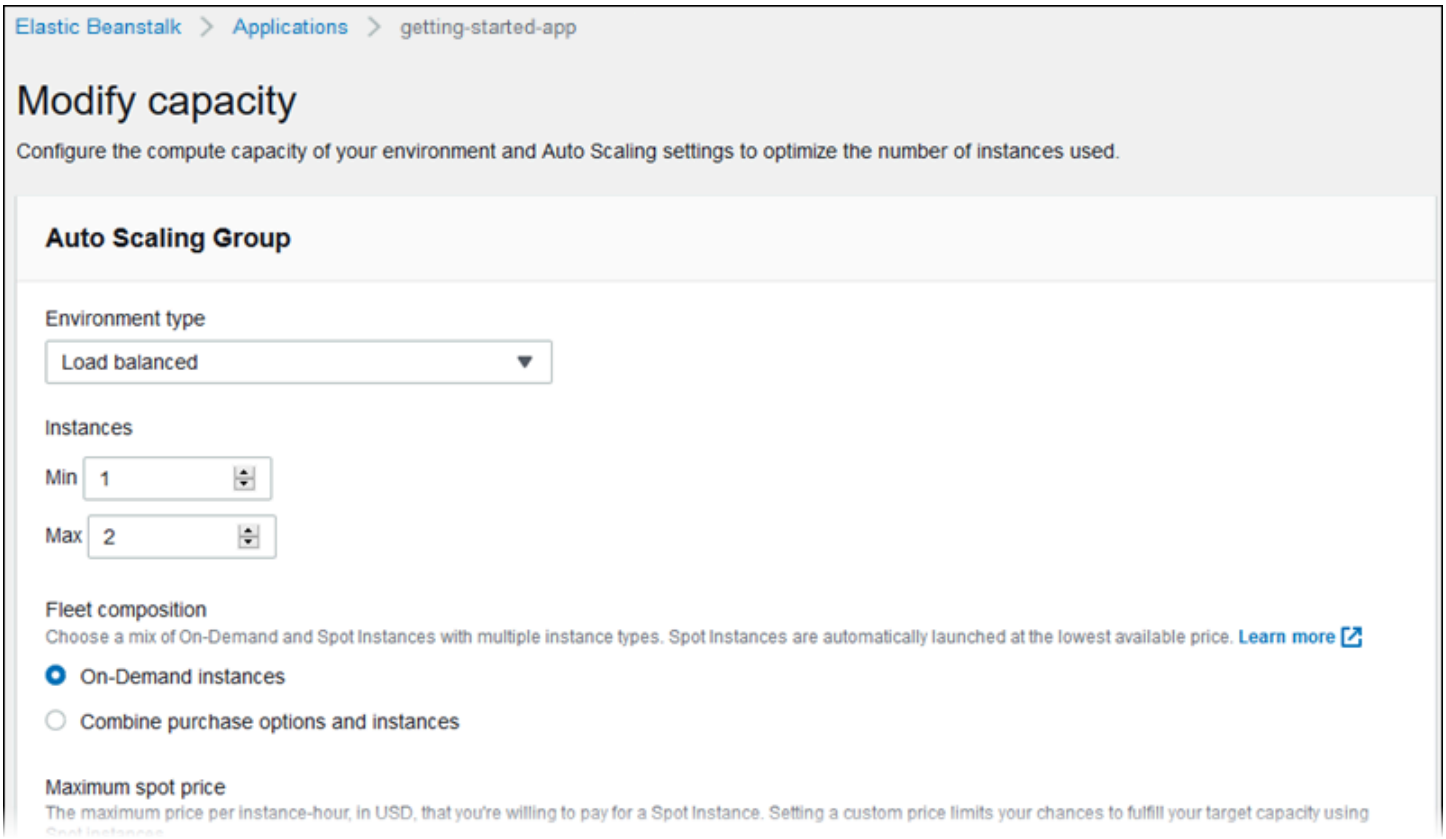
Utilizza la pagina di configurazione Modify capacity (Modifica capacità) per configurare la capacità di elaborazione dell'ambiente e delle impostazioni di Auto Scaling group (Gruppo Auto Scaling) per

ottimizzare il numero e il tipo di istanze in uso. È inoltre possibile modificare la capacità dell'ambiente in base ai trigger o a una pianificazione.

Un ambiente con bilanciamento del carico è in grado di eseguire più istanze per alta disponibilità e impedire tempi di inattività durante gli aggiornamenti della configurazione e le distribuzioni. In un ambiente con bilanciamento del carico il nome di dominio viene mappato al load balancer. In un ambiente a singola istanza viene mappato a un indirizzo IP elastico nell'istanza.

> **Warning**
>
> Un ambiente con una singola istanza non è pronto per la produzione. Se l'istanza diviene instabile durante la distribuzione o Elastic Beanstalk termina e riavvia l'istanza durante un aggiornamento della configurazione, l'applicazione può risultare non disponibile per un certo periodo di tempo. Utilizza ambienti con una singola istanza per sviluppare, testare o gestire temporaneamente. Per la produzione, utilizza ambienti con bilanciamento del carico.

Per ulteriori informazioni sulle impostazioni di capacità dell'ambiente, consulta the section called "Gruppo Auto Scaling" e the section called " EC2 Istanze Amazon".

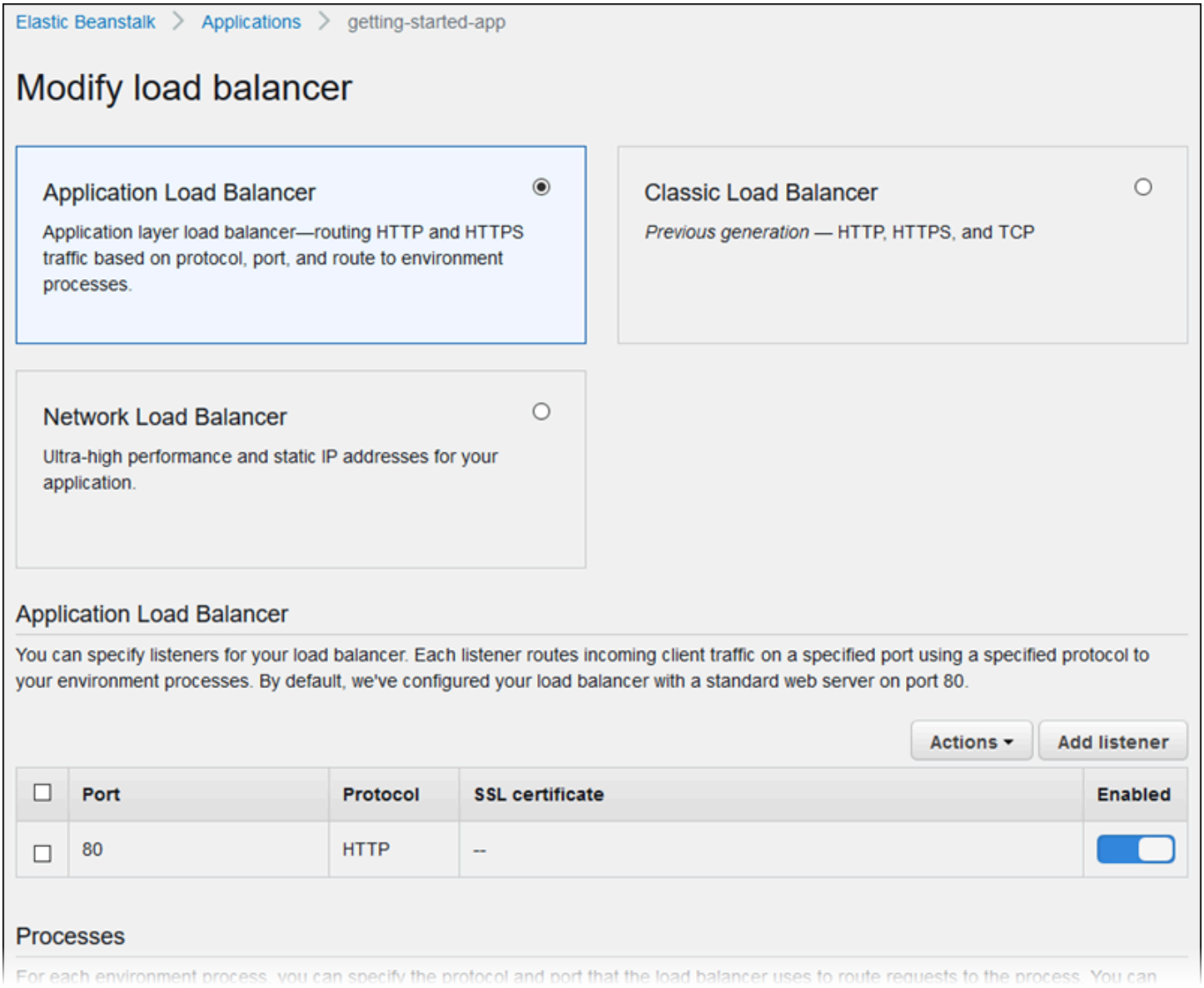


Sistema di bilanciamento del carico

Usa la pagina di configurazione Modify load balancer (Modifica load balancer) per selezionare un tipo di load balancer e configurarne le impostazioni. In un ambiente con bilanciamento del carico il load balancer dell'ambiente è il punto di ingresso per tutto il traffico diretto all'applicazione. Elastic Beanstalk supporta diversi tipi di load balancer. Per impostazione predefinita, la console Elastic Beanstalk crea un Application Load Balancer e lo configura per servire il traffico HTTP sulla porta 80.

> **Note**
>
> È possibile selezionare solo il tipo load balancer dell'ambiente durante la creazione dell'ambiente.

Per ulteriori informazioni sui tipi di load balancer e sulle relative impostazioni, consulta the section called “Bilanciatore del carico” e the section called “HTTPS”.

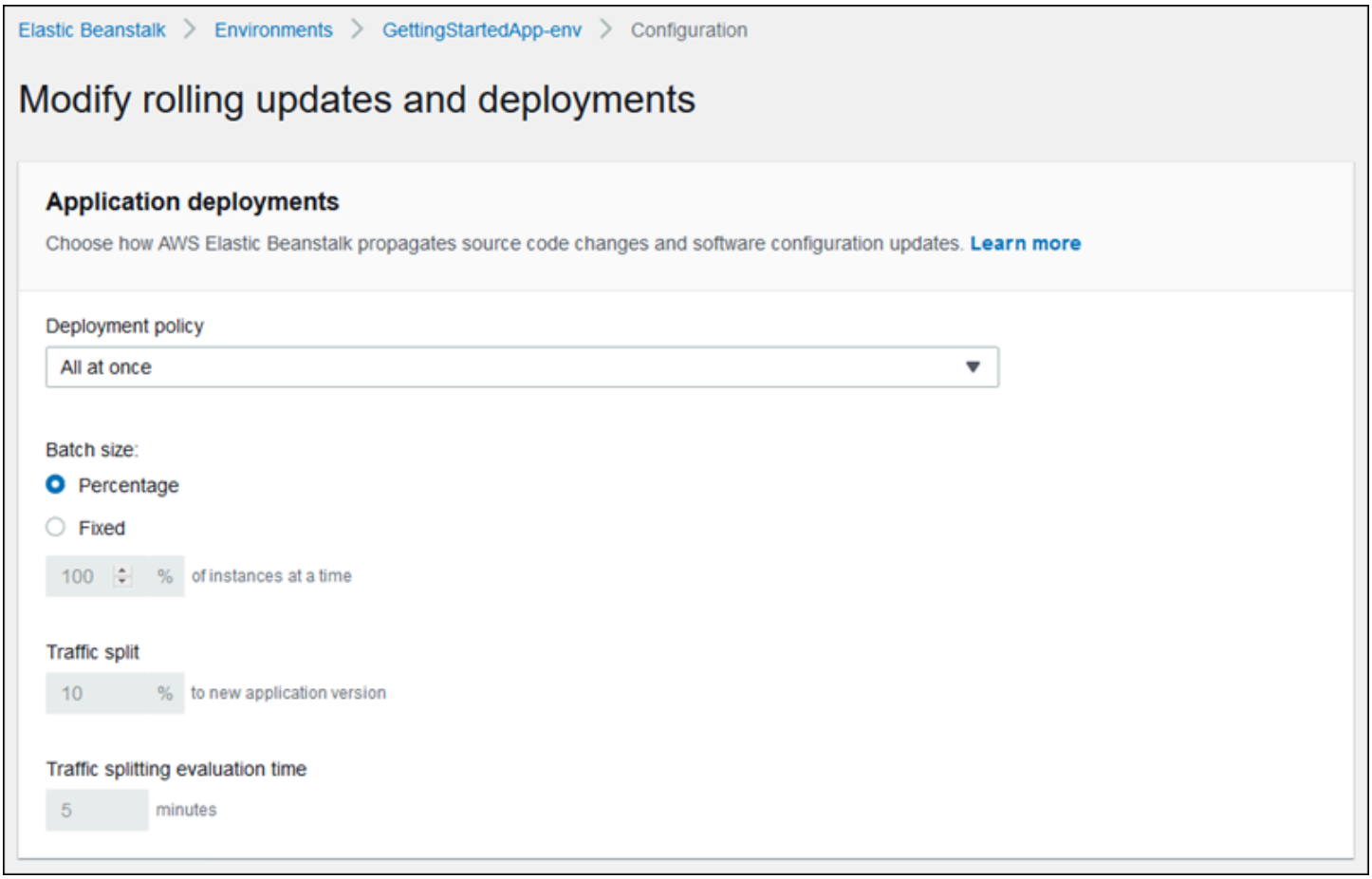


> **Note**
>
> L'opzione Classic Load Balancer (CLB) è disattivata nella procedura guidata della console Crea ambiente. Se disponi di un ambiente esistente configurato con Classic Load Balancer, puoi crearne uno nuovo clonando l'ambiente esistente utilizzando la console Elastic Beanstalk o la CLI EB. È inoltre possibile utilizzare l'EB CLI o AWS CLIcreare un nuovo ambiente configurato con un Classic Load Balancer. Questi strumenti da riga di comando creeranno un nuovo ambiente con un CLB anche se non ne esiste già uno nel tuo account.

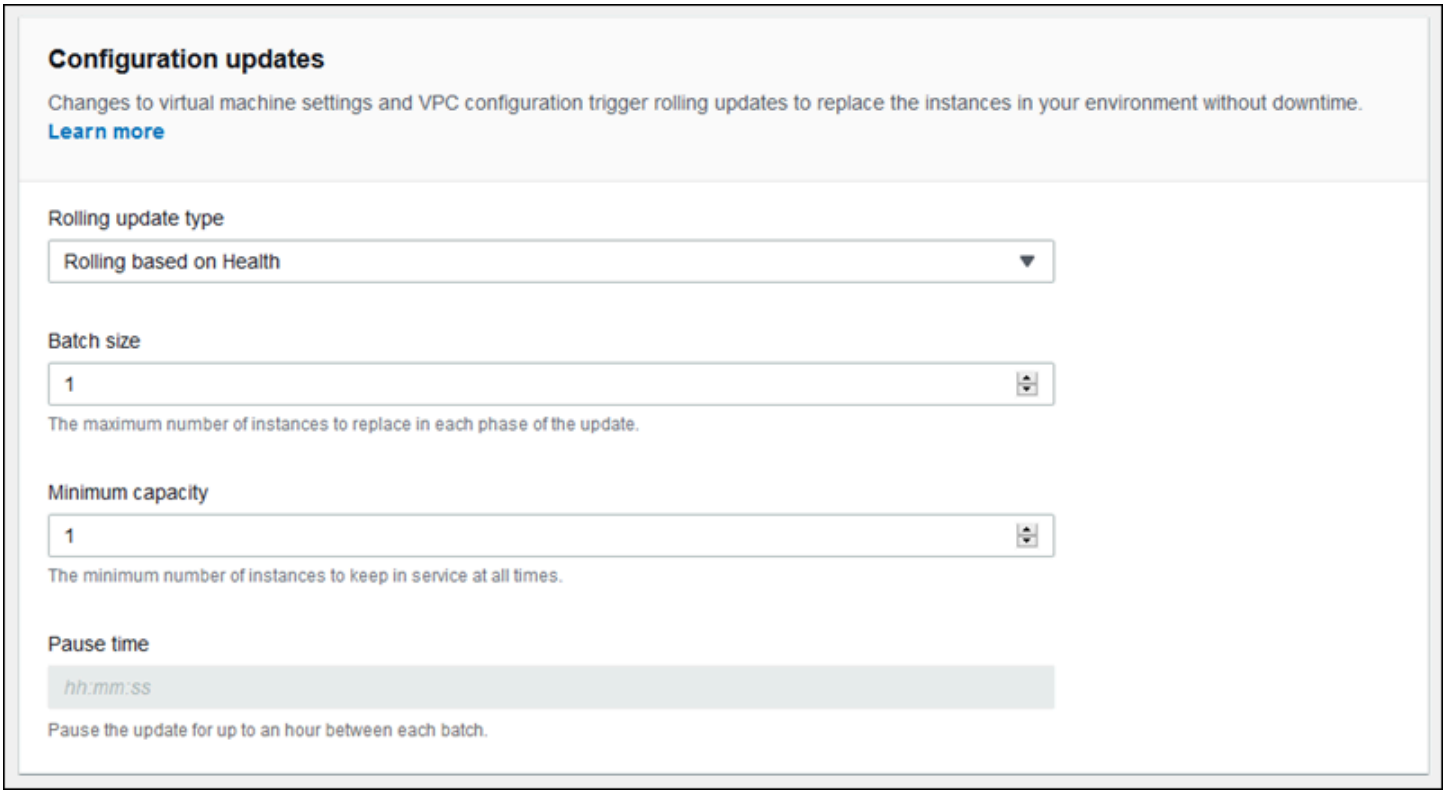
Aggiornamenti e distribuzioni continui

Utilizza la pagina di configurazione Modify rolling updates and deployments (Modifica aggiornamenti e configurazione in sequenza) per configurare il modo in cui Elastic Beanstalk elabora le distribuzioni delle applicazioni e gli aggiornamenti di configurazione per l'ambiente in uso.

Le distribuzioni delle applicazioni si verificano quando si carica un bundle di origine dell'applicazione aggiornato e lo si distribuisce nell'ambiente. Per ulteriori informazioni sulla configurazione delle distribuzioni, consulta the section called “Opzioni di implementazione”.



Le modifiche della configurazione che apportano cambiamenti alla configurazione di avvio o alle impostazioni di VPC richiedono di terminare e sostituire tutte le istanze nel tuo ambiente. Per ulteriori informazioni sull'impostazione del tipo di aggiornamento e di altre opzioni, consulta the section called “Modifiche di configurazione”.



Sicurezza

Utilizza la pagina Configure service access (Configura accesso al servizio) per configurare le impostazioni di sicurezza del servizio e dell'istanza.

Per una descrizione dei concetti di sicurezza di Elastic Beanstalk, consulta Permissions.

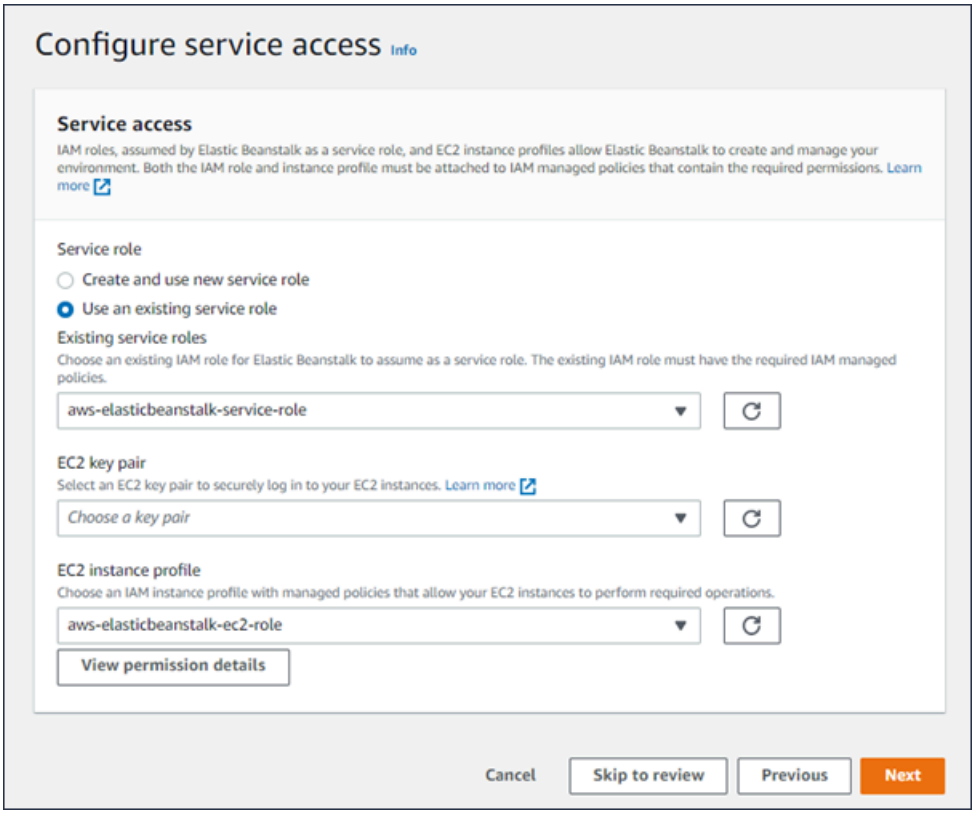
La prima volta che crei un ambiente nella console Elastic Beanstalk, dovrai creare un profilo dell'istanza EC2 con un set predefinito di autorizzazioni. Se l'elenco a discesa Profilo dell'istanza EC2 non riporta alcun valore tra cui scegliere, espandi la procedura che segue. Fornisce i passaggi per creare un ruolo che potrà essere selezionato successivamente per il profilo dell'istanza EC2.

Creazione di un ruolo IAM per il profilo dell'istanza EC2

Per creare o selezionare un profilo di istanza EC2

1. Se hai già creato un profilo di istanza EC2 e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa del profilo dell'istanza EC2 e salta il resto di questi passaggi per creare un profilo di istanza EC2.
2. Se non vedi alcun valore elencato per il profilo dell'istanza EC2 o desideri crearne uno nuovo, continua con i passaggi successivi.

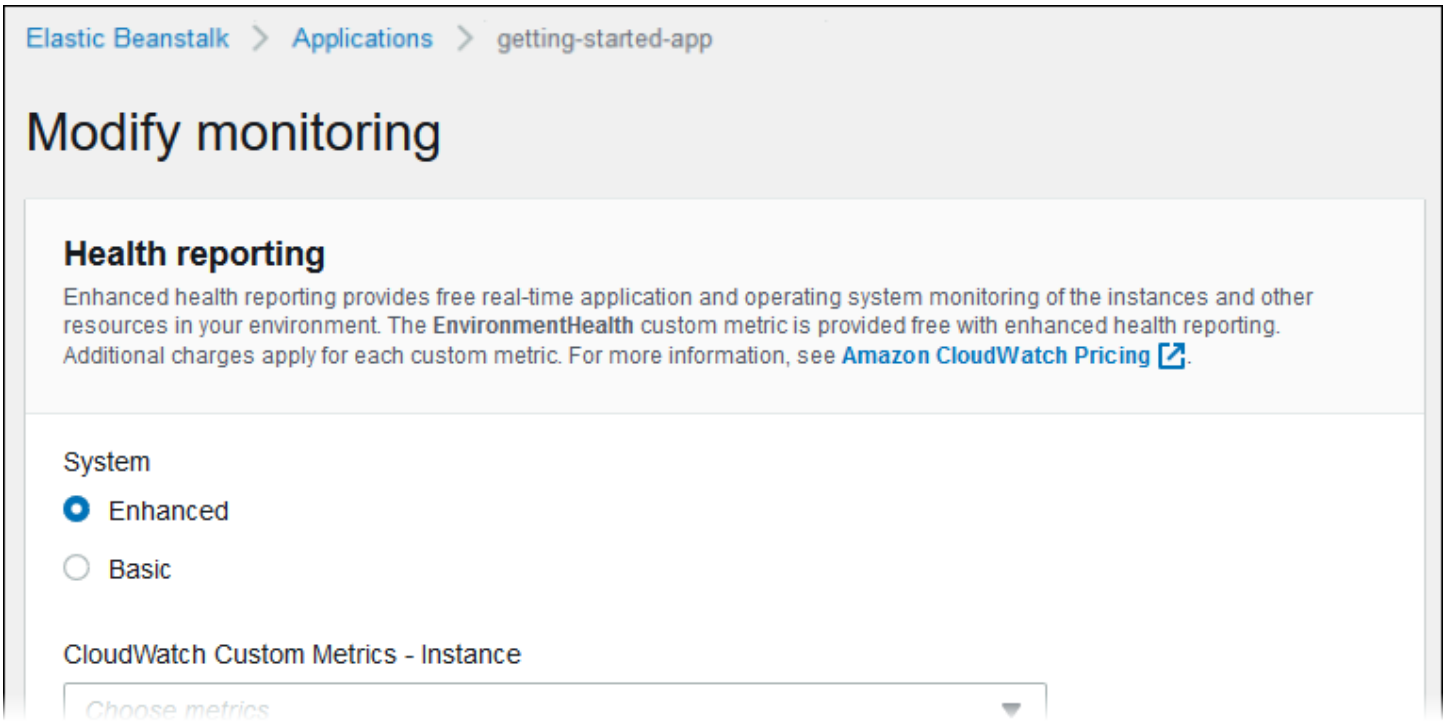
3. Scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Compute.
6. Scegli Next (Successivo).
7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
8. Scegli Crea ruolo.
9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il profilo di istanza EC2 appena creato.



Monitoraggio

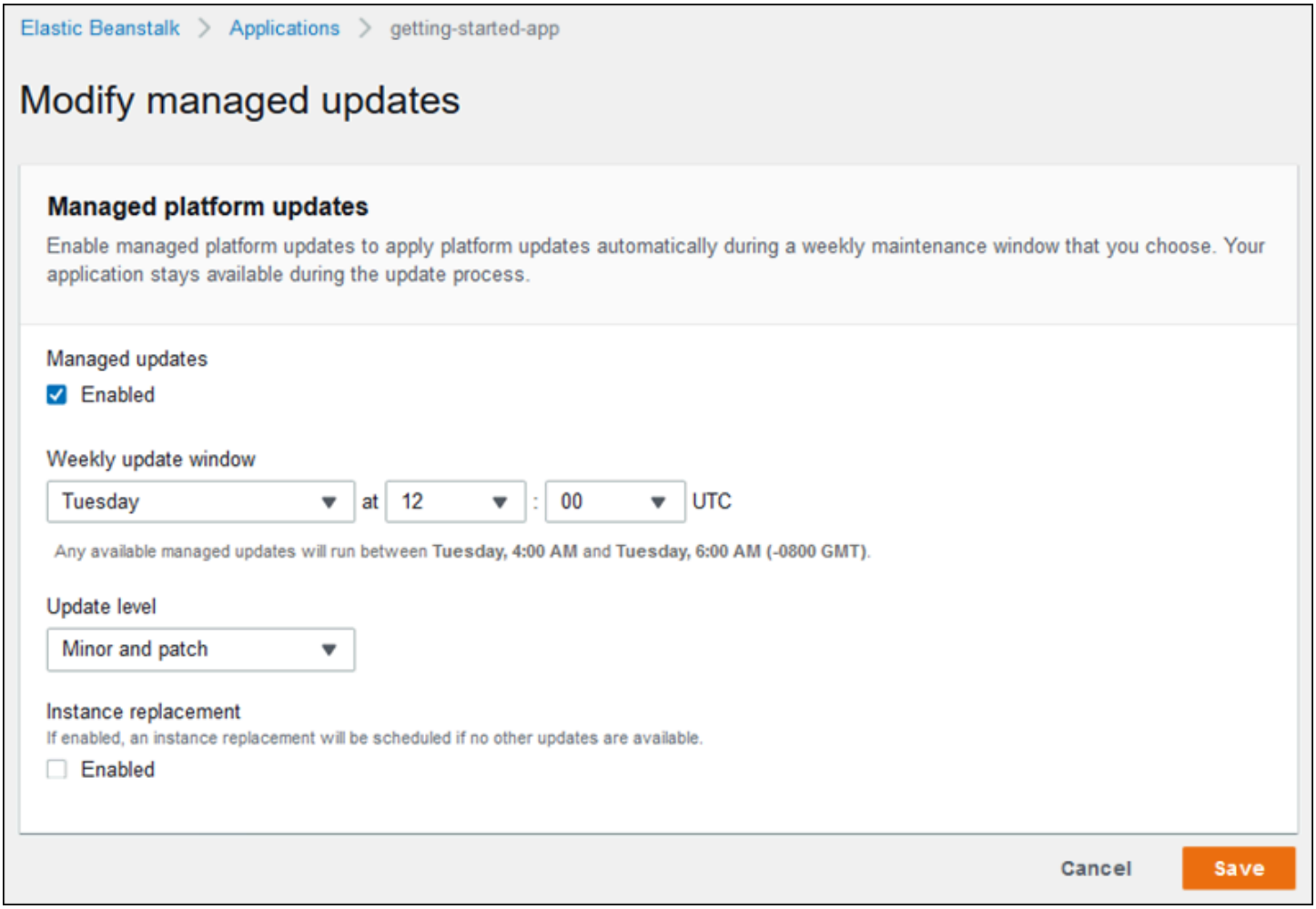
Utilizzare la pagina di configurazione Modify monitoring (Modifica monitoraggio) per configurare report sullo stato, regole di monitoraggio e streaming degli eventi di integrità. Per informazioni

dettagliate, consulta the section called “Abilitazione dello stato avanzato”, the section called “Regole dello stato migliorato” e the section called “Streaming di informazioni sullo stato dell'ambiente”.



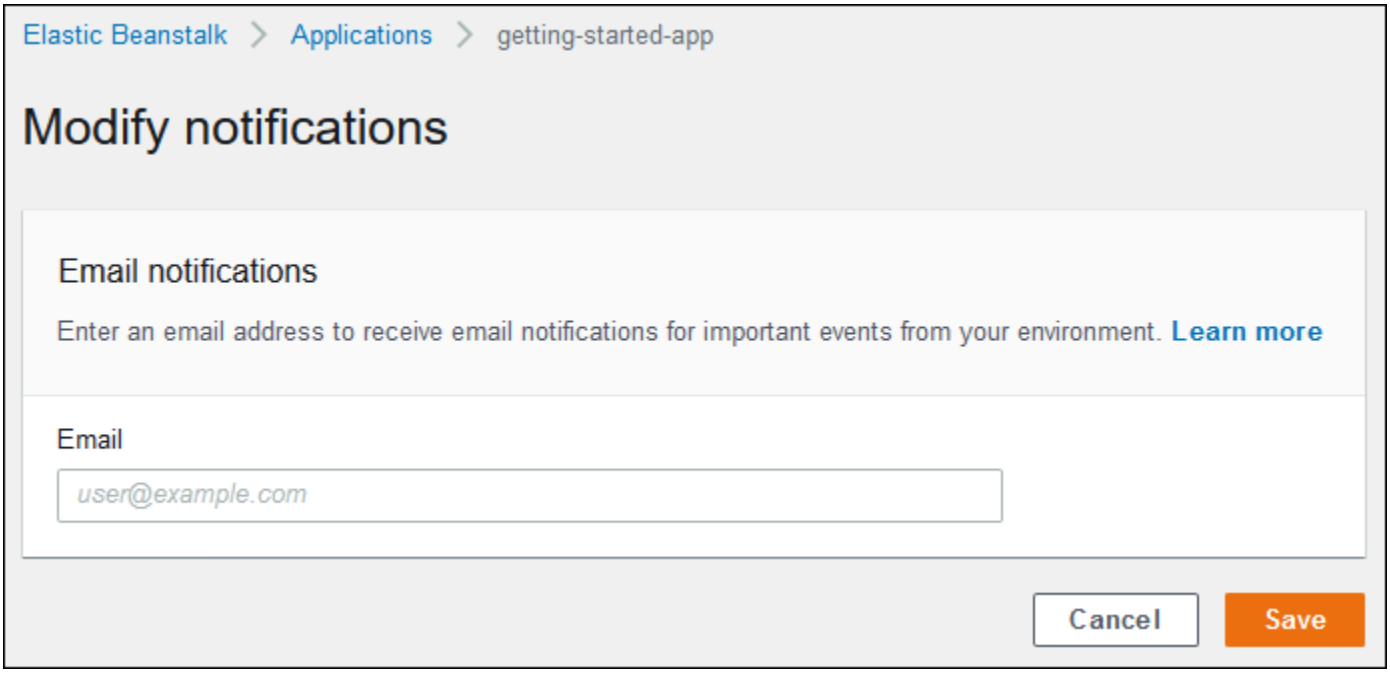
Aggiornamenti gestiti

Utilizzare la pagina di configurazione Modify managed updates (Modifica aggiornamenti gestiti) per configurare gli aggiornamenti della piattaforma gestiti. È possibile decidere se abilitarli, se impostare la pianificazione e se configurare altre proprietà. Per informazioni dettagliate, consulta the section called “Aggiornamenti gestiti”.



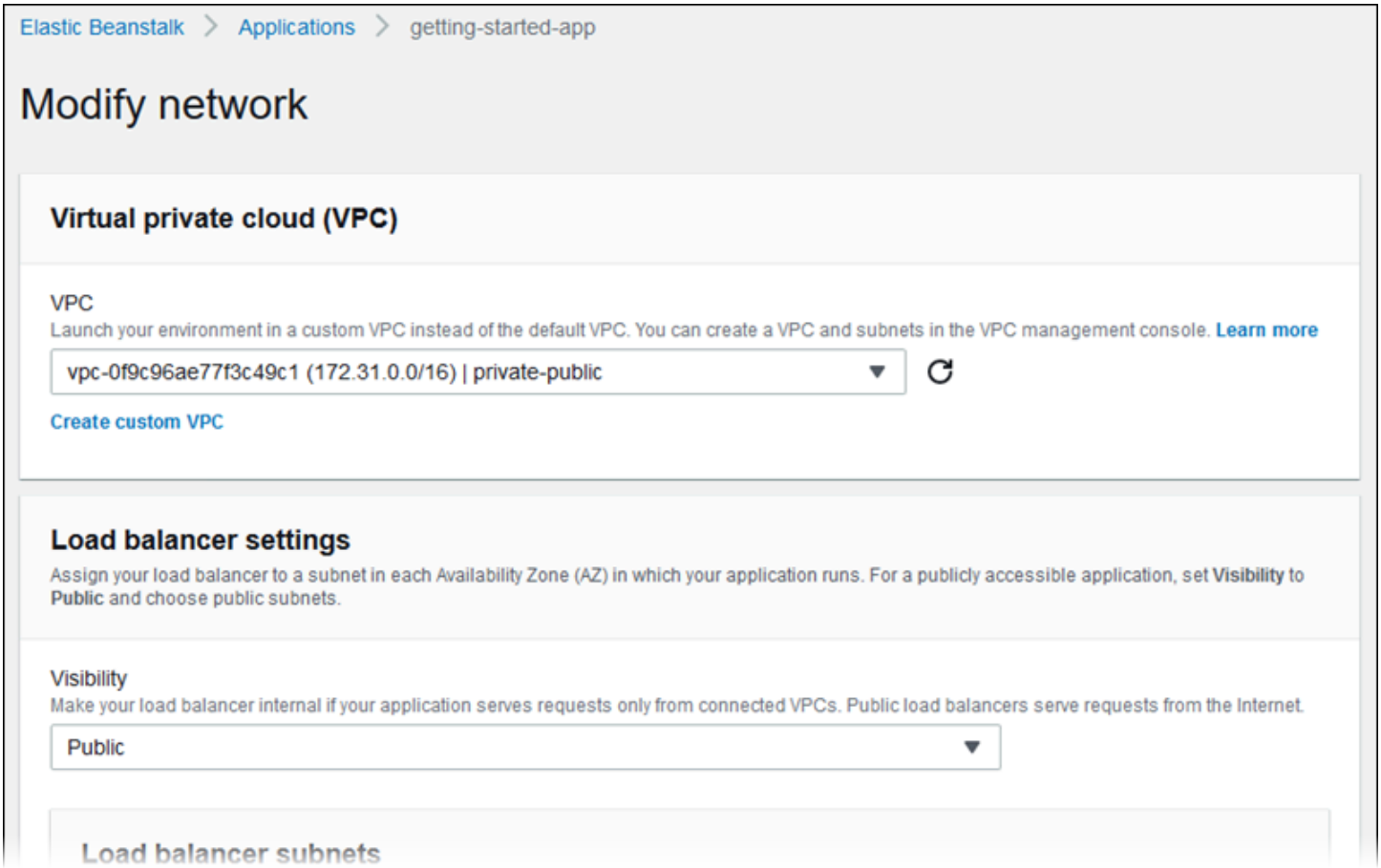
Notifications

Utilizzare la pagina di configurazione Modify notifications (Modifica notifiche) per specificare un indirizzo e-mail per ricevere notifiche e-mail per eventi importanti dall'ambiente.



Rete

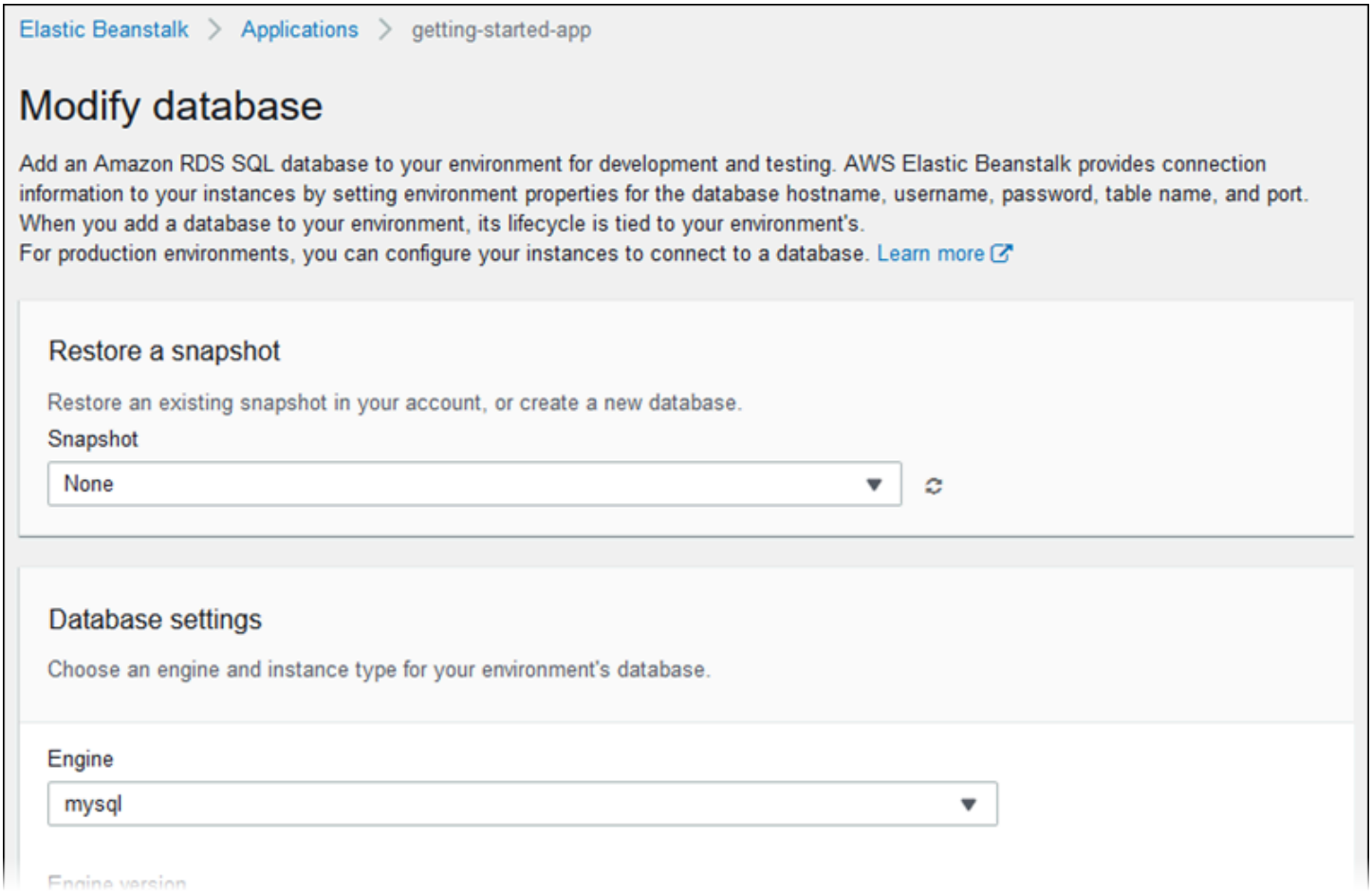
Se è stato creato un VPC personalizzato, la pagina di configurazione Modify network (Modifica rete) consente di configurare l'ambiente per utilizzarlo. Se non scegli un VPC, Elastic Beanstalk utilizza il VPC e le sottoreti predefinite.



Database

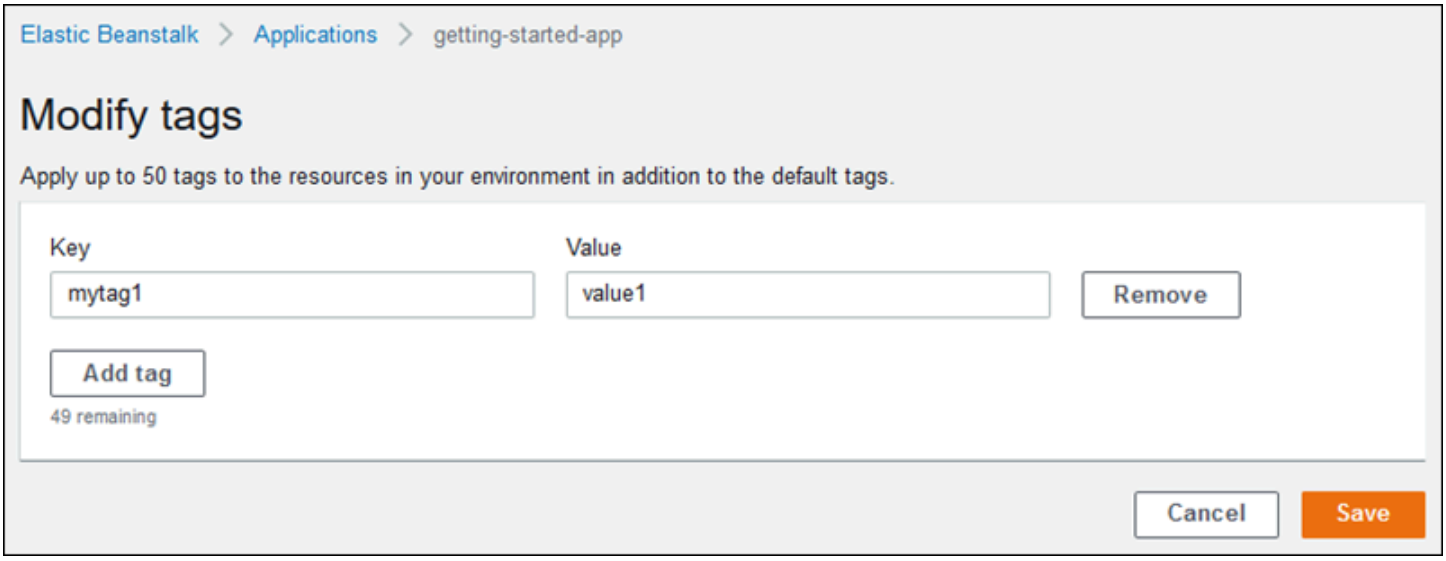
Utilizza la pagina di configurazione Modify database (Modifica database) per aggiungere un database Amazon Relational Database Service (Amazon RDS) al tuo ambiente per lo sviluppo e il test. Elastic Beanstalk fornisce informazioni sulla connessione alle istanze tramite l'impostazione delle proprietà dell'ambiente per nome host, nome utente, password, nome tabella e porta del database.

Per informazioni dettagliate, vedi the section called “Database”.



Tag

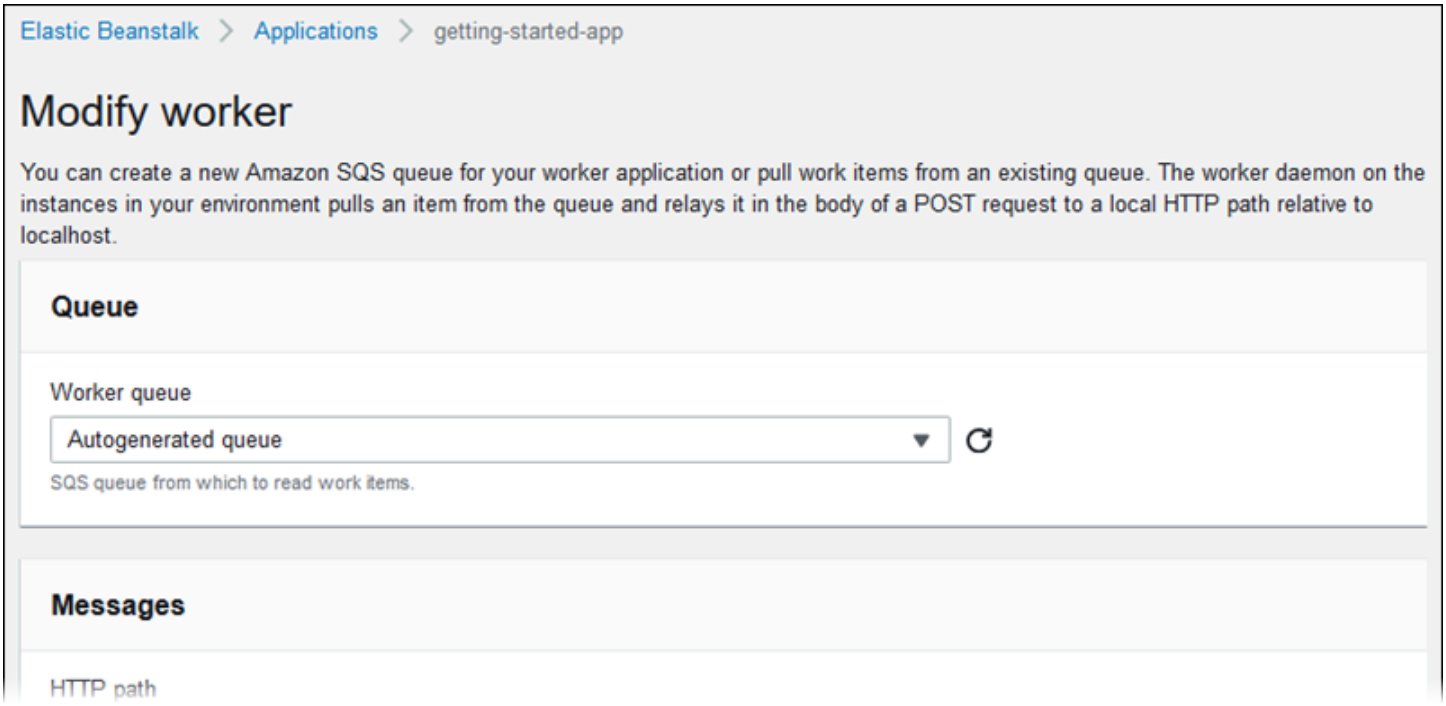
Utilizza la pagina di configurazione Modify tags (Modifica tag) per aggiungere tag alle risorse nell'ambiente. Per ulteriori informazioni sul tagging dell'ambiente, consulta Tagging delle risorse negli ambienti Elastic Beanstalk.



Ambienti worker

Se si sta creando un ambiente a livello di worker, utilizzare la pagina di configurazione Modify worker (Modifica worker) per configurare l'ambiente di lavoro. Il daemon worker nelle istanze dell'ambiente estrae gli elementi da una coda Amazon Simple Queue Service (Amazon SQS) e li inoltra come messaggi di posta all'applicazione worker. È possibile scegliere la coda Amazon SQS da cui il daemon worker legge (generata automaticamente o esistente). È inoltre possibile configurare i messaggi inviati dal daemon worker all'applicazione.

Per ulteriori informazioni, consulta the section called “Ambienti lavoratore”.



Clonazione di un ambiente Elastic Beanstalk

È possibile usare un ambiente Elastic Beanstalk esistente come base per un nuovo ambiente mediante la clonazione dell'ambiente esistente. Ad esempio, puoi creare un clone in modo che sia possibile utilizzare una versione più recente del ramo della piattaforma utilizzato dall'ambiente originale dell'ambiente. Elastic Beanstalk configura il clone con le impostazioni di ambiente utilizzate dall'ambiente originale. Clonando un ambiente esistente anziché creare un nuovo ambiente, non è necessario configurare manualmente le impostazioni delle opzioni, le variabili di ambiente e altre impostazioni effettuate con il servizio Elastic Beanstalk. Elastic Beanstalk crea anche una copia AWS di qualsiasi risorsa associata all'ambiente originale.

È importante essere consapevoli delle seguenti situazioni:

- Durante il processo di clonazione, Elastic Beanstalk non copia i dati da Amazon RDS al clone.
- Elastic Beanstalk non include le modifiche non gestite alle risorse nel clone. Le modifiche apportate alle risorse AWS che esegui utilizzando strumenti diversi dalla console Elastic Beanstalk, dagli strumenti a riga di comando o dalle API sono considerate modifiche non gestite.
- I gruppi di sicurezza per l'ingresso sono considerati modifiche non gestite. Gli ambienti Elastic Beanstalk clonati non trasferiscono i gruppi di sicurezza in ingresso, lasciando l'ambiente aperto a tutto il traffico Internet. Dovrai ristabilire i gruppi di sicurezza in ingresso per l'ambiente clonato.

È possibile clonare un ambiente solo in una versione di piattaforma diversa dello stesso ramo della piattaforma. Un ramo della piattaforma diverso non è garantito per essere compatibile. Per utilizzare un ramo della piattaforma diverso, è necessario creare manualmente un nuovo ambiente, distribuire il codice dell'applicazione e apportare le modifiche necessarie al codice e alle opzioni per garantire che l'applicazione funzioni correttamente sul nuovo ramo della piattaforma.

AWS console di gestione

> **Important**
>
> Gli ambienti Elastic Beanstalk clonati non trasferiscono i gruppi di sicurezza in ingresso, lasciando l'ambiente aperto a tutto il traffico Internet. Dovrai ristabilire i gruppi di sicurezza in ingresso per l'ambiente clonato.
>
> È possibile visualizzare le risorse che potrebbero non essere clonate controllando lo stato di deriva della configurazione dell'ambiente. Per ulteriori informazioni, consulta Rileva la deriva su un intero CloudFormation stack nella Guida per l'utente.AWS CloudFormation

Per clonare un ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina di panoramica dell'ambiente, scegli Operazioni.
4. Scegli Clona ambiente.
5. Nella pagina Clone environment (Clona ambiente), controlla le informazioni nella sezione Original Environment (Ambiente originale) per verificare di aver scelto l'ambiente di cui desideri creare un clone.
6. Nella sezione New Environment (Nuovo ambiente), è possibile eventualmente modificare i valori di Environment name (Nome dell'ambiente), Environment URL (URL dell'ambiente), Description (Descrizione), Platform version (Versione piattaforma) e Service role (Ruolo di servizio) che Elastic Beanstalk imposta automaticamente in base all'ambiente originale.

   > **Note**
   >
   > Se la versione della piattaforma utilizzata nell'ambiente originale non è quella consigliata per l'uso nel ramo della piattaforma, si riceve una notifica che informa che è consigliata una versione diversa della piattaforma. Scegli Platform version (Versione piattaforma) per vedere la versione della piattaforma consigliata nell'elenco, ad esempio 3.3.2 (Recommended) (3.3.2 (versione consigliata)).

7. Quando sei pronto, scegli Clone (Clona).

Interfaccia a riga di comando di Elastic Beanstalk (CLI EB)

> **Important**
>
> Gli ambienti Elastic Beanstalk clonati non trasferiscono i gruppi di sicurezza in ingresso, lasciando l'ambiente aperto a tutto il traffico Internet. Dovrai ristabilire i gruppi di sicurezza in ingresso per l'ambiente clonato.
> È possibile visualizzare le risorse che potrebbero non essere clonate controllando lo stato di deriva della configurazione dell'ambiente. Per ulteriori informazioni, consulta Rileva la deriva su un intero CloudFormation stack nella Guida per l'utente.AWS CloudFormation

Utilizza il comando eb clone per clonare un ambiente in esecuzione nel seguente modo.

```
~/workspace/my-app$ eb clone my-env1
Enter name for Environment Clone
(default is my-env1-clone): my-env2
Enter DNS CNAME prefix
(default is my-env1-clone): my-env2
```

È possibile specificare il nome dell'ambiente di origine nel comando del clone oppure ometterlo per clonare l'ambiente predefinito per la cartella di progetto corrente. La CLI EB richiede di inserire un nome e un prefisso DNS per il nuovo ambiente.

Per impostazione predefinita, eb clone crea il nuovo ambiente con l'ultima versione disponibile della piattaforma dell'ambiente di origine. Per forzare la CLI EB a utilizzare la stessa versione, anche se c'è una versione più recente disponibile, utilizza l'opzione `--exact`.

```
~/workspace/my-app$ eb clone --exact
```

Per ulteriori informazioni su questo comando, consulta eb clone.

Terminazione di un ambiente Elastic Beanstalk

Puoi terminare un AWS Elastic Beanstalk ambiente in esecuzione utilizzando la console Elastic Beanstalk. In questo modo, eviti di incorrere in addebiti per risorse non utilizzate. AWS

> **Note**
>
> Puoi sempre avviare un nuovo ambiente utilizzando la stessa versione in un secondo momento.

Se disponi di dati da un ambiente che desideri conservare, imposta la policy di eliminazione del database su `Retain` prima di terminare l'ambiente. In questo modo, il database rimane operativo al di fuori di Elastic Beanstalk. Dopodiché, qualsiasi ambiente Elastic Beanstalk deve connettersi ad esso come database esterno. Se si desidera eseguire il backup dei dati senza mantenere operativo il database, impostare la policy di eliminazione per fare uno snapshot del database prima di terminare l'ambiente. Per ulteriori informazioni, consultare Ciclo di vita del database nel capitolo Configurazione degli ambienti di questa guida.

Elastic Beanstalk può riscontrare problemi nella chiusura dell'ambiente. Uno dei motivi più comuni è che il gruppo di sicurezza di un altro ambiente ha una relazione con il gruppo di sicurezza dell'ambiente che si desidera terminare. Per istruzioni su come evitare questo problema, consulta la EC2 gruppi di sicurezza pagina EC2Istanze di questa guida.

> **Important**
>
> Se un ambiente viene terminato, è necessario anche eliminare eventuali mappature CNAME create, poiché altri clienti possono riutilizzare un nome host disponibile. Assicurati di eliminare i record DNS che puntano al tuo ambiente terminato per evitare una voce DNS inesatta. Una voce DNS inesatta può esporre il traffico Internet destinato al tuo dominio a vulnerabilità di sicurezza. Può presentare anche altri rischi.
> Per ulteriori informazioni, consulta l'argomento relativo alla Protezione dai record di delega inesatti in Route 53 nella Guida per sviluppatori di Amazon Route 53. Puoi anche saperne di più sulle voci DNS sospese in Enhanced Domain Protections for Amazon CloudFront Requests nel Security Blog.AWS

Console Elastic Beanstalk

Come terminare un ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

   > **Note**
   >
   > Quando termini l'ambiente, il CNAME associato all'ambiente terminato diventa disponibile e può essere utilizzato da un altro utente.

   Elastic Beanstalk impiega alcuni minuti per AWS terminare le risorse in esecuzione nell'ambiente.

AWS CLI

Come terminare un ambiente

- Eseguire il seguente comando seguente.

```
$ aws elasticbeanstalk terminate-environment --environment-name my-env
```

"Hello, World!"

Come terminare un ambiente

- Chiama `TerminateEnvironment` con il seguente parametro:

  `EnvironmentName` = `SampleAppEnv`

```
https://elasticbeanstalk.us-west-2.amazon.com/?EnvironmentName=SampleAppEnv
&Operation=TerminateEnvironment
&AuthParams
```

Creazione di ambienti Elastic Beanstalk con la CLI AWS

Per i dettagli sui AWS CLI comandi per Elastic Beanstalk, consulta il Command Reference.AWS CLI

1. Controlla se è disponibile il CNAME per l'ambiente.

```
$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

2. Verifica che la versione dell'applicazione esista.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label v1
```

Se non si dispone ancora di una versione dell'applicazione per la propria origine, occorre crearla. Ad esempio, il comando seguente crea una versione dell'applicazione da un bundle di origine in Amazon Simple Storage Service (Amazon S3).

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --source-bundle S3Bucket=amzn-s3-demo-bucket,S3Key=my-source-
bundle.zip
```

3. Crea un modello di configurazione per l'applicazione.

```
$ aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name v1 --solution-stack-name "64bit Amazon Linux 2015.03 v2.0.0 running
 Ruby 2.2 (Passenger Standalone)"
```

4. Crea l'ambiente.

```
$ aws elasticbeanstalk create-environment --cname-prefix my-cname --application-
name my-app --template-name v1 --version-label v1 --environment-name v1clone --
option-settings file://options.txt
```

Le impostazioni delle opzioni sono definite nel file options.txt:

```
[
    {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": "aws-elasticbeanstalk-ec2-role"
    }
]
```

L'impostazione dell'opzione sopra indicata definisce il profilo dell'istanza di IAM. Puoi specificare l'ARN o il nome del profilo.

5. Determina se il nuovo ambiente è Green e Ready.

```
$ aws elasticbeanstalk describe-environments --environment-names my-env
```

Se il nuovo ambiente non è Green e Ready, dovrai decidere se ritentare l'operazione o lasciare l'ambiente nel suo stato corrente per un'indagine. Dopo aver completato, termina l'ambiente ed esegui il cleanup delle risorse inutilizzate.

> **Note**
>
> Puoi regolare il periodo di timeout nel caso in cui l'ambiente non si avvii in un periodo di tempo ragionevole.

Creazione di ambienti Elastic Beanstalk con l'API

1. Chiama `CheckDNSAvailability` con il seguente parametro:

   - `CNAMEPrefix` = `SampleApp`

   Example

   ```
   https://elasticbeanstalk.us-east-2.amazonaws.com/?CNAMEPrefix=sampleapplication
   &Operation=CheckDNSAvailability
   &AuthParams
   ```

2. Chiama `DescribeApplicationVersions` con i seguenti parametri:

   - `ApplicationName` = `SampleApp`
   - `VersionLabel` = `Version2`

   Example

   ```
   https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
   &VersionLabel=Version2
   &Operation=DescribeApplicationVersions
   &AuthParams
   ```

3. Chiama `CreateConfigurationTemplate` con i seguenti parametri:

   - `ApplicationName` = `SampleApp`
   - `TemplateName` = `MyConfigTemplate`
   - `SolutionStackName` = `64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby%202.2%20(Passenger%20Standalone)`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&TemplateName=MyConfigTemplate
&Operation=CreateConfigurationTemplate
&SolutionStackName=64bit%20Amazon%20Linux%202015.03%20v2.0.0%20running%20Ruby
%202.2%20(Passenger%20Standalone)
&AuthParams
```

4. Chiama `CreateEnvironment` con uno dei seguenti set di parametri.

   a. Usa i seguenti parametri per un livello ambiente di server Web:

      - `EnvironmentName` = `SampleAppEnv2`
      - `VersionLabel` = `Version2`
      - `Description` = `description`
      - `TemplateName` = `MyConfigTemplate`
      - `ApplicationName` = `SampleApp`
      - `CNAMEPrefix` = `sampleapplication`
      - `OptionSettings.member.1.Namespace` = `aws:autoscaling:launchconfiguration`
      - `OptionSettings.member.1.OptionName` = `IamInstanceProfile`
      - `OptionSettings.member.1.Value` = `aws-elasticbeanstalk-ec2-role`

      Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&CNAMEPrefix=sampleapplication
&Description=description
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
```

```
&AuthParams
```

b. Usa i seguenti parametri per un livello ambiente lavoratore:

- `EnvironmentName` = `SampleAppEnv2`
- `VersionLabel` = `Version2`
- `Description` = `description`
- `TemplateName` = `MyConfigTemplate`
- `ApplicationName` = `SampleApp`
- `Tier` = `Worker`
- `OptionSettings.member.1.Namespace` = `aws:autoscaling:launchconfiguration`
- `OptionSettings.member.1.OptionName` = `IamInstanceProfile`
- `OptionSettings.member.1.Value` = `aws-elasticbeanstalk-ec2-role`
- `OptionSettings.member.2.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.2.OptionName` = `WorkerQueueURL`
- `OptionSettings.member.2.Value` = `sqsd.elasticbeanstalk.us-east-2.amazonaws.com`
- `OptionSettings.member.3.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.3.OptionName` = `HttpPath`
- `OptionSettings.member.3.Value` = `/`
- `OptionSettings.member.4.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.4.OptionName` = `MimeType`
- `OptionSettings.member.4.Value` = `application/json`
- `OptionSettings.member.5.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.5.OptionName` = `HttpConnections`
- `OptionSettings.member.5.Value` = `75`
- `OptionSettings.member.6.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.6.OptionName` = `ConnectTimeout`
- `OptionSettings.member.6.Value` = `10`
- `OptionSettings.member.7.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.7.OptionName` = `InactivityTimeout`

- `OptionSettings.member.7.Value` = `10`
- `OptionSettings.member.8.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.8.OptionName` = `VisibilityTimeout`
- `OptionSettings.member.8.Value` = `60`
- `OptionSettings.member.9.Namespace` = `aws:elasticbeanstalk:sqsd`
- `OptionSettings.member.9.OptionName` = `RetentionPeriod`
- `OptionSettings.member.9.Value` = `345600`

Example

```
https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
&VersionLabel=Version2
&EnvironmentName=SampleAppEnv2
&TemplateName=MyConfigTemplate
&Description=description
&Tier=Worker
&Operation=CreateEnvironment
&OptionSettings.member.1.Namespace=aws%3Aautoscaling%3Alaunchconfiguration
&OptionSettings.member.1.OptionName=IamInstanceProfile
&OptionSettings.member.1.Value=aws-elasticbeanstalk-ec2-role
&OptionSettings.member.2.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.2.OptionName=WorkerQueueURL
&OptionSettings.member.2.Value=sqsd.elasticbeanstalk.us-east-2.amazonaws.com
&OptionSettings.member.3.Namespace=aws%3elasticbeanstalk%3sqsd
&OptionSettings.member.3.OptionName=HttpPath
&OptionSettings.member.3.Value=%2F
&OptionSettings.member.4.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.4.OptionName=MimeType
&OptionSettings.member.4.Value=application%2Fjson
&OptionSettings.member.5.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.5.OptionName=HttpConnections
&OptionSettings.member.5.Value=75
&OptionSettings.member.6.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.6.OptionName=ConnectTimeout
&OptionSettings.member.6.Value=10
&OptionSettings.member.7.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.7.OptionName=InactivityTimeout
&OptionSettings.member.7.Value=10
&OptionSettings.member.8.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.8.OptionName=VisibilityTimeout
```

```
&OptionSettings.member.8.Value=60
&OptionSettings.member.9.Namespace=aws%3Aelasticbeanstalk%3Asqsd
&OptionSettings.member.9.OptionName=RetentionPeriod
&OptionSettings.member.9.Value=345600
&AuthParams
```

Creazione di un URL Launch Now

È possibile creare un URL personalizzato per consentire a chiunque di distribuire ed eseguire rapidamente una predeterminata applicazione Web in AWS Elastic Beanstalk. Questo URL è denominato Launch Now. Un URL Launch Now è utile, ad esempio, per illustrare un'applicazione Web sviluppata per l'esecuzione in Elastic Beanstalk. Con un URL Launch Now, puoi utilizzare dei parametri per aggiungere prima le informazioni necessarie alla procedura guidata Create Application (Crea applicazione). Dopo aver aggiunto questa informazione alla procedura guidata, chiunque può utilizzare il collegamento dell'URL per avviare un ambiente Elastic Beanstalk con il codice sorgente della tua applicazione Web in pochi passaggi. Ciò significa che gli utenti non devono caricare o specificare manualmente la posizione del bundle di origine dell'applicazione. Non è necessario fornire ulteriori informazioni alla procedura guidata.

Un URL Launch Now fornisce a Elastic Beanstalk le informazioni minime necessarie a creare un'applicazione: il nome dell'applicazione, lo stack di soluzioni, il tipo di istanza e il tipo di ambiente. Elastic Beanstalk utilizza i valori di default per altri dettagli di configurazione non indicati in modo esplicito nell'URL Launch Now personalizzato.

Un URL Launch Now utilizza la sintassi URL standard. Per ulteriori informazioni, consulta RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax.

Parametri dell'URL

L'URL deve contenere i seguenti parametri, per i quali viene fatta la distinzione tra maiuscole e minuscole:

- regione: specifica una AWS regione. Per un elenco delle Regioni supportate da Elastic Beanstalk, consulta Endpoint e quote di AWS Elastic Beanstalk nella Riferimenti generali di AWS.
- applicationName - Specifica il nome dell'applicazione. Elastic Beanstalk visualizza il nome dell'applicazione nella console Elastic Beanstalk per distinguerlo dalle altre applicazioni. Per impostazione predefinita, il nome dell'applicazione costituisce anche la base del nome dell'ambiente e dell'URL dell'ambiente.

- platform - Specifica la versione della piattaforma da utilizzare per l'ambiente. Utilizza uno dei seguenti metodi, quindi codifica in formato URL la tua scelta:
  - Specifica un ARN di piattaforma senza la versione. Elastic Beanstalk seleziona l'ultima versione della piattaforma della versione principale della piattaforma corrispondente. Ad esempio, per selezionare la versione della piattaforma Python 3.6 più recente, specifica `Python 3.6 running on 64bit Amazon Linux`.
  - Specifica il nome della piattaforma. Elastic Beanstalk seleziona l'ultima versione dell'ultimo runtime del linguaggio della piattaforma (ad esempio, `Python`).

  Per una descrizione di tutte le piattaforme disponibili e delle relative versioni, consulta Piattaforme supportate da Elastic Beanstalk.

  Puoi usare AWS Command Line Interface(AWS CLI) per ottenere un elenco di tutte le versioni della piattaforma disponibili con le rispettive ARNs. Il comando `list-platform-versions` elenca informazioni dettagliate su tutte le versioni della piattaforma disponibili. Utilizza l'argomento `--filters` per ridurre l'ambito dell'elenco. Ad esempio, puoi esaminare l'elenco solo per mostrare le versioni della piattaforma di un linguaggio specifico.

  Nell'esempio seguente viene effettuata una query per tutte le versioni della piattaforma Python e l'output viene reindirizzato attraverso una serie di comandi. Il risultato è un elenco delle versioni della piattaforma ARNs (senza la `/version` coda), in un formato leggibile dall'uomo, senza codifica URL.

```
$ aws elasticbeanstalk list-platform-versions --filters
 'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk -
F '"' '{print $4}' | awk -F '/' '{print $2}'
Preconfigured Docker - Python 3.4 running on 64bit Debian
Preconfigured Docker - Python 3.4 running on 64bit Debian
Python 2.6 running on 32bit Amazon Linux
Python 2.6 running on 32bit Amazon Linux 2014.03
...
Python 3.6 running on 64bit Amazon Linux
```

  L'esempio seguente aggiunge un comando Perl per l'ultimo esempio per la codifica in formato URL dell'output.

```
$ aws elasticbeanstalk list-platform-versions --filters
 'Type="PlatformName",Operator="contains",Values="Python"' | grep PlatformArn | awk
```

```
 -F '"' '{print $4}' | awk -F '/' '{print $2}' | perl -MURI::Escape -ne 'chomp;print
 uri_escape($_),"\n"'
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Preconfigured%20Docker%20-%20Python%203.4%20running%20on%2064bit%20Debian
Python%202.6%20running%20on%2032bit%20Amazon%20Linux
Python%202.6%20running%20on%2032bit%20Amazon%20Linux%202014.03
...
Python%203.6%20running%20on%2064bit%20Amazon%20Linux
```

Un URL Launch Now può eventualmente contenere i parametri elencati di seguito. Se non includi i parametri facoltativi nell'URL Launch Now, Elastic Beanstalk utilizza i valori di default per creare ed eseguire la tua applicazione. Quando non includi il sourceBundleUrlparametro, Elastic Beanstalk utilizza l'applicazione di esempio predefinita per la piattaforma specificata.

- sourceBundleUrl— Specificate la posizione del pacchetto sorgente dell'applicazione Web in formato URL. Ad esempio, se hai caricato il tuo pacchetto sorgente in un bucket Amazon S3, puoi specificare il valore del sourceBundleUrlparametro come. `https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject`

  > **Note**
  >
  > Puoi specificare il valore del sourceBundleUrlparametro come URL HTTP, ma il browser Web dell'utente convertirà i caratteri secondo necessità applicando la codifica URL HTML.

- environmentType - Indica se l'ambiente è con carico bilanciato e dimensionabile oppure è una singola istanza. Per ulteriori informazioni, consulta Tipi di ambiente. Puoi specificare `LoadBalancing` o `SingleInstance`.
- tierName - Indica se l'ambiente supporta un'applicazione Web che elabora richieste Web o un'applicazione Web che esegue processi in background. Per ulteriori informazioni, consulta Ambienti worker Elastic Beanstalk. Puoi specificare `WebServer` o `Worker`.
- instanceType - Specifica un server con le caratteristiche (inclusa la quantità di memoria e la potenza della CPU) più adatte per la tua applicazione. Per ulteriori informazioni sulle famiglie e i tipi di EC2 istanze Amazon, consulta Tipi di istanze nella Amazon EC2 User Guide. Per ulteriori informazioni sui tipi di istanze disponibili nelle diverse regioni, consulta la sezione Tipi di istanze disponibili nella Amazon EC2 User Guide.

- withVpc - Indica se creare l'ambiente in un Amazon VPC. È possibile specificare `true` o `false`. Per ulteriori informazioni sull'utilizzo di Elastic Beanstalk con Amazon VPC, consulta Utilizzo di Elastic Beanstalk con Amazon VPC.
- withRds - Specifica se creare un'istanza database Amazon RDS con l'ambiente. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon RDS. È possibile specificare `true` o `false`.
- rds DBEngine: specifica il motore di database che desideri utilizzare per le tue EC2 istanze Amazon in questo ambiente. Puoi indicare `mysql`, `oracle-sel`, `sqlserver-ex`, `sqlserver-web` o `sqlserver-se`. Il valore predefinito è `mysql`.
- rds DBAllocated Storage: specifica la dimensione dello storage del database allocato in gigabyte (GB). Puoi specificare le seguenti valori:
  - MySQL: da `5` a `1024`. Il valore di default è `5`.
  - Oracle: da `10` a `1024`. Il valore di default è `10`.
  - Microsoft SQL Server Express Edition – `30`.
  - Microsoft SQL Server Web Edition – `30`.
  - Microsoft SQL Server Standard Edition – `200`.
- rds DBInstance Class: specifica il tipo di istanza del database. Il valore predefinito è `db.t2.micro` (`db.m1.large` è per un ambiente non in esecuzione in un Amazon VPC). Per l'elenco delle classi di istanza database supportate da Amazon RDS, consultare Classe di istanza database nella Guida per l'utente di Amazon Relational Database Service.
- RDSMulti AZDatabase — Specificare se Elastic Beanstalk deve creare l'istanza del database su più zone di disponibilità. È possibile specificare `true` o `false`. Per ulteriori informazioni sulle distribuzioni di più zone di disponibilità con Amazon RDS, consultare Regioni e zone di disponibilità nella Guida per l'utente di Amazon Relational Database Service.
- rds DBDeletion Policy: specifica se eliminare o creare un'istantanea dell'istanza del database alla chiusura dell'ambiente. È possibile specificare `Delete` o `Snapshot`.

Esempio

Di seguito è riportato un esempio di URL Launch Now. Una volta creato l'URL, puoi passarlo ai tuoi utenti. Ad esempio, lo puoi incorporare in una pagina Web o in materiale didattico. Quando gli utenti creano un'applicazione utilizzando l'URL Launch Now, la procedura guidata Create an Application (Crea un'applicazione) di Elastic Beanstalk non richiede altri dati.

```
https://console.aws.amazon.com/elasticbeanstalk/home?region=us-west-2#/newApplication?
applicationName=YourCompanySampleApp
      &platform=PHP%207.3%20running%20on%2064bit%20Amazon%20Linux&sourceBundleUrl=
        http://s3.amazonaws.com/amzn-s3-demo-bucket/
myobject&environmentType=SingleInstance&tierName=WebServer
      &instanceType=m1.small&withVpc=true&withRds=true&rdsDBEngine=

 postgres&rdsDBAllocatedStorage=6&rdsDBInstanceClass=db.m1.small&rdsMultiAZDatabase=
          true&rdsDBDeletionPolicy=Snapshot
```

Utilizzo dell'URL Launch Now

1. Scegli l'URL Launch Now.
2. Dopo aver aperto la console Elastic Beanstalk, scegli Review and Launch (Analizza e avvia) nella pagina Create a web app (Crea un'app per il Web) per visualizzare le impostazioni utilizzate da Elastic Beanstalk per creare l'applicazione e avviare l'ambiente in cui l'applicazione viene eseguita.
3. Nella pagina Configure (Configura), scegli Create app (Crea app) per creare l'applicazione.

Creazione e aggiornamento di gruppi di ambienti Elastic Beanstalk

Con l' AWS Elastic Beanstalk ComposeEnvironmentsAPI, puoi creare e aggiornare gruppi di ambienti Elastic Beanstalk all'interno di una singola applicazione. Ogni ambiente del gruppo può eseguire un componente separato di un'applicazione con architettura orientata ai servizi. L'API `Compose Environments` utilizza un elenco di versioni dell'applicazione e un nome di gruppo opzionale. Elastic Beanstalk crea un ambiente per ciascuna versione dell'applicazione oppure, se gli ambienti esistono già, vi distribuisce le versioni delle applicazioni.

Crea i collegamenti tra gli ambienti Elastic Beanstalk per designare un ambiente come dipendenza di un altro. Quando si crea un gruppo di ambienti con l'API `Compose Environments`, Elastic Beanstalk crea ambienti dipendenti solo dopo che le rispettive dipendenze sono attive e in esecuzione. Per ulteriori informazioni sui collegamenti degli ambienti, consulta Creazione di collegamenti tra gli ambienti Elastic Beanstalk.

L'API `Compose Environments` utilizza un manifest di ambiente per archiviare i dettagli di configurazione che sono condivisi da gruppi di ambienti. Per ogni applicazione componente deve essere presente un file di configurazione `env.yaml` nel bundle di origine dell'applicazione che specifichi i parametri utilizzati per creare il rispettivo ambiente.

`Compose Environments` richiede che vengano specificati `EnvironmentName` e `SolutionStack` nel manifest dell'ambiente per ogni applicazione componente.

Puoi utilizzare l'`Compose EnvironmentsAPI` con l'interfaccia a riga di comando Elastic Beanstalk (EB CLI), o un SDK. AWS CLI Per istruzioni su CLI EB, vedi Gestione di più ambienti elastici Elastic Beanstalk come gruppo con la CLI EB.

Uso dell'API **`Compose Environments`**

Ad esempio, puoi creare un'applicazione denominata `Media Library` che consenta agli utenti di caricare e gestire immagini e video archiviati in Amazon Simple Storage Service (Amazon S3). L'applicazione ha un ambiente di front-end, `front`, che esegue un'applicazione Web che consente agli utenti di caricare e scaricare singoli file, visualizzare la libreria e avviare processi di elaborazione in batch.

Invece di elaborare i processi direttamente, l'applicazione front-end aggiunge un processo a una coda Amazon SQS. Il secondo ambiente, `worker`, preleva le attività dalla coda e le elabora. `worker` utilizza un tipo di istanza G2 che dispone di una GPU ad alte prestazioni, mentre `front` può essere eseguito su un tipo di istanza generico più conveniente.

È possibile organizzare la cartella del progetto, `Media Library`, in directory separate per ogni componente, con ciascuna directory contenente un file di definizione dell'ambiente (`env.yaml`), ognuno con il proprio codice sorgente:

```
~/workspace/media-library
|-- front
|   `-- env.yaml
`-- worker
    `-- env.yaml
```

I seguenti elenchi mostrano il file `env.yaml` per ogni applicazione componente.

`~/workspace/media-library/front/env.yaml`

```
EnvironmentName: front+
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
  Name: WebServer
  Type: Standard
```

```
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: m4.large
```

~/workspace/media-library/worker/env.yaml

```
EnvironmentName: worker+
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.4 running Java 8
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: g2.2xlarge
```

Dopo aver creato una versione dell'applicazione per i componenti applicazione front-end (`front-v1`) e lavoratore (`worker-v1`), chiama l'API `Compose Environments` con i nomi di versione. In questo esempio abbiamo utilizzato la AWS CLI per chiamare l'API.

```
# Create application versions for each component:
~$ aws elasticbeanstalk create-application-version --application-name media-
library --version-label front-v1 --process --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="front-v1.zip"
  {
    "ApplicationVersion": {
        "ApplicationName": "media-library",
        "VersionLabel": "front-v1",
        "Description": "",
        "DateCreated": "2015-11-03T23:01:25.412Z",
        "DateUpdated": "2015-11-03T23:01:25.412Z",
        "SourceBundle": {
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Key": "front-v1.zip"
        }
    }
  }
~$ aws elasticbeanstalk create-application-version --application-name media-library
 --version-label worker-v1 --process --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="worker-v1.zip"
  {
```

```
    "ApplicationVersion": {
        "ApplicationName": "media-library",
        "VersionLabel": "worker-v1",
        "Description": "",
        "DateCreated": "2015-11-03T23:01:48.151Z",
        "DateUpdated": "2015-11-03T23:01:48.151Z",
        "SourceBundle": {
            "S3Bucket": "amzn-s3-demo-bucket",
            "S3Key": "worker-v1.zip"
        }
    }
  }
# Create environments:
~$ aws elasticbeanstalk compose-environments --application-name media-library --group-
name dev --version-labels front-v1 worker-v1
```

La terza chiamata crea due ambienti, `front-dev` e `worker-dev`. L'API crea i nomi degli ambienti concatenando il nome `EnvironmentName` specificato nel file `env.yaml` con l'opzione `group name` specificata nella chiamata `Compose Environments`, separati da un trattino. La lunghezza totale di queste due opzioni e del trattino non deve superare la lunghezza massima consentita per i nomi di ambiente, ovvero 23 caratteri.

L'applicazione in esecuzione nell'ambiente `front-dev` può accedere al nome della coda Amazon SQS collegato all'ambiente `worker-dev` leggendo la variabile `WORKERQUEUE`. Per ulteriori informazioni sui collegamenti degli ambienti, consulta Creazione di collegamenti tra gli ambienti Elastic Beanstalk.

Gestione di più ambienti elastici Elastic Beanstalk come gruppo con la CLI EB

È possibile utilizzare l'EB CLI per creare gruppi di AWS Elastic Beanstalk ambienti, ciascuno dei quali esegue un componente separato di un'applicazione di architettura orientata ai servizi. L'EB CLI gestisce tali gruppi utilizzando l'ComposeEnvironmentsAPI.

> **Note**
>
> I gruppi di ambienti sono diversi dai vari container in un ambiente docker multicontainer. Con i gruppi di ambiente, ogni componente della tua applicazione viene eseguito in un ambiente Elastic Beanstalk distinto, con una serie di istanze Amazon EC2 dedicate. Ogni componente è in grado di ridimensionarsi separatamente. Con docker multicontainer, puoi combinare più

componenti di un'applicazione in un unico ambiente. Tutti i componenti condividono lo stesso set di istanze Amazon EC2, ognuna delle quali esegue più container docker. Scegli una delle architetture in base alle esigenze della tua attività.
Per informazioni dettagliate sui docker multicontainer, consultare Utilizzo del ramo della piattaforma Docker gestito da ECS in Elastic Beanstalk.

Organizza i componenti dell'applicazione nella seguente struttura di cartelle:

```
~/project-name
|-- component-a
|   `-- env.yaml
`-- component-b
    `-- env.yaml
```

Ogni sottocartella contiene il codice sorgente per un componente indipendente di un'applicazione che verrà eseguito nel relativo ambiente e un file di definizione di ambiente denominato `env.yaml`. Per informazioni dettagliate sul formato `env.yaml`, consulta Manifest di ambiente (`env.yaml`).

Per utilizzare l'API `Compose Environments`, dapprima esegui eb init dalla cartella di progetto, specificando ogni componente in base al nome della cartella che lo contiene con l'opzione `--modules`:

```
~/workspace/project-name$ eb init --modules component-a component-b
```

L'interfaccia a riga di comando EB richiede la configurazione di ogni componente e quindi crea la directory `.elasticbeanstalk` nella cartella di ciascun componente. L'interfaccia a riga di comando EB non crea i file di configurazione nella directory principale.

```
~/project-name
|-- component-a
|   |-- .elasticbeanstalk
|   `-- env.yaml
`-- component-b
    |-- .elasticbeanstalk
    `-- env.yaml
```

Successivamente, esegui il comando eb create con un elenco di ambienti da creare (uno per ogni componente):

```
~/workspace/project-name$ eb create --modules component-a component-b --env-group-
suffix group-name
```

Questo comando crea un ambiente per ogni componente. I nomi degli ambienti vengono creati concatenando il valore `EnvironmentName` specificato nel file `env.yaml` con il nome del gruppo, separato da un trattino. La lunghezza totale di queste due opzioni e del trattino non deve superare la lunghezza massima consentita per i nomi di ambiente, ovvero 23 caratteri.

Per aggiornare l'ambiente, utilizza il comando eb deploy:

```
~/workspace/project-name$ eb deploy --modules component-a component-b
```

Puoi aggiornare ogni componente individualmente oppure in gruppo. Specifica i componenti che intendi aggiornare con l'opzione `--modules`.

L'interfaccia a riga di comando EB memorizza il nome di gruppo utilizzato con eb create nella sezione `branch-defaults` del file di configurazione dell'interfaccia in `/.elasticbeanstalk/config.yml`. Per distribuire l'applicazione in un gruppo differente, utilizza l'opzione `--env-group-suffix` quando esegui eb deploy. Se il gruppo non esiste ancora, l'interfaccia a riga di comando EB creerà un nuovo gruppo di ambienti:

```
~/workspace/project-name$ eb deploy --modules component-a component-b --env-group-
suffix group-2-name
```

Per terminare gli ambienti, esegui eb terminate nella cartella di ogni modulo. Per impostazione predefinita, l'interfaccia a riga di comando EB visualizzerà un errore se tenti di terminare un ambiente da cui dipende un altro ambiente in esecuzione. Termina dapprima l'ambiente dipendente, oppure utilizza l'opzione `--ignore-links` per sovrascrivere il comportamento di default:

```
~/workspace/project-name/component-b$ eb terminate --ignore-links
```

Distribuzione di applicazioni in ambienti Elastic Beanstalk

Puoi utilizzare la AWS Elastic Beanstalk console per caricare un pacchetto sorgente aggiornato e distribuirlo nel tuo ambiente Elastic Beanstalk oppure ridistribuire una versione caricata in precedenza.

Ogni distribuzione è identificata da un ID della distribuzione. Gli ID distribuzione iniziano da 1 e vengono incrementati di uno a ogni distribuzione o modifica della configurazione dell'istanza. Attivando il reporting sullo stato migliorato, Elastic Beanstalk mostra l'ID distribuzione sia nella console dello stato, sia nella CLI EB quando comunica lo stato dell'istanza. L'ID della distribuzione consente di stabilire lo stato dell'ambiente quando un aggiornamento in sequenza ha esito negativo.

Elastic Beanstalk fornisce diverse policy e impostazioni di distribuzione. Per informazioni dettagliate sulla configurazione di una policy e di impostazioni aggiuntive, consulta the section called "Opzioni di implementazione". Nella tabella seguente sono elencate le policy e i tipi di ambienti che le supportano.

Policy di distribuzione supportate

Policy di distribuzione	Load-balanced ambienti	Single-instance ambienti	Ambienti di Windows Server legacy†
Tutto contemporaneamente	✓ Sì	✓ Sì	✓ Sì
In sequenza	✓ Sì	× No	✓ Sì
In sequenza con batch aggiuntivi	✓ Sì	× No	× No
Non modificabile	✓ Sì	✓ Sì	× No
Suddivisione del traffico	✓ Sì (Application Load Balancer)	× No	× No

† In questa tabella, un ambiente Windows Server legacy è un ambiente basato su una configurazione della piattaforma Windows Server che utilizza una versione IIS precedente a IIS 8.5.

> **Warning**
>
> Alcuni criteri sostituiscono tutte le istanze durante la distribuzione o l'aggiornamento. Ciò causa la perdita di tutti i saldi burst di Amazon EC2 accumulati. Succede nei seguenti casi:
>
> - Aggiornamenti della piattaforma gestiti con sostituzione delle istanze abilitata
> - Aggiornamenti non modificabili

- Distribuzioni con aggiornamenti immutabili o suddivisione del traffico abilitata

Scelta di una policy di distribuzione

La scelta della policy di distribuzione appropriata per l'applicazione è un compromesso di alcune considerazioni e dipende dalle esigenze specifiche. La pagina the section called “Opzioni di implementazione” contiene ulteriori informazioni su ogni policy e una descrizione dettagliata del funzionamento di alcune di esse.

Nell'elenco seguente vengono fornite informazioni di riepilogo sulle diverse policy di distribuzione con considerazioni correlate.

- Tutto contemporaneamente: il metodo di distribuzione più rapido. Adatto se puoi accettare una breve perdita del servizio e se le distribuzioni rapide sono per te importanti. Con questo metodo Elastic Beanstalk distribuisce la nuova versione dell'applicazione in ogni istanza. Quindi, potrebbe essere necessario riavviare il proxy Web o il server applicazioni. Di conseguenza, l'applicazione potrebbe non essere disponibile per gli utenti (o avere bassa disponibilità) per un breve periodo.
- In sequenza: evita i tempi di inattività e contiene al minimo la disponibilità ridotta, ma con tempi di distribuzione più lunghi. Adatto se non puoi accettare un periodo con perdita del servizio totale. Con questo metodo, l'applicazione viene distribuita nell'ambiente in uso per un batch di istanze alla volta. La maggior parte della larghezza di banda viene mantenuta per tutta la distribuzione.
- In sequenza con batch aggiuntivi: evita la disponibilità ridotta, ma con tempi di distribuzione ancora più lunghi rispetto al metodo In sequenza. Adatto se è necessario mantenere la stessa larghezza di banda per tutta la distribuzione. Con questo metodo Elastic Beanstalk avvia un batch aggiuntivo di istanze, quindi esegue una distribuzione in sequenza. L'avvio del batch aggiuntivo richiede tempo e garantisce che la stessa larghezza di banda venga mantenuta per tutta la distribuzione.
- Non modificabile: metodo di distribuzione più lento che assicura che la nuova versione dell'applicazione venga sempre distribuita nelle nuove istanze, anziché aggiornare le istanze esistenti. Ha anche il vantaggio aggiuntivo di un rollback rapido e sicuro nel caso in cui la distribuzione non riesca. Con questo metodo, Elastic Beanstalk esegue un aggiornamento non modificabile per distribuire l'applicazione. In un aggiornamento non modificabile viene avviato un secondo gruppo Auto Scaling nel tuo ambiente e la nuova versione serve il traffico insieme alla versione precedente fino a quando le nuove istanze superano i controlli dello stato.

- Suddivisione del traffico: metodo di distribuzione con test canary. Adatto se vuoi verificare l'integrità della nuova versione dell'applicazione utilizzando una parte del traffico in entrata, mantenendo il resto del traffico servito dalla versione precedente dell'applicazione.

Nella tabella seguente vengono confrontate le proprietà dei metodi di distribuzione.

Metodi di distribuzione

Metodo	Impatto di una distribuzione non riuscita	Orario di distribuzione	Nessun tempo di inattività	Nessun modifica DNS	Process di rollback	Codice distribui to in
Tutto contem aneame	Tempo di inattività	🕒	✕ No	✓ Sì	Ridistrib uzione manual	Istanze esistenti
In sequen	Batch singolo fuori servizio: un qualsiasi batch completato con successo prima del fallimento dell'esecuzione di una nuova versione dell'applicazione	🕒 🕒	✓ Sì	✓ Sì	Ridistrib uzione manual	Istanze esistenti
In sequen con batch aggiunti i	Minima se il primo batch ha esito negativo; altrimenti simile a Rolling (In sequenza)	🕒 🕒	✓ Sì	✓ Sì	Ridistrib uzione manual	Istanze nuove ed esistenti
Non modifica ile	Minima	🕒 🕒	✓ Sì	✓ Sì	Termina nuove istanze	Nuove istanze
Suddivi one del traffico	Percentuale di traffico client instradato alla nuova versione interessata temporaneamente	🕒 🕒	✓ Sì	✓ Sì	Reindiri za il traffico e	Nuove istanze

Metodo	Impatto di una distribuzione non riuscita	Orario di distribuzione	Nessun tempo di inattività	Nessun modifica DNS	Process di rollback	Codice distribui to in
					termina le nuove istanze	
Blue/ green	Minima	🕒 🕒	✓ Sì	× No	Scambi URL	Nuove istanze

† Varia in base alle dimensioni del batch.

†† Varia in base all'impostazione dell'opzione del tempo di valutazione.

Distribuzione di una nuova versione dell'applicazione

È possibile eseguire le distribuzioni dal pannello di controllo dell'ambiente.

Per distribuire una nuova versione dell'applicazione in un ambiente Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Selezionare Upload and Deploy (Carica e distribuisci).
4. Utilizza il modulo su schermo per caricare il bundle di origine dell'applicazione.
5. Seleziona Implementa.

Ridistribuzione di una versione precedente

Puoi anche distribuire una versione dell'applicazione caricata in precedenza su uno qualsiasi dei suoi ambienti dalla pagina delle versioni dell'applicazione.

Per distribuire una versione dell'applicazione esistente in un ambiente esistente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Application versions (Versioni applicazione).
4. Selezionare la versione dell'applicazione da distribuire.
5. Scegliere Actions (Operazioni), quindi Deploy (Distribuisci).
6. Selezionare un ambiente, quindi scegliere Deploy (Distribuisci).

Altri modi per distribuire l'applicazione

Se si distribuisce spesso, è consigliabile utilizzare l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) per gestire gli ambienti. La CLI EB crea un repository insieme al codice sorgente. Può anche creare un bundle di origine, caricarlo in Elastic Beanstalk e distribuirlo con un singolo comando.

Per le distribuzioni che dipendono dalle modifiche alla configurazione delle risorse o da una nuova versione che non può essere eseguita insieme alla versione precedente, puoi avviare un nuovo ambiente con la nuova versione ed eseguire uno scambio di CNAME per una distribuzione. blue/green

Per automatizzare i processi di compilazione, test e distribuzione, puoi implementare l'integrazione continua e la distribuzione continua (CI/CD) con il tuo ambiente Elastic Beanstalk. Per ulteriori informazioni, consulta CI/CD Implementazione dell'integrazione con l'ambiente Elastic Beanstalk.

Impostazioni e policy di distribuzione

AWS Elastic Beanstalk offre diverse opzioni per l'elaborazione delle distribuzioni, tra cui politiche di distribuzione (All at once, Rolling, Rolling with batch aggiuntivo, Immutable e Traffic splitting) e opzioni che consentono di configurare le dimensioni dei batch e il comportamento dei controlli di integrità durante le distribuzioni. Per impostazione predefinita, l'ambiente utilizza le distribuzioni. all-at-once Se hai creato l'ambiente con la CLI EB e si tratta di un ambiente di scalabile (non hai specificato l'opzione `--single`), vengono usate le implementazioni in sequenza.

Con distribuzioni continue, Elastic Beanstalk divide le istanze EC2 Amazon dell'ambiente in batch e distribuisce la nuova versione dell'applicazione in un batch alla volta. Lascia il resto delle istanze nell'ambiente che esegue la vecchia versione dell'applicazione. Durante una distribuzione in

sequenza, alcune istanze servono le richieste con la versione precedente dell'applicazione, mentre le istanze in batch completati servono altre richieste con la nuova versione. Per informazioni dettagliate, vedi the section called “Funzionamento delle distribuzioni in sequenza”.

Per mantenere la capacità completa durante le distribuzioni, puoi configurare l'ambiente per avviare un nuovo batch di istanze prima di considerare qualsiasi istanza fuori servizio. Questa opzione è nota come distribuzione in sequenza con batch aggiuntivi. Quando la distribuzione viene completato, Elastic Beanstalk termina il batch aggiuntivo di istanze.

Le distribuzioni non modificabili eseguono un aggiornamento non modificabile per avviare un set completo di nuove istanze in esecuzione con la nuova versione dell'applicazione in un gruppo Auto Scaling separato contemporaneamente alle istanze che eseguono la versione precedente. Le distribuzioni non modificabili possono evitare i problemi causati da distribuzioni in sequenza parzialmente completate. Se le nuove istanze non superano i controlli dello stato, Elastic Beanstalk le termina, lasciando le istanze originali intatte.

Ledistribuzioni con suddivisione del traffico consentono di eseguire i test canary come parte della distribuzione dell'applicazione. In una distribuzione di suddivisione del traffico, Elastic Beanstalk avvia un set completo di nuove istanze proprio come durante una distribuzione immutabile. Inoltra quindi una percentuale specificata di traffico client in ingresso alla nuova versione dell'applicazione per un periodo di valutazione specificato. Se le nuove istanze rimangono integre, Elastic Beanstalk inoltra loro tutto il traffico e termina le istanze precedenti. Se le nuove istanze non superano i controlli dello stato o se si sceglie di interrompere la distribuzione, Elastic Beanstalk riporta il traffico alle vecchie istanze e termina quelle nuove. Non ci sono mai interruzioni di servizio. Per informazioni dettagliate, vedi the section called “Come funzionano le distribuzioni con suddivisione del traffico”.

> **Warning**
>
> Alcuni criteri sostituiscono tutte le istanze durante la distribuzione o l'aggiornamento. Ciò causa la perdita di tutti i saldi Amazon EC2 Burst accumulati. Succede nei seguenti casi:
>
> - Aggiornamenti della piattaforma gestiti con sostituzione delle istanze abilitata
> - Aggiornamenti non modificabili
> - Distribuzioni con aggiornamenti immutabili o suddivisione del traffico abilitata

Se la tua applicazione non supera tutti i controlli dello stato, ma funziona comunque correttamente a uno stato inferiore, è possibile consentire alle istanze di passare i controlli dello stato con uno stato

inferiore, ad esempio `Warning`, modificando l'opzione Healthy threshold (Soglia integrità). Se la tue distribuzioni non vanno a buon fine perché non superano i controlli dello stato e devi forzare un aggiornamento, indipendentemente dallo stato del sistema, specifica l'opzione Ignore health check (Ignora controllo dello stato).

Quando specifichi una dimensione del batch per gli aggiornamenti in sequenza, Elastic Beanstalk utilizza questo valore anche per i riavvii delle applicazioni in sequenza. Utilizza i riavvii in sequenza quando è necessario riavviare il proxy e i server delle applicazioni in esecuzione sulle istanze dell'ambiente senza tempi di inattività.

Configurazione delle distribuzioni delle applicazioni

Nella console di gestione dell'ambiente, abilita e configura le distribuzioni delle versioni dell'applicazione in blocco modificando Updates and Deployments (Aggiornamenti e Distribuzioni) nella pagina Configuration (Configurazione) dell'ambiente.

Per configurare le distribuzioni (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Rolling updates and deployments (Aggiornamenti in sequenza e distribuzioni), scegliere Edit (Modifica).
5. Nella sezione Application Deployments (Distribuzioni delle applicazioni), scegli una Deployment policy (Policy di distribuzione), le impostazioni dei batch e le opzioni di controllo dello stato.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

La sezione Distribuzioni delle applicazioni della pagina Aggiornamenti in sequenza e distribuzioni presenta le seguenti opzioni per le distribuzioni delle applicazioni:

- Deployment policy (Policy di distribuzione): scegli tra le seguenti opzioni di distribuzione:
  - All at once (Tutto contemporaneamente): distribuisci la nuova versione a tutte le istanze contemporaneamente. Tutte le istanze nell'ambiente sono inutilizzabili per un breve periodo di tempo durante la distribuzione.

- Rolling (In sequenza): distribuisci la nuova versione in batch. Ogni batch non è funzionante durante la fase di distribuzione, riducendo così la capacità dell'ambiente per il numero di istanze in un batch.
- Rolling with additional batch (In sequenza con batch aggiuntivi): distribuisci la nuova versione in batch, ma prima avvia un nuovo batch di istanze per garantire la capacità completa durante il processo di distribuzione.
- Immutable (Non modificabile): distribuisci la nuova versione a un nuovo gruppo di istanze eseguendo un aggiornamento non modificabile.
- Traffic splitting (Suddivisione del traffico): distribuisci la nuova versione in un nuovo gruppo di istanze e suddividi temporaneamente il traffico client in ingresso tra la versione dell'applicazione esistente e quella nuova.

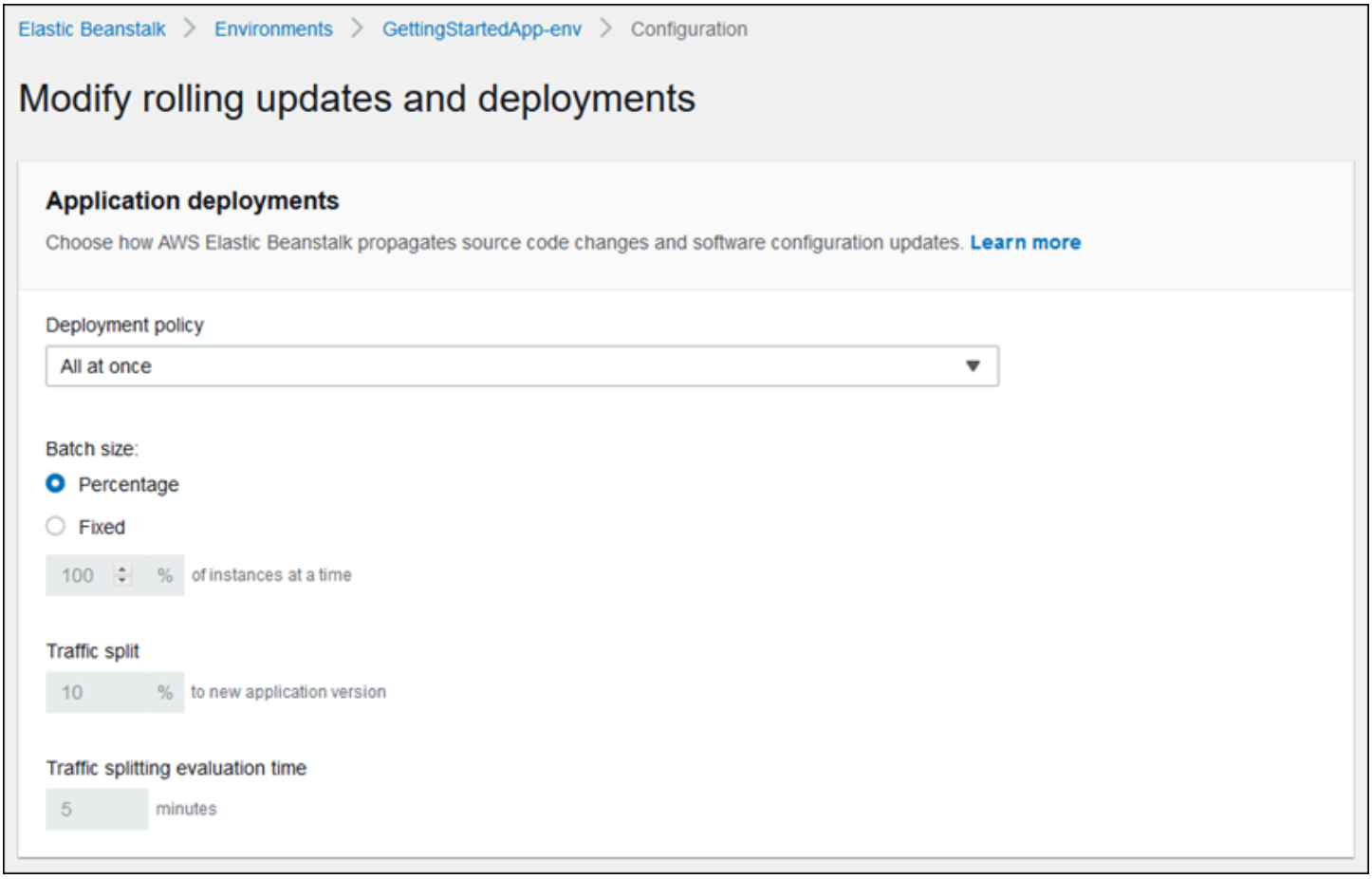
Per le policy di distribuzione Rolling (In sequenza) e In sequenza con batch aggiuntivi puoi configurare:

- Batch size (Dimensione batch): dimensione del set di istanze da distribuire in ciascuna batch.

  Scegli Percentuale per configurare una percentuale del numero totale di EC2 istanze nel gruppo Auto Scaling (fino al 100 percento) oppure scegli Fisso per configurare un numero fisso di istanze (fino al numero massimo di istanze nella configurazione Auto Scaling del tuo ambiente).

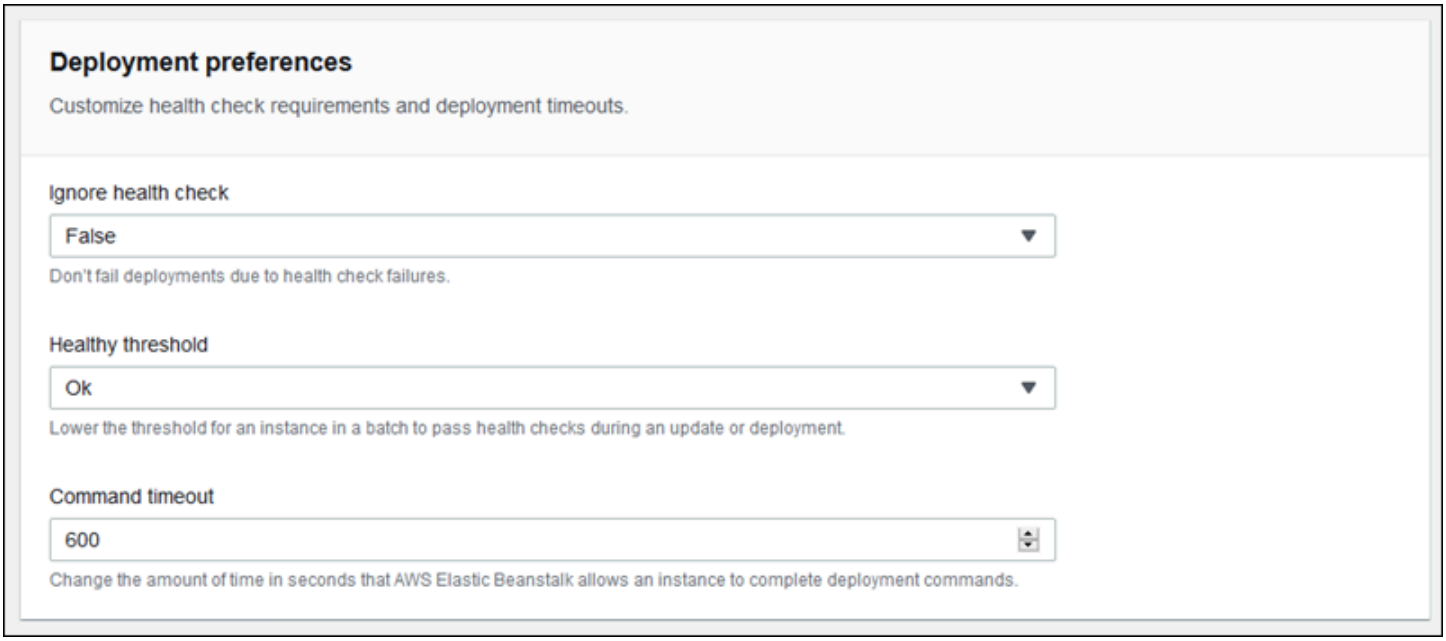
Per la policy di distribuzione Traffic splitting (Suddivisione del traffico) puoi configurare quanto segue:

- Traffic split (Divisione del traffico): percentuale iniziale di traffico client in entrata che Elastic Beanstalk passa alle istanze dell'ambiente che eseguono la nuova versione dell'applicazione che stai distribuendo.
- Traffic splitting evaluation time (Tempo di valutazione per suddivisione del traffico): il tempo di attesa espresso in minuti per Elastic Beanstalk dopo una distribuzione iniziale integra prima di procedere allo spostamento di tutto il traffico client in ingresso alla nuova versione dell'applicazione che stai distribuendo.



La sezione Deployment preferences (Preferenze di distribuzione) contiene le opzioni relative ai controlli dello stato.

- Ignore health check (Ignora controllo dello stato): impedisce il rollback di una distribuzione quando un batch non supera i controlli dello stato entro l'intervallo di tempo specificato in Command timeout (Timeout di comando).
- Healthy threshold (Soglia integrità): riduce la soglia alla quale un'istanza viene considerata integra durante le implementazioni in sequenza, gli aggiornamenti in sequenza e gli aggiornamenti non modificabili.
- Command timeout (Timeout di comando): intervallo in secondi entro il quale un'istanza deve superare i controlli dello stato prima dell'annullamento della distribuzione oppure, se l'opzione Ignore health check (Ignora controllo dello stato) è impostata, prima del passaggio al batch successivo.



Funzionamento delle distribuzioni in sequenza

Quando elabori un batch, Elastic Beanstalk distacca tutte le istanze nel batch dal sistema di bilanciamento del carico, distribuisce la nuova versione dell'applicazione, quindi ricollega le istanze. Se abiliti il drenaggio delle connessioni, Elastic Beanstalk prosciuga le connessioni esistenti dalle istanze EC2 Amazon in ogni batch prima di iniziare la distribuzione.

Dopo aver ricollegato le istanze di un batch al sistema di bilanciamento del carico, Elastic Load Balancing attende finché non passano un numero minimo di controlli dello stato Elastic Load Balancing (il valore Healthy check count threshold (Soglia di conteggio dei controlli integri)), quindi avvia l'instradamento del traffico. Se non è configurato alcun URL di controllo dello stato, questo può accadere molto rapidamente, perché un'istanza passa il controllo dello stato non appena è in grado di accettare una connessione TCP. Se è stato impostato un URL di controllo dello stato, il sistema di bilanciamento del carico non instrada il traffico verso le istanze aggiornate finché non restituiscono un codice di stato `200 OK` in risposta a una richiesta `HTTP GET` all'URL di controllo dello stato.

Elastic Beanstalk attende fino a quando tutte le istanze di un batch sono ritenute integre prima di passare al batch successivo. Con il report dello stato di base, lo stato dell'istanza dipende dallo stato del controllo di integrità di Elastic Load Balancing. Quando tutte le istanze del batch superano un numero sufficiente di controlli dello stato da essere considerate integre da Elastic Load Balancing, il batch è completo. Se il report dello stato avanzato è abilitato Elastic Beanstalk considera vari altri fattori, tra cui il risultato delle richieste in entrata. Grazie al reporting sullo stato migliorato, tutte le

istanze devono superare 12 controlli dello stato consecutivi con lo stato OK entro due minuti per gli ambienti di server Web, e 18 controlli dello stato entro tre minuti per gli ambienti lavoratore.

Se un batch di istanze non diventa integro entro il timeout del comando, la distribuzione ha esito negativo. Se una distribuzione non va a buon fine, controlla lo stato delle istanze nel tuo ambiente per ottenere informazioni sulla causa dell'errore. Quindi esegui un'altra distribuzione con una versione fissa o funzionante dell'applicazione per eseguire il rollback.

Se una distribuzione ha esito negativo dopo che uno o più batch sono stati completati con successo, i batch completati eseguono la nuova versione dell'applicazione mentre qualsiasi batch in attesa continuare a eseguire la versione precedente. È possibile identificare la versione in esecuzione sulle istanze nel tuo ambiente sulla pagina dello stato nella console. Questa pagina visualizza l'ID della distribuzione più recente eseguita su ogni istanza nell'ambiente. Se termini le istanze della distribuzione non riuscita, Elastic Beanstalk le sostituisce con istanze che eseguono la versione dell'applicazione della distribuzione più recente andata a buon fine.

Come funzionano le distribuzioni con suddivisione del traffico

Le distribuzioni con suddivisione del traffico consentono di eseguire i test canary. Puoi indirizzare il traffico client in ingresso alla nuova versione dell'applicazione per verificare l'integrità dell'applicazione prima di eseguire il commit alla nuova versione e indirizzarle tutto il traffico.

Durante una distribuzione con suddivisione del traffico, Elastic Beanstalk crea un nuovo set di istanze in un gruppo Auto Scaling temporaneo separato. Elastic Beanstalk indica quindi al sistema di bilanciamento del carico di indirizzare una certa percentuale del traffico in entrata dell'ambiente alle nuove istanze. Quindi, per un periodo di tempo configurato, Elastic Beanstalk tiene traccia dell'integrità del nuovo set di istanze. Se tutto va bene, Elastic Beanstalk sposta il traffico rimanente alle nuove istanze e le collega al gruppo Auto Scaling originale dell'ambiente, sostituendo le vecchie istanze. Quindi Elastic Beanstalk pulisce e termina le vecchie istanze e rimuove il gruppo Auto Scaling temporaneo.

> **Note**
>
> La capacità dell'ambiente non cambia durante una distribuzione con suddivisione del traffico. Elastic Beanstalk avvia lo stesso numero di istanze nel gruppo Auto Scaling temporaneo presente nel gruppo Auto Scaling originale al momento dell'avvio della distribuzione. Mantiene quindi un numero di istanze in entrambi i gruppi Auto Scaling per la durata della

distribuzione. Prendi in considerazione queste informazioni quando configuri il tempo di valutazione per la suddivisione del traffico dell'ambiente.

Il rollback della distribuzione alla versione precedente dell'applicazione è rapido e non influisce sul traffico client. Se le nuove istanze non superano i controlli dello stato o se si sceglie di interrompere la distribuzione, Elastic Beanstalk riporta il traffico alle vecchie istanze e termina quelle nuove. Puoi interrompere qualsiasi distribuzione utilizzando la pagina di panoramica dell'ambiente nella console Elastic Beanstalk e scegliendo Abort current operation (Interrompi operazione corrente) in Environment actions (Operazioni ambiente). Puoi anche chiamare l'API o il comando equivalente. AbortEnvironmentUpdate AWS CLI

Le distribuzioni con suddivisione del traffico richiedono un sistema Application Load Balancer. Elastic Beanstalk utilizza questo tipo di sistema di bilanciamento del carico per impostazione predefinita quando crei l'ambiente utilizzando la console Elastic Beanstalk o la CLI EB.

Spazi dei nomi delle opzioni di distribuzione

Puoi utilizzare le opzioni di configurazione nello spazio dei nomi `aws:elasticbeanstalk:command` per configurare le distribuzioni. Se scegli la policy di suddivisione del traffico, nello spazio dei nomi `aws:elasticbeanstalk:trafficsplitting` sono disponibili altre opzioni per questa policy.

Utilizza l'opzione `DeploymentPolicy` per impostare il tipo di distribuzione. Sono supportati i seguenti valori:

- `AllAtOnce`: disabilita le distribuzioni in sequenza e distribuisce sempre in tutte le istanze contemporaneamente.
- `Rolling`: abilita le distribuzioni in sequenza standard.
- `RollingWithAdditionalBatch`: avvia un ulteriore batch di istanze prima di avviare la distribuzione per mantenere la capacità completa.
- `Immutable`: esegue un aggiornamento non modificabile per ogni distribuzione.
- `TrafficSplitting`: esegue distribuzioni con suddivisione del traffico per eseguire il test canary delle distribuzioni dell'applicazione.

Quando abiliti le distribuzioni in sequenza, imposta le opzioni `BatchSize` e `BatchSizeType` per configurare le dimensioni di ciascun batch. Ad esempio, per distribuire il 25 percento di tutte le istanze in ogni batch, specifica le opzioni e i valori riportati di seguito.

Example.ebextensions/rolling-updates.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Rolling
    BatchSizeType: Percentage
    BatchSize: 25
```

Per distribuire a cinque istanze in ogni batch indipendentemente dal numero di istanze in esecuzione e per avere un ulteriore batch di cinque istanze in esecuzione con la nuova versione prima di ritirare alcune delle istanze, specifica le opzioni e i valori seguenti.

Example.ebextensions/rolling-additionalbatch.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: RollingWithAdditionalBatch
    BatchSizeType: Fixed
    BatchSize: 5
```

Per eseguire un aggiornamento non modificabile per ciascuna distribuzione con una soglia di controllo dello stato di Warning (Avvertenza) e procedere con la distribuzione, anche se le istanze in un batch non superano i controlli dello stato entro un timeout di 15 minuti, specifica le opzioni e i valori seguenti.

Example.ebextensions/immutable-ignorehealth.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
    HealthCheckSuccessThreshold: Warning
    IgnoreHealthCheck: true
    Timeout: "900"
```

Per eseguire distribuzioni con suddivisione del traffico, inoltrare il 15% del traffico client alla nuova versione dell'applicazione e valutare lo stato per 10 minuti, specifica le opzioni e i valori seguenti.

Example.ebextensions/traffic-splitting.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: TrafficSplitting
  aws:elasticbeanstalk:trafficsplitting:
    NewVersionPercent: "15"
    EvaluationTime: "10"
```

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Distribuzioni blu/verde con Elastic Beanstalk

Poiché AWS Elastic Beanstalk esegue un aggiornamento sul posto quando aggiorni le versioni dell'applicazione, l'applicazione potrebbe non essere disponibile agli utenti per un breve periodo di tempo. Per evitare ciò, esegui una blue/green distribuzione. A tale scopo, distribuisci la nuova versione in un ambiente separato, quindi scambia i CNAMEs due ambienti per reindirizzare immediatamente il traffico verso la nuova versione.

È inoltre necessaria una blue/green distribuzione se si desidera aggiornare un ambiente a una versione della piattaforma incompatibile. Per ulteriori informazioni, consulta the section called “Aggiornamenti della piattaforma”.

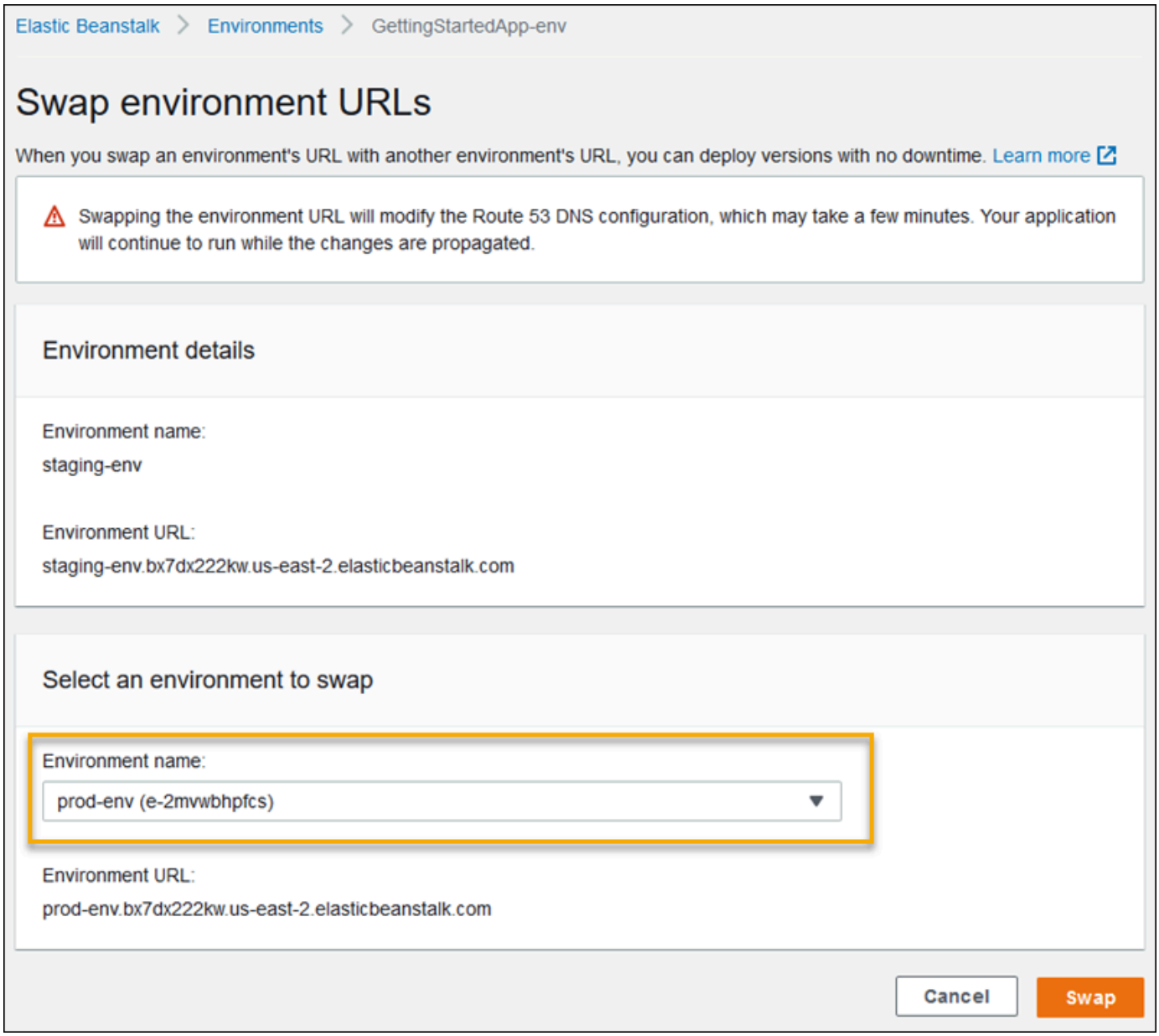
Le distribuzioni blu/verde richiedono che l'ambiente venga eseguito in modo indipendente dal database di produzione, se la tua applicazione ne impiega uno. Se l'ambiente include un database creato da Elastic Beanstalk per tuo conto, il database e la connessione dell'ambiente non vengono conservati a meno che non si adottino azioni specifiche. Se si dispone di un database che si desidera mantenere, utilizzare una delle opzioni del ciclo di vita del database Elastic Beanstalk. È possibile scegliere l'opzione Conserva per mantenere operativi il database e l'ambiente dopo aver disaccoppiato il database. Per ulteriori informazioni, consultare Ciclo di vita del database nel capitolo Configurazione degli ambienti di questa guida.

Per le istruzioni su come configurare l'applicazione per connettersi a un'istanza Amazon RDS non gestita da Elastic Beanstalk, consultare Utilizzo di Elastic Beanstalk con Amazon RDS.

Per eseguire una distribuzione blue/green

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Clona l'ambiente corrente o avvia un nuovo ambiente per eseguire la versione desiderata della piattaforma.
3. Distribuisci la nuova versione dell'applicazione nel nuovo ambiente.
4. Prova la nuova versione nel nuovo ambiente.
5. Nella pagina di panoramica dell'ambiente, scegli Azioni, quindi scegli Ambiente Swap. URLs
6. Per Environment name (Nome ambiente), selezionare l'ambiente corrente.



7. Scegli Swap (Scambia).

Elastic Beanstalk scambia i registri CNAME del vecchio e del nuovo ambiente, reindirizzando il traffico dalla versione precedente a quella nuova.

Una volta completata l'operazione di scambio da parte di Elastic Beanstalk, verifica che il nuovo ambiente risponda quando tenti di connetterti all'URL precedente dell'ambiente. Tuttavia, non terminare l'ambiente precedente finché le modifiche DNS non vengono propagate e i vecchi record DNS non sono scaduti. I server DNS non eliminano sempre i vecchi record dalla cache in base al time-to-live (TTL) configurato sui record DNS.

CI/CD Implementazione dell'integrazione con l'ambiente Elastic Beanstalk

Elastic Beanstalk si integra CI/CD con molti strumenti per automatizzare il flusso di lavoro di sviluppo delle applicazioni. CI/CD le pratiche consentono di creare, testare e distribuire automaticamente le applicazioni con un intervento manuale minimo. Continuous delivery/deployment (CD) estende l'integrazione continua (CI) automatizzando il processo di distribuzione. Puoi creare pipeline di distribuzione semplificate utilizzando AWS servizi simili CodePipeline o strumenti di terze parti come Jenkins e GitLab per garantire distribuzioni coerenti e affidabili nei tuoi ambienti Elastic Beanstalk.

Argomenti

- AWS fonti per iniziare
- Risorse aggiuntive
- Utilizzo GitHub delle azioni per la distribuzione su Elastic Beanstalk

AWS fonti per iniziare

L'elenco seguente evidenzia CI/CD gli strumenti e le AWS risorse corrispondenti che forniscono step-by-step indicazioni per la creazione di pipeline di distribuzione automatizzate negli ambienti Elastic Beanstalk:

- AWS CodePipeline— Questo tutorial del AWS Getting Started Resource Center mostra come configurare una pipeline di distribuzione continua su Elastic GitHub Beanstalk da, S3 o. AWS CodeCommit
- GitHub Azioni: scopri Utilizzo GitHub delle azioni per la distribuzione su Elastic Beanstalk come configurare i flussi di lavoro basati su YAML per configurare una pipeline di distribuzione continua su Elastic Beanstalk direttamente da. GitHub
- GitLab— Questo post sul AWS DevOps Developer Productivity Blog dimostra come configurare pipeline GitLab continue per distribuire applicazioni Node.js in ambienti Elastic Beanstalk Docker.

- Azure DevOps: questo post.NET on AWS Blog ti guida nell'implementazione di una pipeline di distribuzione continua da un repository DevOps Git di Azure a Elastic Beanstalk usando Azure Pipelines.

Risorse aggiuntive

I seguenti strumenti e risorse di terze parti possono aiutarti a implementare pipeline di distribuzione automatizzate negli ambienti Elastic Beanstalk:

- Jenkins: il plug-in AWS EBDeployment Jenkins consente la distribuzione diretta negli ambienti Elastic Beanstalk dalla pagina Jenkins Job Configuration.
- Circle CI: — The Orbs for Elastic Beanstalk fornisce pacchetti di configurazione riutilizzabili per distribuire e scalare le applicazioni su Elastic Beanstalk.
- Bitbucket Pipelines — L'articolo Deploy Elastic Beanstalk Application using Bitbucket Pipelines fornisce un esempio di configurazione di base per l'implementazione di Bitbucket Pipelines con Elastic Beanstalk.

Utilizzo GitHub delle azioni per la distribuzione su Elastic Beanstalk

GitHub Le azioni possono distribuire automaticamente l'applicazione su Elastic Beanstalk quando invii modifiche al codice nel tuo repository. L'azione Elastic Beanstalk Deploy fornisce una semplice interfaccia YAML che gestisce la creazione di versioni delle applicazioni, il caricamento di bundle di sorgenti su Amazon S3 e la distribuzione nell'ambiente Elastic Beanstalk.

Flusso di lavoro di esempio

Il flusso di lavoro di esempio seguente distribuisce un'applicazione in un ambiente Elastic Beanstalk ogni volta che si esegue il push alla filiale. `main` Crea un `.yml` file nel tuo repository sotto. `.github/workflows/`

Example GitHub Flusso di lavoro delle azioni per la distribuzione di Elastic Beanstalk

```
name: Deploy to Elastic Beanstalk

on:
  push:
    branches:
      - main
```

```
permissions:
  id-token: write
  contents: read

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v4
        with:
          role-to-assume: arn:aws:iam::123456789012:role/my-github-actions-role
          aws-region: us-east-1

      - name: Deploy to Elastic Beanstalk
        uses: aws-actions/aws-elasticbeanstalk-deploy@v1.0.0
        with:
          aws-region: us-east-1
          application-name: my-application
          environment-name: my-application-env
```

Questo flusso di lavoro controlla il tuo repository, utilizza OpenID Connect (OIDC) per AWS l'autenticazione tramite l'azione Configure Credentials, quindi distribuisce AWS l'applicazione su Elastic Beanstalk. L'azione di distribuzione impacchetta i contenuti del tuo repository, carica il pacchetto sorgente su Amazon S3, crea una nuova versione dell'applicazione e crea o aggiorna il tuo ambiente. Per impostazione predefinita, attende il completamento della distribuzione e il ripristino dello stato di integrità dell'ambiente.

Per ulteriori opzioni di configurazione ed esempi avanzati, consulta il README dell'azione Elastic Beanstalk Deploy su. GitHub

Risorse aggiuntive

- Azione Elastic Beanstalk Deploy su GitHub
- Configura l'azione Credenziali su AWS GitHub
- Configurazione di OpenID Connect in Amazon Web GitHub Services (documentazione)

Modifiche di configurazione

Quando si modificano le impostazioni delle opzioni di configurazione nella sezione Configurazione della console di gestione dell'ambiente, AWS Elastic Beanstalk propaga la modifica a tutte le risorse interessate. Queste risorse includono il sistema di bilanciamento del carico che distribuisce il traffico verso EC2 le istanze Amazon che eseguono l'applicazione, il gruppo Auto Scaling che gestisce tali istanze e le istanze stesse. EC2

Molte modifiche di configurazione possono essere applicate a un ambiente in esecuzione senza sostituire le istanze esistenti. Ad esempio, l'impostazione di un URL di controllo dello stato attiva un aggiornamento dell'ambiente per modificare le impostazioni del sistema di bilanciamento del carico, tuttavia non crea tempi di inattività perché le istanze che eseguono l'applicazione continuano a elaborare richieste durante la propagazione dell'aggiornamento.

Le modifiche della configurazione che apportano cambiamenti alla configurazione di avvio o alle impostazioni di VPC richiedono di terminare e sostituire tutte le istanze nel tuo ambiente. Ad esempio, quando modifichi il tipo di istanza o l'impostazione della chiave SSH per il tuo ambiente, le istanze devono essere terminate e sostituite. EC2 Elastic Beanstalk fornisce diverse policy che determinano come viene eseguita questa sostituzione.

- Aggiornamenti in sequenza: Elastic Beanstalk applica le modifiche alla configurazione in batch, mantenendo un numero minimo di istanze in esecuzione e distribuendo il traffico in ogni momento. Questo approccio evita tempi di inattività durante il processo di aggiornamento. Per informazioni dettagliate, vedi Aggiornamenti continui.
- Aggiornamenti non modificabili: Elastic Beanstalk avvia un gruppo Auto Scaling temporaneo al di fuori dell'ambiente con un insieme separato di istanze in esecuzione con la nuova configurazione. Quindi Elastic Beanstalk posiziona queste istanze dietro il sistema di bilanciamento del carico dell'ambiente. Le istanze vecchie e nuove servono entrambe il traffico fino a quando le nuove istanze superano i controlli di integrità. In quel momento, Elastic Beanstalk sposta le nuove istanze nel gruppo Auto Scaling dell'ambiente e termina il gruppo temporaneo e le vecchie istanze. Per informazioni dettagliate, vedi Aggiornamenti non modificabili.
- Disabilitato: Elastic Beanstalk non fa alcun tentativo per evitare tempi di inattività. Termina le istanze esistenti dell'ambiente e le sostituisce con nuove istanze in esecuzione con la nuova configurazione.

> **Warning**
>
> Alcuni criteri sostituiscono tutte le istanze durante la distribuzione o l'aggiornamento. Ciò causa la perdita di tutti i saldi Amazon EC2 Burst accumulati. Succede nei seguenti casi:
>
> - Aggiornamenti della piattaforma gestiti con sostituzione delle istanze abilitata
> - Aggiornamenti non modificabili
> - Distribuzioni con aggiornamenti immutabili o suddivisione del traffico abilitata

Tipi di aggiornamenti supportati

Impostazione degli aggiornamenti in sequenza	Ambienti con bilanciamento del carico	Ambienti di istanza singola	Ambienti di Windows Server legacy†
Disabilitato	✓ Sì	✓ Sì	✓ Sì
Propagazione basata sullo stato	✓ Sì	× No	✓ Sì
Propagazione basata sul tempo	✓ Sì	× No	✓ Sì
Non modificabile	✓ Sì	✓ Sì	× No

s†Ai fini della presente tabella, un ambiente Windows Server legacy è un ambiente basato su una configurazione della piattaforma Windows Server che utilizza una versione IIS precedente a IIS 8.5.

Argomenti

- Aggiornamenti in sequenza della configurazione dell'ambiente Elastic Beanstalk
- Aggiornamenti di ambiente non modificabili

Aggiornamenti in sequenza della configurazione dell'ambiente Elastic Beanstalk

Quando una modifica della configurazione richiede la sostituzione delle istanze, Elastic Beanstalk è in grado di eseguire l'aggiornamento in batch per evitare tempi di inattività durante la distribuzione della modifica. Durante un aggiornamento in sequenza, la capacità è solo ridotta dalle dimensioni di un singolo batch, che puoi configurare. Elastic Beanstalk seleziona un batch di istanze non funzionanti, lo termina, quindi avvia un batch con la nuova configurazione. Quando il nuovo batch avvia l'elaborazione delle richieste, Elastic Beanstalk si sposta al batch successivo.

I batch di aggiornamento delle configurazioni in sequenza possono essere elaborati periodicamente (in base al tempo), con un ritardo tra ogni batch, o in base allo stato. Per gli aggiornamenti in sequenza basati sul tempo, è possibile configurare la quantità di tempo che Elastic Beanstalk attende dopo aver completato il lancio di un batch di istanze prima di passare al batch successivo. Questa pausa temporale consente alla tua applicazione di eseguire il bootstrap e di avviare l'elaborazione delle richieste.

Con gli aggiornamenti in sequenza basati sullo stato, Elastic Beanstalk attende che le istanze in un batch superino i controlli dello stato prima di passare al batch successivo. Lo stato di un'istanza viene determinato dal sistema di reportistica dello stato, che può essere di base o migliorato. Per lo stato di base, un batch è considerato integro non appena tutte le istanze in esso contenute passano i controlli dello stato di Elastic Load Balancing (ELB).

Con il reporting sullo stato migliorato, tutte le istanze in un batch devono superare più controlli dello stato consecutivi prima che Elastic Beanstalk possa passare al batch successivo. Oltre ai controlli dello stato di ELB, che verificano solo le istanze, il reporting dello stato migliorato monitora i log delle applicazioni e lo stato delle altre risorse del tuo ambiente. In un ambiente server Web con lo stato migliorato, tutte le istanze devono superare 12 controlli dello stato nel corso di due minuti (18 controlli di oltre tre minuti per gli ambienti lavoratore). Se una qualsiasi delle istanze non supera un controllo dello stato, il conteggio ricomincia.

Se l'integrità di un batch non viene garantita entro il timeout dell'aggiornamento in sequenza (il valore predefinito è 30 minuti), l'aggiornamento viene annullato. Il timeout dell'aggiornamento in sequenza è un'opzione di configurazione disponibile nello spazio dei nomi `aws:autoscaling:updatepolicy:rollingupdate`. Se l'applicazione non supera i controlli dello stato con `Ok` ma è stabile a un livello diverso, puoi impostare l'opzione `HealthCheckSuccessThreshold` nello spazio dei nomi

`aws:elasticbeanstalk:healthreporting:system` per cambiare il livello a cui Elastic Beanstalk considera integra un'istanza.

Se il processo di aggiornamento in sequenza ha esito negativo, Elastic Beanstalk avvia un altro aggiornamento in sequenza per eseguire il rollback alla configurazione precedente. Un aggiornamento in sequenza può non riuscire a causa di controlli dello stato non riusciti oppure se l'avvio di nuove istanze comporta il superamento delle quote dell'account. Se raggiungi una quota sul numero di EC2 istanze Amazon, ad esempio, l'aggiornamento progressivo può fallire quando tenta di fornire un batch di nuove istanze. In questo caso, il rollback ha esito negativo.

Un rollback non riuscito termina il processo di aggiornamento e lascia l'ambiente in uno stato non integro. I batch non elaborati eseguono ancora le istanze con la configurazione precedente, mentre i batch con esito positivo utilizzano la nuova configurazione. Per risolvere le problematiche di un ambiente dopo un rollback non riuscito, devi prima risolvere il problema sottostante che ha portato al fallimento dell'aggiornamento, quindi avviare un altro aggiornamento dell'ambiente.

Un metodo alternativo consiste nel distribuire la nuova versione dell'applicazione in un ambiente diverso, quindi eseguire uno scambio di CNAME per reindirizzare il traffico e senza tempi di inattività. Per ulteriori informazioni, consulta Distribuzioni blu/verde con Elastic Beanstalk.

Aggiornamenti e distribuzioni in sequenza a confronto

Gli aggiornamenti continui si verificano quando modifichi le impostazioni che richiedono il provisioning di nuove EC2 istanze Amazon per il tuo ambiente. Questo include le modifiche alla configurazione del gruppo Auto Scaling come il tipo di istanza, le impostazioni della coppia di chiavi e le modifiche alla configurazione del VPC. In un aggiornamento in sequenza, ogni batch di istanze viene terminato prima che venga eseguito il provisioning di un nuovo batch per sostituirlo.

Le distribuzioni in sequenza si verificano ogni volta che distribuisci l'applicazione e possono in genere essere eseguite senza sostituire le istanze dell'ambiente. Elastic Beanstalk seleziona ogni batch non funzionante, distribuisce la nuova versione dell'applicazione, quindi rende il batch nuovamente funzionante.

Fa eccezione il caso in cui modifichi le impostazioni che richiedono la sostituzione delle istanze nello stesso momento in cui distribuisci una nuova versione dell'applicazione. Ad esempio, se modifichi le impostazioni del nome della chiave in un file di configurazione nel tuo bundle di origine e lo distribuisci nel tuo ambiente, attivi un aggiornamento in sequenza. Invece di distribuire la nuova versione dell'applicazione per ogni batch di istanze esistenti, viene eseguito il provisioning di un nuovo batch di istanze con la nuova configurazione. In questo caso non avviene una distribuzione separata, poiché alle nuove istanze viene assegnata la nuova versione dell'applicazione.

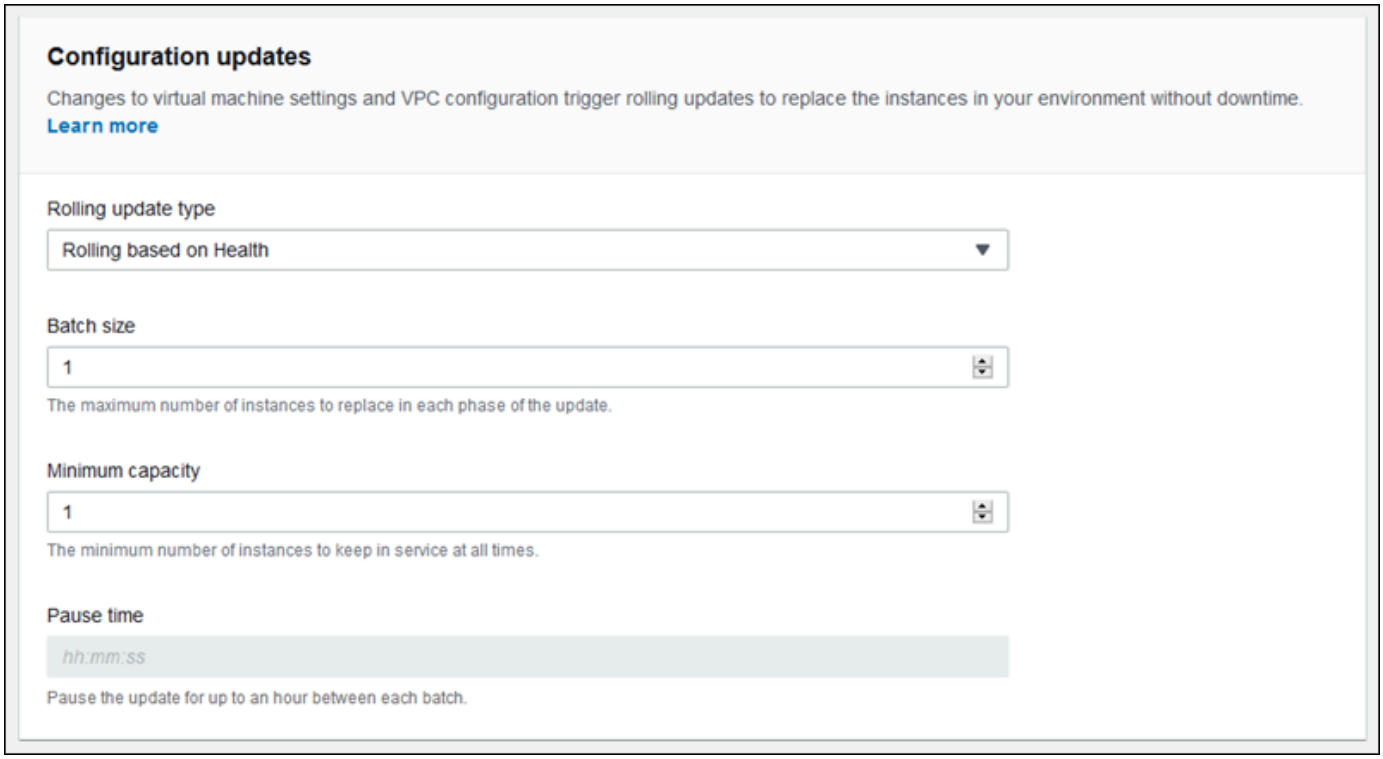
Ogni volta che le nuove istanze vengono assegnate come parte di un aggiornamento dell'ambiente, c'è una fase di distribuzione in cui il codice sorgente dell'applicazione viene distribuito alle nuove istanze e vengono applicate le impostazioni di configurazione che modificano il sistema operativo o il software sulle istanze. Le impostazioni del controllo dello stato della distribuzione (Ignore health check (Ignora controllo dello stato), Healthy threshold (Soglia integrità) e Command timeout (Timeout del comando)) si applicano allo stato con aggiornamenti in sequenza e non modificabili durante la fase di distribuzione.

Configurazione degli aggiornamenti in sequenza

Puoi attivare e configurare gli aggiornamenti in sequenza nella console Elastic Beanstalk.

Per abilitare gli aggiornamenti in sequenza

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Rolling updates and deployments (Aggiornamenti in sequenza e distribuzioni), scegliere Edit (Modifica).
5. Nella sezione Configuration updates (Aggiornamenti alla configurazione), per Rolling update type (Tipo di aggiornamento in sequenza), seleziona una delle opzioni Rolling (In sequenza).



6. Scegli le impostazioni Batch size (Dimensione batch), Minimum capacity (Capacità minima) e Pause time (Tempo di pausa).
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

La sezione Configuration updates (Aggiornamenti configurazione) della pagina Rolling updates and deployments (Aggiornamenti in sequenza e distribuzioni) presenta le seguenti opzioni per gli aggiornamenti in sequenza:

- Rolling update type (Tipo aggiornamento in sequenza): Elastic Beanstalk attende il termine dell'aggiornamento di un batch di istanze prima di passare al batch successivo, per consentire a tali istanze di finire il processo di bootstrap e cominciare a servire il traffico. Seleziona una delle opzioni seguenti:
  - Rolling based on Health (Sequenza basata sullo stato): attendi che le istanze nel batch attuale siano integre prima di ripristinare le istanze e avviare il batch successivo.
  - Rolling based on Time (Sequenza basata sul tempo): specifica un intervallo di attesa tra l'avvio di nuove istanze e il loro ripristino prima di avviare il batch successivo.
  - Immutable (Non modificabile): applica la modifica di configurazione a un nuovo gruppo di istanze eseguendo un aggiornamento non modificabile.

- Batch size (Dimensione batch): numero di istanze da sostituire in ogni batch, compreso tra **1** e **10000**. Per impostazione predefinita, questo valore è pari a un terzo della dimensione minima del gruppo Auto Scaling, arrotondata a un numero intero.
- Minimum capacity (Capacità minima): numero minimo di istanze per proseguire nell'esecuzione mentre altre istanze vengono aggiornate, compreso tra **0** e **9999**. Il valore predefinito è la dimensione minima del gruppo Auto Scaling o una dimensione inferiore alla dimensione massima del gruppo Auto Scaling, a seconda di qual è più bassa.
- Pause time (Tempo di pausa) (basato solo sul tempo): tempo di attesa dopo l'aggiornamento di un batch prima di passare al batch successivo, per consentire all'applicazione di iniziare a ricevere il traffico. Tra 0 secondi e 1 ora.

Spazio dei nomi aws:autoscaling:updatepolicy:rollingupdate

Puoi inoltre utilizzare le opzioni di configurazione nello spazio dei nomi `aws:autoscaling:updatepolicy:rollingupdate` per configurare le distribuzioni in sequenza.

Utilizza l'opzione `RollingUpdateEnabled` per abilitare gli aggiornamenti in sequenza e `RollingUpdateType` per scegliere il tipo di aggiornamento. Sono supportati i seguenti valori per `RollingUpdateType`:

- `Health` (Sequenza basata sullo stato) - Attendi fino a quando le istanze nel batch attuale sono integre prima di ripristinare le istanze e avviare il batch successivo.
- `Time` - Specifica un intervallo di attesa tra l'avvio di nuove istanze e il loro ripristino prima di avviare il batch successivo.
- `Immutable`: applica la modifica di configurazione a un nuovo gruppo di istanze eseguendo un aggiornamento non modificabile.

Quando abiliti gli aggiornamenti in sequenza, imposta le opzioni `MaxBatchSize` e `MinInstancesInService` per configurare le dimensioni di ciascun batch. Per gli aggiornamenti in sequenza basati sul tempo e sullo stato, è anche possibile configurare, rispettivamente, `PauseTime` e `Timeout`.

Ad esempio, per avviare fino a cinque istanze alla volta, mantenendo almeno due istanze in servizio, e attendere cinque minuti e 30 secondi tra i batch, specifica i seguenti valori e opzioni.

Example.ebextensions/timebased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Time
    PauseTime: PT5M30S
```

Per abilitare gli aggiornamenti in sequenza basati sullo stato, con un timeout di 45 minuti per ogni batch, specifica i seguenti valori e opzioni.

Example.ebextensions/healthbased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Health
    Timeout: PT45M
```

`Timeoute` `PauseTime` i valori devono essere specificati in ISO8601 duration:`PT#H#M#S`, dove ogni # è il numero di ore, minuti o secondi, rispettivamente.

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Aggiornamenti di ambiente non modificabili

Gli aggiornamenti non modificabili degli ambienti rappresentano un'alternativa agli aggiornamenti in sequenza. Gli aggiornamenti non modificabili dell'ambiente garantiscono un'implementazione efficiente e sicura delle modifiche di configurazione che comportano la sostituzione delle istanze. Se un aggiornamento di ambiente immutabile non riesce, il processo di rollback comporta unicamente la terminazione di un gruppo Auto Scaling. Un aggiornamento in sequenza non riuscito richiede invece l'esecuzione di un ulteriore aggiornamento in sequenza per eseguire il rollback delle modifiche.

Per eseguire un aggiornamento di ambiente immutabile, Elastic Beanstalk crea un secondo gruppo Auto Scaling temporaneo dietro il sistema di bilanciamento del carico dell'ambiente per contenere

le nuove istanze. Elastic Beanstalk avvia anzitutto una singola istanza con la nuova configurazione nel nuovo gruppo. Questa istanza serve il traffico insieme a tutte le istanze nel gruppo Auto Scaling originale in cui viene eseguita la configurazione precedente.

Quando la prima istanza supera i controlli dello stato, Elastic Beanstalk avvia ulteriori istanze con la nuova configurazione, fino ad avere un numero di istanze corrispondente a quello delle istanze nel gruppo Auto Scaling originale. Quando tutte le nuove istanze superano i controlli dello stato, Elastic Beanstalk le trasferisce al gruppo Auto Scaling originale e termina il gruppo Auto Scaling temporaneo e le vecchie istanze.

> **Note**
>
> Durante un aggiornamento di ambiente immutabile, la capacità dell'ambiente raddoppia per un breve periodo di tempo quando le istanze nel nuovo gruppo Auto Scaling iniziano a servire le richieste e prima che le istanze del gruppo Auto Scaling originale vengano terminate. Se l'ambiente ha molte istanze o tu disponi di una quota di istanze On demand bassa, accertati di avere capacità sufficiente per eseguire un aggiornamento non modificabile dell'ambiente. Se la quota è prossima, ti consigliamo di utilizzare piuttosto gli aggiornamenti in sequenza.

Gli aggiornamenti non modificabili richiedono il reporting dello stato migliorato per valutare lo stato dell'ambiente durante l'aggiornamento. Il reporting sullo stato migliorato combina i controlli dello stato standard del sistema di bilanciamento del carico e il monitoraggio delle istanze affinché le istanze in cui è in esecuzione la nuova configurazione servano le richieste.

Puoi anche utilizzare aggiornamenti non modificabili per distribuire nuove versioni della tua applicazione, come alternativa alle distribuzioni in sequenza. Quando configuri Elastic Beanstalk per utilizzare aggiornamenti non modificabili per la distribuzione dell'applicazione, tutte le istanze nell'ambiente vengono sostituite ogni volta che distribuisci una nuova versione dell'applicazione. Se una distribuzione immutabile dell'applicazione non riesce, Elastic Beanstalk annulla immediatamente le modifiche terminando il nuovo gruppo Auto Scaling. Ciò impedisce le distribuzioni parziali del parco istanze, che possono verificarsi quando una distribuzione in sequenza non riesce dopo il completamento di alcuni batch.

> **Warning**
>
> Alcuni criteri sostituiscono tutte le istanze durante la distribuzione o l'aggiornamento. Ciò causa la perdita di tutti i saldi Amazon EC2 Burst accumulati. Succede nei seguenti casi:

- Aggiornamenti della piattaforma gestiti con sostituzione delle istanze abilitata
- Aggiornamenti non modificabili
- Distribuzioni con aggiornamenti immutabili o suddivisione del traffico abilitata

Se un aggiornamento immutabile ha esito negativo, le nuove istanze caricano i log di bundle in Amazon S3 prima che vengano terminati da Elastic Beanstalk. Elastic Beanstalk lascia i log relativi a un aggiornamento immutabile non riuscito in Amazon S3 per un'ora prima di eliminarli, anziché per 15 minuti come avviene per i log di bundle e di coda.

> **Note**
>
> Se utilizzi aggiornamenti non modificabili per le distribuzioni delle versioni dell'applicazione, ma non per la configurazione, è possibile che si verifichi un errore se tenti di distribuire una versione dell'applicazione contenente modifiche di configurazione che normalmente attivano un aggiornamento in sequenza (ad esempio, configurazioni che modificano il tipo di istanza). Per evitare che ciò accada, esegui le modifiche di configurazione in un aggiornamento distinto, oppure configura aggiornamenti non modificabili per le distribuzioni e le modifiche di configurazione.

Non puoi eseguire un aggiornamento non modificabile insieme alle modifiche di configurazione delle risorse. Ad esempio, non puoi modificare le impostazioni che richiedono la sostituzione delle istanze ed eseguire al contempo l'aggiornamento di altre impostazioni, oppure eseguire una distribuzione non modificabile con file di configurazione che modificano le impostazioni di configurazione o ulteriori risorse nel codice sorgente. Se tenti di modificare le impostazioni delle risorse (ad esempio, le impostazioni del sistema di bilanciamento del carico) e nello stesso tempo esegui un aggiornamento non modificabile, Elastic Beanstalk restituisce un errore.

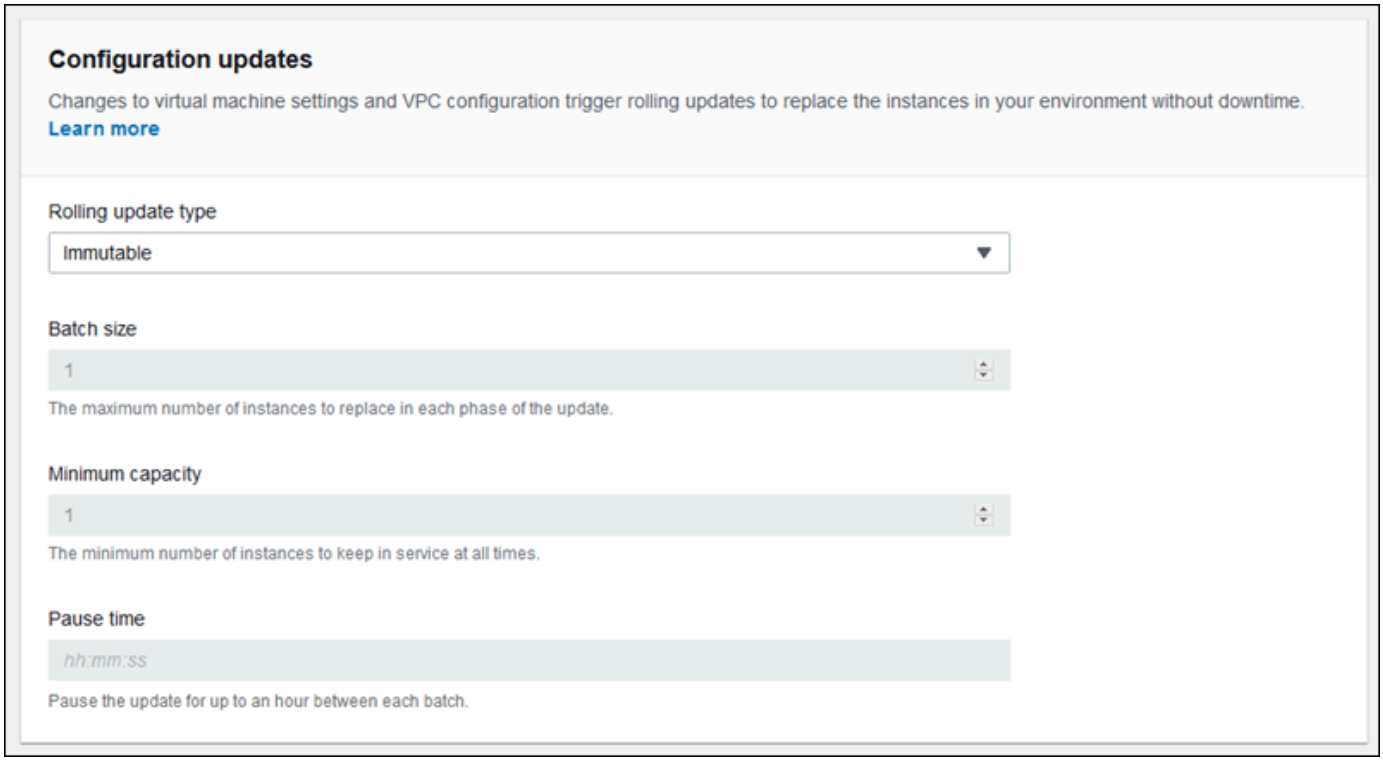
Se le modifiche di configurazione delle risorse non dipendono dalla modifica del codice sorgente o dalla configurazione delle istanze, eseguile in due aggiornamenti. In caso contrario, esegui piuttosto una distribuzione blu/verde.

Configurazione di aggiornamenti non modificabili

Puoi attivare e configurare gli aggiornamenti non modificabili nella console Elastic Beanstalk.

Attivazione degli aggiornamenti non modificabili (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Rolling updates and deployments (Aggiornamenti in sequenza e distribuzioni), scegliere Edit (Modifica).
5. Nella sezione Configuration Updates (Aggiornamenti configurazione), imposta Rolling update type (Tipo aggiornamento in sequenza) su Immutable (Non modificabile).



6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Spazio dei nomi aws:autoscaling:updatepolicy:rollingupdate

È inoltre possibile utilizzare le opzioni nello spazio dei nomi `aws:autoscaling:updatepolicy:rollingupdate` per configurare aggiornamenti non modificabili. L'esempio di file di configurazione seguente attiva aggiornamenti non modificabili per le modifiche di configurazione.

Example.ebextensions/immutable-updates.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
```

L'esempio seguente attiva aggiornamenti non modificabili per le distribuzioni e le modifiche di configurazione.

Example.ebextensions/immutable-all.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
```

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Aggiornamento della versione della piattaforma dell'ambiente Elastic Beanstalk

Elastic Beanstalk rilascia regolarmente nuove versioni della piattaforma per aggiornare tutte le piattaforme basate su Linux e su Windows Server. Le nuove versioni della piattaforma forniscono gli aggiornamenti ai componenti software esistenti e il supporto per le nuove caratteristiche e opzioni di configurazione. Per ulteriori informazioni sulle piattaforme e sulle relative versioni, consulta Glossario delle piattaforme Elastic Beanstalk.

Puoi utilizzare la console Elastic Beanstalk o la CLI EB per aggiornare la versione della piattaforma dell'ambiente. A seconda della versione della piattaforma di destinazione dell'aggiornamento, Elastic Beanstalk consiglia uno dei due metodi disponibili per eseguire gli aggiornamenti della piattaforma.

- Metodo 1: aggiornamento della versione della piattaforma dell'ambiente. Questo metodo è consigliabile quando si esegue l'aggiornamento alla versione più recente della piattaforma all'interno di un ramo di piattaforma con lo stesso runtime, server web, server applicazioni e sistema operativo, e senza modifiche nella versione principale della piattaforma. Questo è l'aggiornamento della piattaforma più comune e di routine.

- Metodo 2: eseguire una distribuzione Blue/Green. Questo metodo è consigliabile quando si esegue l'aggiornamento a una versione di piattaforma in un ramo di piattaforma diverso, con un runtime, un server Web, un server applicazioni o un sistema operativo diverso o a una diversa versione principale della piattaforma. Questo è un buon approccio quando si desidera sfruttare le nuove funzionalità di runtime o le funzionalità più recenti di Elastic Beanstalk, oppure quando si desidera abbandonare un ramo di piattaforma obsoleto o ritirato.

  La migrazione da una versione precedente della piattaforma richiede un' blue/green implementazione, poiché queste versioni della piattaforma sono incompatibili con le versioni attualmente supportate.

  La migrazione di un'applicazione Linux su Amazon Linux 2 richiede una blue/green distribuzione, poiché le versioni della piattaforma Amazon Linux 2 sono incompatibili con le precedenti versioni della piattaforma AMI Amazon Linux.

Per ulteriori assistenza nella scelta del metodo di aggiornamento della piattaforma migliore, espandi la relativa sezione della piattaforma del tuo ambiente.

Docker

Utilizza il metodo 1 per eseguire aggiornamenti della piattaforma.

Multicontainer Docker

Utilizza il metodo 1 per eseguire aggiornamenti della piattaforma.

Docker preconfigurato

Considera i casi seguenti:

- Se stai eseguendo la migrazione dell'applicazione in un'altra piattaforma, ad esempio da Go 1.4 (Docker) a Go 1.11 o da Python 3.4 (Docker) a Python 3.6, utilizza il metodo 2.
- Se stai eseguendo la migrazione dell'applicazione a una versione diversa del container Docker, ad esempio da Glassfish 4.1 (Docker) a Glassfish 5.0 (Docker), utilizza il metodo 2.
- Se stai eseguendo l'aggiornamento a una versione della piattaforma più recente senza modificare la versione del container o la versione principale, utilizza il metodo 1.

Go

Utilizza il metodo 1 per eseguire aggiornamenti della piattaforma.

Java SE

Considera i casi seguenti:

- Se stai eseguendo la migrazione dell'applicazione a una versione runtime Java diversa, ad esempio da Java 7 a Java 8, utilizza il metodo 2.
- Se stai eseguendo l'aggiornamento a una versione della piattaforma più recente senza modificare la versione runtime, utilizza il metodo 1.

Java con Tomcat

Considera i casi seguenti:

- Se stai eseguendo la migrazione in una versione runtime Java o versione del server di applicazioni Tomcat diversa, ad esempio da Java 7 con Tomcat 7 a Java 8 con Tomcat 8.5, utilizza il metodo 2.
- Se stai eseguendo la migrazione dell'applicazione tra versioni della piattaforma Tomcat con Java principali (v1.x.x, v2.x.x e v3.x.x), utilizza il metodo 2.
- Se stai eseguendo l'aggiornamento a una versione della piattaforma più recente senza modificare la versione runtime, la versione del server di applicazioni o la versione principale, utilizza il metodo 1.

.NET su Windows Server con IIS

Considera i casi seguenti:

- Se stai eseguendo la migrazione dell'applicazione in una versione del sistema operativo Windows diversa, ad esempio da Windows Server 2008 R2 a Windows Server 2016, utilizza il metodo 2.
- Se stai eseguendo la migrazione dell'applicazione tra versioni della piattaforma Windows Server principali, consulta Migrazione da versioni principali della piattaforma Windows Server e utilizza il metodo 2.
- Se l'applicazione è attualmente in esecuzione su una piattaforma Windows Server V2.x.x e stai eseguendo l'aggiornamento alla versione della piattaforma più recente, utilizza il metodo 1.

> **Note**
>
> Versioni della piattaforma Windows Server precedenti alla v2 non hanno semanticamente versioni. È possibile avviare solo la versione più recente di ciascuna di queste versioni della piattaforma principali di Windows Server e non è possibile eseguire il rollback dopo un aggiornamento.

Node.js

Utilizza il metodo 2 per eseguire gli aggiornamenti della piattaforma.

PHP

Considera i casi seguenti:

- Se stai eseguendo la migrazione dell'applicazione a una versione runtime PHP diversa, ad esempio da PHP 5.6 a PHP 7.2, utilizza il metodo 2.
- Se stai eseguendo la migrazione dell'applicazione tra versioni della piattaforma PHP principali (v1.x.x v2.x.x), utilizza il metodo 2.
- Se stai eseguendo l'aggiornamento a una versione della piattaforma più recente senza modifica della versione runtime o della versione principale, utilizza il metodo 1.

Python

Considera i casi seguenti:

- Se stai eseguendo la migrazione dell'applicazione a una versione runtime Python diversa, ad esempio da PHP 2.7 a PHP 3.6, utilizza il metodo 2.
- Se stai eseguendo la migrazione dell'applicazione tra versioni della piattaforma Python principali (v1.x.x v2.x.x), utilizza il metodo 2.
- Se stai eseguendo l'aggiornamento a una versione della piattaforma più recente senza modifica della versione runtime o della versione principale, utilizza il metodo 1.

Ruby

Considera i casi seguenti:

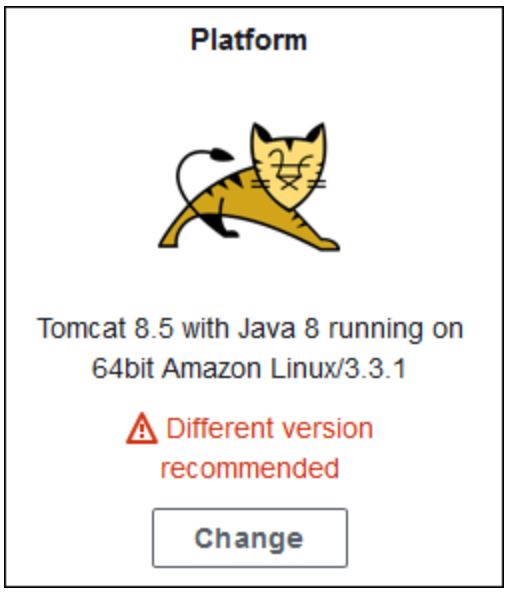
- Se stai eseguendo la migrazione a una versione runtime Ruby o versione del server di applicazioni diversa, ad esempio da Ruby 2.3 con Puma a Ruby 2.6 con Puma, utilizza il metodo 2.
- Se stai eseguendo la migrazione dell'applicazione tra versioni della piattaforma Ruby principali (v1.x.x v2.x.x), utilizza il metodo 2.
- Se stai eseguendo l'aggiornamento a una versione della piattaforma più recente senza modificare la versione runtime, la versione del server di applicazioni o la versione principale, utilizza il metodo 1.

Metodo 1: aggiornamento della versione della piattaforma dell'ambiente

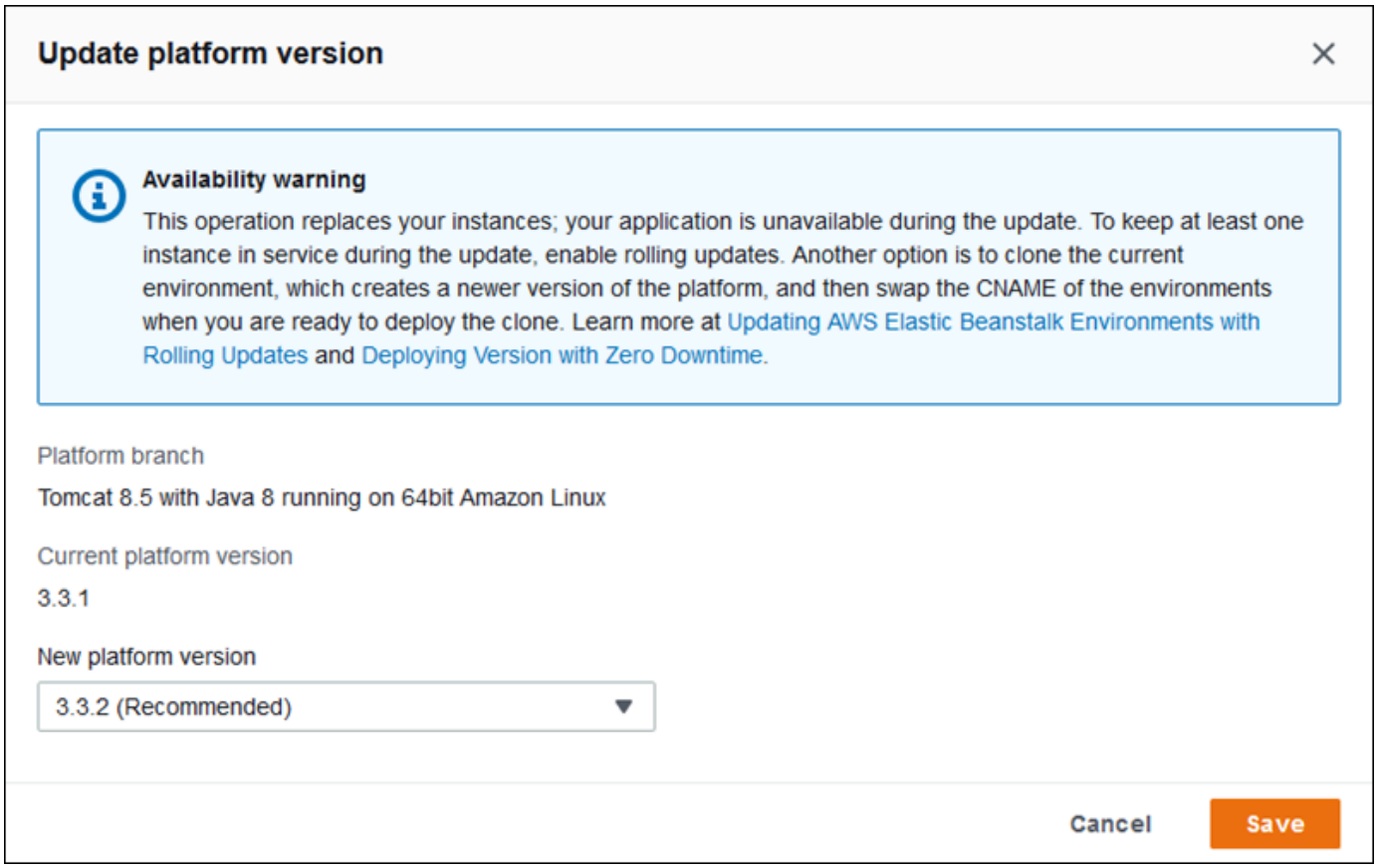
Utilizza questo metodo per eseguire l'aggiornamento alla versione più recente del ramo della piattaforma del tuo ambiente. Se hai già creato un ambiente utilizzando una versione della piattaforma precedente, o hai aggiornato l'ambiente da una versione precedente, puoi anche usare questo metodo per ripristinare una versione della piattaforma precedente, purché si trovi sullo stesso ramo della piattaforma.

Per aggiornare la versione della piattaforma del tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina panoramica dell'ambiente, in Platform (Piattaforma), scegliere Change (Cambia).



4. Nella finestra di dialogo Update platform version (Aggiorna versione piattaforma), selezionare una versione di piattaforma. La versione più recente (consigliata) della piattaforma nel ramo viene selezionata automaticamente. È possibile eseguire l'aggiornamento a qualsiasi versione utilizzata in passato.



5. Scegliere Save (Salva).

Per semplificare ulteriormente gli aggiornamenti della piattaforma, Elastic Beanstalk è in grado di gestirli per tuo conto. Puoi configurare l'ambiente in modo che gli aggiornamenti delle versioni minori e di patch vengano applicati automaticamente durante una finestra di manutenzione settimanale configurabile. Elastic Beanstalk applica aggiornamenti gestiti senza tempi di inattività o riduzione di capacità e annulla l'aggiornamento immediatamente se le istanze che eseguono l'applicazione sulla nuova versione non superano i controlli dello stato. Per informazioni dettagliate, vedi Aggiornamenti gestiti della piattaforma.

Metodo 2: eseguire una distribuzione Blue/Green

Utilizza questo metodo per eseguire l'aggiornamento a un ramo di piattaforma diverso con un runtime, un server Web, un server applicazioni o un sistema operativo diverso o a una diversa

versione principale della piattaforma. Questo è in genere necessario quando desideri sfruttare nuove funzionalità runtime o la funzionalità Elastic Beanstalk più recente. È necessario anche quando si esegue la migrazione da un ramo di piattaforma obsoleto o ritirato.

Quando esegui la migrazione tra versioni della piattaforma principali o a versioni della piattaforma con aggiornamenti dei componenti principali, è probabile che l'applicazione, o alcuni suoi aspetti, non funzionino come previsto sulla nuova versione della piattaforma e potrebbero richiedere delle modifiche.

Prima di eseguire la migrazione, esegui l'aggiornamento del computer di sviluppo locale alle versioni runtime più recenti e altri componenti della piattaforma verso cui intendi eseguire la migrazione. Verifica che l'applicazione funzioni ancora come previsto e apporta le eventuali correzioni di codice e modifiche necessarie. Quindi utilizza la seguente procedura di best practice per migrare in modo sicuro il tuo ambiente alla nuova versione della piattaforma.

Per migrare l'ambiente a una versione della piattaforma con aggiornamenti principali

1. Creare un nuovo ambiente, utilizzando la nuova versione della piattaforma di destinazione, e distribuire il codice dell'applicazione in tale ambiente. Il nuovo ambiente deve essere nell'applicazione Elastic Beanstalk che contiene l'ambiente di cui si sta eseguendo la migrazione. Non terminare ancora l'ambiente esistente.
2. Utilizzare il nuovo ambiente per eseguire la migrazione dell'applicazione. In particolare:
   - Trovare e risolvere eventuali problemi di compatibilità dell'applicazione che non è stato possibile rilevare durante la fase di sviluppo.
   - Verificare che eventuali personalizzazioni eseguite dall'applicazione utilizzando i file di configurazione funzionino correttamente nel nuovo ambiente. Queste potrebbero includere impostazioni di opzioni, pacchetti installati aggiuntivi, policy di sicurezza personalizzate e script o i file di configurazione installati in istanze degli ambienti.
   - Se l'applicazione utilizza un’Amazon Machine Image (AMI), creare una nuova AMI personalizzata in base all'AMI della nuova versione della piattaforma. Per ulteriori informazioni, vedi Utilizzo di un'immagine Amazon Machine Image (AMI) personalizzata nell'ambiente Elastic Beanstalk. In particolare, questo è richiesto se l'applicazione utilizza la piattaforma Windows Server con un'AMI personalizzata e stai eseguendo la migrazione a una versione della piattaforma Windows Server V2. In questo caso, consulta anche Migrazione da versioni principali della piattaforma Windows Server.

Continuare con il test e l'implementazione delle correzioni finché l'applicazione nel nuovo ambiente non sembra soddisfacente.

3. Introdurre il nuovo ambiente nell'ambiente di produzione scambiando il suo CNAME con quello dell'ambiente di produzione esistente. Per informazioni dettagliate, consulta Distribuzioni blu/verde con Elastic Beanstalk.
4. Se lo stato del nuovo ambiente in produzione è soddisfacente, terminare l'ambiente precedente. Per informazioni dettagliate, vedi Terminazione di un ambiente Elastic Beanstalk.

Aggiornamenti gestiti della piattaforma

AWS Elastic Beanstalk rilascia regolarmente aggiornamenti della piattaforma per fornire correzioni, aggiornamenti software e nuove funzionalità. Grazie agli aggiornamenti gestiti della piattaforma, è possibile configurare il proprio ambiente per eseguire automaticamente l'aggiornamento alla versione più recente di una piattaforma durante una finestra di manutenzione pianificata. L'applicazione in uso rimane operativa durante il processo di aggiornamento senza alcun impatto sulla capacità. Gli aggiornamenti gestiti sono disponibili per gli ambienti sia a istanza singola sia con bilanciamento del carico.

> **Note**
>
> Questa caratteristica non è disponibile sulle versioni della piattaforma Windows Server precedenti alla versione 2 (v2).

Puoi configurare il tuo ambiente in modo che vengano applicati automaticamente gli aggiornamenti della versione di patch o entrambi gli aggiornamenti, ovvero della patch e della versione secondaria. Gli aggiornamenti della piattaforma gestita non supportano gli aggiornamenti tra i rami della piattaforma (aggiornamenti a diverse versioni principali di componenti della piattaforma, come sistema operativo, runtime o componenti Elastic Beanstalk), poiché possono introdurre modifiche non compatibili con le versioni precedenti.

Quando si abilitano gli aggiornamenti gestiti della piattaforma, è inoltre possibile configurare Elastic Beanstalk in modo che vengano sostituite tutte le istanze dell'ambiente durante la finestra di manutenzione, anche se un aggiornamento della piattaforma non è disponibile. La sostituzione di tutte le istanze dell'ambiente è utile se l'applicazione rileva bug e problemi di memoria quando viene eseguita per un lungo periodo.

Negli ambienti creati il 25 novembre 2019 o versioni successive utilizzando la console Elastic Beanstalk, gli aggiornamenti gestiti vengono abilitati per impostazione predefinita quando possibile. Gli aggiornamenti gestiti richiedono l'abilitazione dell'integrità avanzata . L'integrità avanzata è abilitata per impostazione predefinita quando si seleziona uno dei preset di configurazione e disabilitata quando si seleziona Configurazione personalizzata. La console non è in grado di abilitare gli aggiornamenti gestiti per le versioni precedenti della piattaforma che non supportano l'integrità avanzata o quando l'integrità avanzata è disabilitata. Quando la console abilita gli aggiornamenti gestiti per un nuovo ambiente, la finestra di aggiornamento settimanale viene impostata su un giorno casuale della settimana in un momento casuale. Il livello di aggiornamento è impostato su Minore e patch e la sostituzione istanza è disabilitata. È possibile disattivare o riconfigurare gli aggiornamenti gestiti prima del passaggio di creazione dell'ambiente finale.

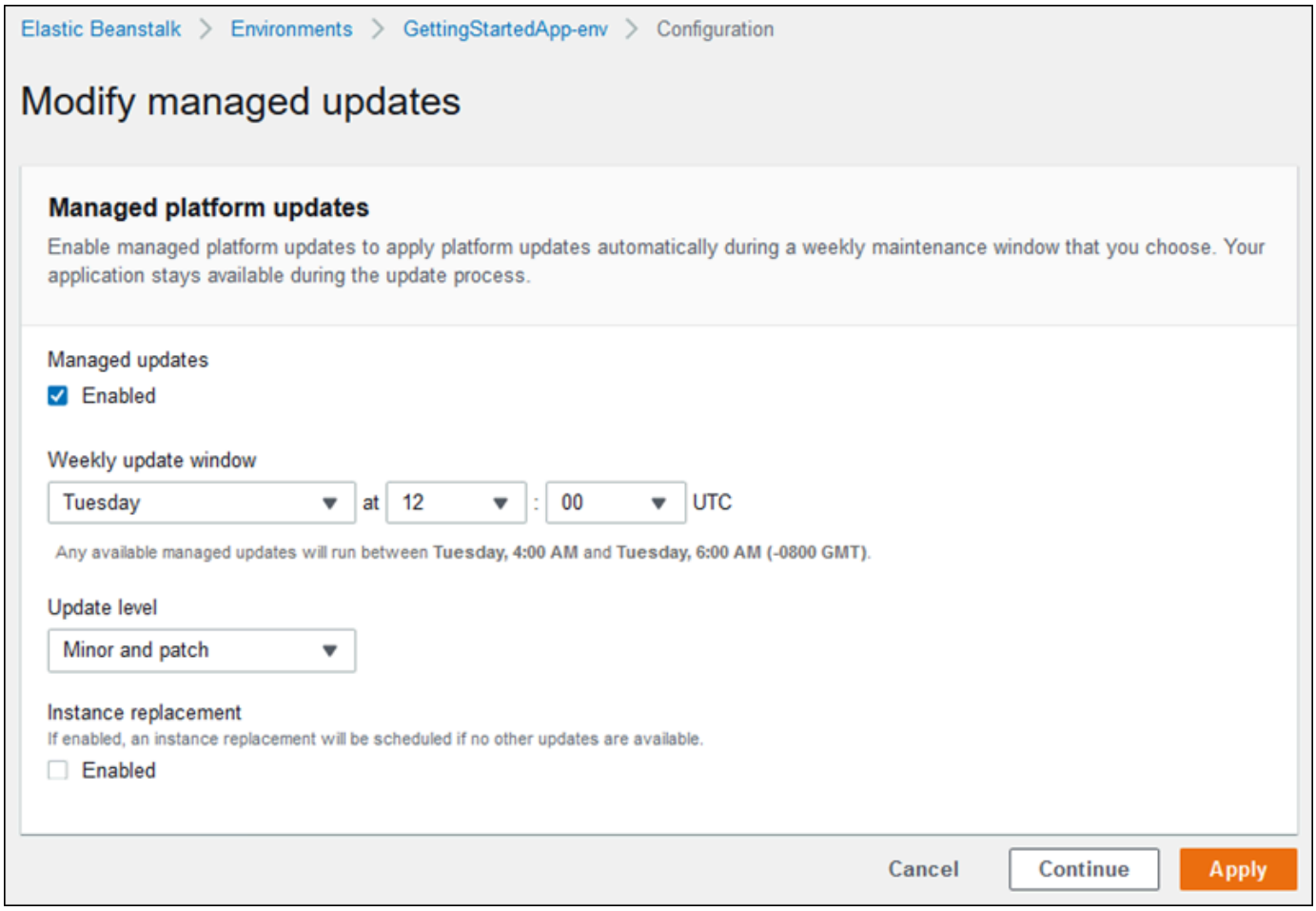
Per un ambiente esistente, utilizzare la console Elastic Beanstalk in qualsiasi momento per configurare gli aggiornamenti della piattaforma gestita.

> **Important**
>
> Un numero elevato di ambienti Beanstalk in un account AWS può presentare un rischio di problemi di limitazione (della larghezza di banda della rete) durante gli aggiornamenti gestiti. Numero elevato è una quantità relativa che dipende dalla precisione con cui pianifichi gli aggiornamenti gestiti per i tuoi ambienti. Oltre 200 ambienti in un account pianificati attentamente potrebbero causare problemi di limitazione (della larghezza di banda della rete), sebbene anche un numero inferiore possa essere problematico.
> Per bilanciare il carico di risorse per gli aggiornamenti gestiti, consigliamo di suddividere le finestre di manutenzione programmata per gli ambienti in un unico account.
> Inoltre, è bene considerare una strategia multi-account. Per ulteriori informazioni, consulta Organizzare AWS l'ambiente utilizzando più account sul sito Web AWS Whitepapers & Guides.

Per configurare gli aggiornamenti della piattaforma gestita

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria Managed updates (Aggiornamenti gestiti), scegliere Edit (Modifica).

5. Disabilitare o abilitare Aggiornamenti gestiti.
6. Se gli aggiornamenti gestiti sono abilitati, selezionare una finestra di manutenzione e quindi selezionare un livello di aggiornamento.
7. (Facoltativo) Seleziona Instance replacement (Sostituzione istanza) per abilitare la sostituzione settimanale delle istanze.



8. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Il reporting dello stato avanzato indica se lo stato dell'applicazione è sufficientemente integro da considerare riuscito l'aggiornamento della piattaforma. Per istruzioni, consulta Abilitazione del reporting dello stato avanzato Elastic Beanstalk.

Sezioni

- Autorizzazioni necessarie per eseguire gli aggiornamenti gestiti della piattaforma
- Finestra di manutenzione degli aggiornamenti gestiti
- Aggiornamenti della versione secondaria e di patch

- Aggiornamenti di ambiente non modificabili
- Gestione degli aggiornamenti gestiti
- Spazio dei nomi delle opzioni delle operazioni gestite

Autorizzazioni necessarie per eseguire gli aggiornamenti gestiti della piattaforma

Per avviare un aggiornamento della piattaforma per tuo conto, Elastic Beanstalk necessita dell'autorizzazione. Per ottenere queste autorizzazioni, Elastic Beanstalk assume il ruolo di servizio degli aggiornamenti gestiti. Quando si utilizza il ruolo di servizio predefinito per l'ambiente, la console Elastic Beanstalk lo utilizza anche come ruolo di servizio degli aggiornamenti gestiti. La console assegna la policy gestita `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` al ruolo di servizio. Questa policy dispone di tutte le autorizzazioni Elastic Beanstalk necessarie per eseguire gli aggiornamenti gestiti della piattaforma.

Per ulteriori informazioni su altri modi per impostare il ruolo del servizio degli aggiornamenti gestiti, consulta the section called “Ruoli di servizio”.

> **Note**
>
> Se si utilizzano i file di configurazione per estendere l'ambiente e includere risorse aggiuntive, potrebbe essere necessario aggiungere autorizzazioni al ruolo del servizio degli aggiornamenti gestiti. In genere è necessario aggiungere autorizzazioni quando si fa riferimento a queste risorse in altre sezioni o file mediante il nome.

Se un aggiornamento non riesce, puoi individuare il motivo dell'errore nella pagina relativa agli Aggiornamenti gestiti.

Finestra di manutenzione degli aggiornamenti gestiti

Quando AWS rilascia una nuova versione della piattaforma dell'ambiente, Elastic Beanstalk pianifica un aggiornamento gestito della piattaforma durante la finestra di manutenzione settimanale successiva. Le finestre di manutenzione sono della durata di due ore. Elastic Beanstalk avvia un aggiornamento pianificato durante la finestra di manutenzione. L'aggiornamento potrebbe non essere completato fino alla fine della finestra temporale.

> **Note**
>
> Nella maggior parte dei casi, Elastic Beanstalk pianifica l'aggiornamento gestito durante la successiva finestra di manutenzione settimanale. Il sistema considera diversi aspetti della sicurezza dell'aggiornamento e della disponibilità del servizio per programmare gli aggiornamenti gestiti. In alcuni rari casi, un aggiornamento potrebbe non essere pianificato per la prima finestra di manutenzione pianificata. In questo caso, il sistema effettua un nuovo tentativo durante la finestra di manutenzione successiva. Per applicare manualmente l'aggiornamento gestito, scegliere Apply now (Applica ora) come spiegato nella sezione Gestione degli aggiornamenti gestiti in questa pagina.

Aggiornamenti della versione secondaria e di patch

È possibile abilitare gli aggiornamenti della piattaforma gestiti per applicare solo gli aggiornamenti della versione di patch o per entrambi gli aggiornamenti, ovvero delle versione secondaria e di patch. Gli aggiornamenti della versione di patch forniscono miglioramenti delle prestazioni e correzioni dei bug e possono includere piccole modifiche di configurazione alle opzioni di configurazione, agli script e al software non di istanza. Gli aggiornamenti della versione secondaria forniscono il supporto per le nuove caratteristiche di Elastic Beanstalk. Gli aggiornamenti gestiti della piattaforma non prevedono l'applicazione di aggiornamenti della versione principale, che potrebbero introdurre modifiche non compatibili con la versione precedente.

In un numero di versione della piattaforma il secondo numero è l'aggiornamento della versione secondaria e il terzo è la versione di patch. Ad esempio, in una versione di piattaforma 2.0.7 la versione secondaria è 0 e la versione di patch è 7.

Aggiornamenti di ambiente non modificabili

Gli aggiornamenti della piattaforma gestiti eseguono aggiornamenti non modificabili dell'ambiente per effettuare l'aggiornamento del tuo ambiente a una nuova versione della piattaforma. Gli aggiornamenti non modificabili aggiornano l'ambiente senza modificarlo e senza interrompere il servizio di nessuna istanza, prima di confermare che le istanze che eseguono la nuova configurazione superano i controlli dello stato.

In un aggiornamento non modificabile, Elastic Beanstalk distribuisce tutte le istanze attualmente in esecuzione con la nuova versione della piattaforma. Le nuove istanze iniziano ad accogliere le richieste insieme a quelle che eseguono la versione precedente. Se il nuovo set di istanze trasferisce

tutti i controlli dello stato, Elastic Beanstalk termina il set di istanze precedente, lasciando solo le istanze con la nuova versione.

Gli aggiornamenti della piattaforma gestiti eseguono sempre aggiornamenti non modificabili, anche quando vengono applicati al di fuori della finestra di manutenzione. Se si modifica la versione della piattaforma da Dashboard (Pannello di controllo), Elastic Beanstalk applica la policy di aggiornamento che hai scelto per gli aggiornamenti della configurazione.

> **Warning**
>
> Alcuni criteri sostituiscono tutte le istanze durante la distribuzione o l'aggiornamento. Ciò causa la perdita di tutti i saldi burst di Amazon EC2 accumulati. Succede nei seguenti casi:
>
> - Aggiornamenti della piattaforma gestiti con sostituzione delle istanze abilitata
> - Aggiornamenti non modificabili
> - Distribuzioni con aggiornamenti immutabili o suddivisione del traffico abilitata

Gestione degli aggiornamenti gestiti

La console Elastic Beanstalk mostra informazioni dettagliate sugli aggiornamenti gestiti nella pagina Managed updates overview (Panoramica aggiornamenti gestiti).

Per visualizzare informazioni sugli aggiornamenti gestiti (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegliere Managed Updates (Aggiornamenti gestiti).

Nella sezione Panoramica aggiornamenti gestiti sono disponibili informazioni sugli aggiornamenti gestiti pianificati e in sospeso. Nella sezione History (Cronologia) sono elencati gli aggiornamenti riusciti e i tentativi non riusciti.

Puoi scegliere di applicare subito un aggiornamento pianificato, anziché attendere la finestra di manutenzione.

Per applicare subito un aggiornamento della piattaforma gestito (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegliere Managed Updates (Aggiornamenti gestiti).
4. Scegli Apply now (Applica ora).
5. Verificare i dettagli dell'aggiornamento e quindi scegliere Apply (Applica).

Quando si applica un aggiornamento della piattaforma gestito all'esterno della finestra di manutenzione, Elastic Beanstalk esegue un aggiornamento non modificabile. Se si aggiorna la piattaforma dell'ambiente dal pannello di controllo o utilizzando un client diverso, Elastic Beanstalk usa il tipo di aggiornamento selezionato per le modifiche di configurazione.

Se non disponi di un aggiornamento gestito programmato, l'ambiente potrebbe già eseguire la versione più recente. Altri motivi per non avere un aggiornamento programmato includono:

- È disponibile un aggiornamento della versione secondaria, ma l'ambiente è configurato in modo che vengano applicati automaticamente solo gli aggiornamenti della versione di patch.
- Non è stata eseguita alcuna scansione dell'ambiente da quando l'aggiornamento è stato rilasciato. In genere Elastic Beanstalk controlla la presenza di aggiornamenti ogni ora.
- Un aggiornamento è sospeso o è già in corso.

Quando viene avviata la finestra di manutenzione o quando si sceglie Apply now (Applica ora), viene attivato lo stato in sospeso degli aggiornamenti pianificati prima dell'esecuzione.

Spazio dei nomi delle opzioni delle operazioni gestite

Puoi utilizzare le opzioni di configurazione negli spazi dei nomi `aws:elasticbeanstalk:managedactions` e `aws:elasticbeanstalk:managedactions:platformupdate` per abilitare e configurare gli aggiornamenti della piattaforma gestiti.

L'opzione `ManagedActionsEnabled` attiva gli aggiornamenti della piattaforma gestiti. Imposta questa opzione su `true` per abilitare gli aggiornamenti della piattaforma gestiti e utilizza altre opzioni per configurare il comportamento degli aggiornamenti.

Utilizzalo `PreferredStartTime` per configurare l'inizio della finestra di manutenzione settimanale nel formato*day*::*hour*. *minute*

Imposta `UpdateLevel` `minor` o `patch` per applicare entrambi gli aggiornamenti, ovvero della versione secondaria e di patch, oppure solo gli aggiornamenti della versione di patch.

Quando gli aggiornamenti della piattaforma gestiti sono abilitati, puoi abilitare la sostituzione delle istanze impostando l'opzione `InstanceRefreshEnabled` su `true`. Quando questa impostazione è abilitata, Elastic Beanstalk esegue un aggiornamento non modificabile sull'ambiente ogni settimana, anche se non è disponibile una nuova versione della piattaforma.

Il seguente file di configurazione di esempio abilita gli aggiornamenti gestiti della piattaforma per gli aggiornamenti della versione di patch con una finestra di manutenzione che inizia ogni martedì alle 9:00 UTC.

Example.ebextensions/ managed-platform-update .config

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Migrazione dell'applicazione da una versione legacy della piattaforma

Se hai distribuito un'applicazione Elastic Beanstalk che utilizza un tipo di versione della piattaforma legacy, dovresti migrare l'applicazione a un nuovo ambiente utilizzando un tipo di versione della piattaforma non legacy, in modo da ottenere l'accesso alle nuove funzionalità. Se non sei sicuro di eseguire la tua applicazione utilizzando una versione della piattaforma legacy, puoi verificarlo nella console Elastic Beanstalk. Per istruzioni, consulta Per verificare se si utilizza una versione di piattaforma legacy.

Quali nuove funzionalità mancano nelle versioni legacy della piattaforma?

Le piattaforme legacy non supportano le seguenti caratteristiche:

- I file di configurazione, come descritto nell'argomento Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`)
- I controlli dello stato ELB, come descritto nell'argomento Report dello stato di base

- I profili delle istanze, come descritto nell'argomento Gestione dei profili dell'istanza Elastic Beanstalk
- VPCs, come descritto nell'Utilizzo di Elastic Beanstalk con Amazon VPCargomento
- I piani di dati, come descritto nell'argomento Aggiunta di un database all'ambiente Elastic Beanstalk
- I piani lavoratore, come descritto nell'argomento Ambienti worker Elastic Beanstalk
- Gli ambienti di istanza singola, come descritto nell'argomento Tipi di ambiente
- I tag, come descritto nell'argomento Tagging delle risorse negli ambienti Elastic Beanstalk
- Gli aggiornamenti in sequenza, come descritto nell'argomento Aggiornamenti in sequenza della configurazione dell'ambiente Elastic Beanstalk

Perché alcune versioni di piattaforma sono contrassegnate come legacy?

Alcune versioni della piattaforma meno recenti non supportano le ultime caratteristiche di Elastic Beanstalk. Queste versioni sono contrassegnate come (legacy) sulla pagina di panoramica dell'ambiente nella console Elastic Beanstalk.

Per verificare se si utilizza una versione di piattaforma legacy

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina panoramica dell'ambiente, visualizzare il nome di Platform (Piattaforma).

   L'applicazione utilizza una versione di piattaforma legacy se (legacy) appare accanto al nome della piattaforma.

Per migrare la tua applicazione

1. Distribuisci l'applicazione in un nuovo ambiente. Per istruzioni, vai su Creazione di un ambiente Elastic Beanstalk.
2. Se disponi di un'istanza database di Amazon RDS, aggiorna il gruppo di sicurezza del database per consentire l'accesso al gruppo di sicurezza EC2 per il nuovo ambiente. Per istruzioni su come trovare il nome del tuo gruppo di sicurezza EC2 utilizzando la Console di AWS gestione, consulta. EC2 gruppi di sicurezza Per ulteriori informazioni sulla configurazione del gruppo di sicurezza EC2, vai alla sezione relativa all'autorizzazione dell'accesso di rete per un gruppo

di sicurezza di Amazon EC2 nel capitolo Gestione dei gruppi di sicurezza DB della Guida per l'utente di Amazon Relational Database Service.

3. Scambia l'URL dell'ambiente. Per istruzioni, vai su Distribuzioni blu/verde con Elastic Beanstalk.
4. Per terminare il tuo ambiente precedente. Per istruzioni, vai su Terminazione di un ambiente Elastic Beanstalk.

> **Note**
>
> Se utilizzi AWS Identity and Access Management (IAM), dovrai aggiornare le tue politiche per includere CloudFormation Amazon RDS (se applicabile). Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con AWS Identity and Access Management.

Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2

Questa sezione descrive come eseguire la migrazione dell'applicazione utilizzando uno dei seguenti percorsi di migrazione.

- Migrazione da una ramificazione della piattaforma Amazon Linux 2 a una ramificazione della piattaforma Amazon Linux 2023.
- Esegui la migrazione da un ramo della piattaforma Amazon Linux AMI (AL1) a un ramo della piattaforma Amazon Linux 2023 (consigliato) o a un ramo della piattaforma Amazon Linux 2.

Argomenti

- Migrazione da Amazon Linux 2 ad Amazon Linux 2023
- Migrazione da Amazon Linux AMI (AL1) a AL2 o AL2 023

Migrazione da Amazon Linux 2 ad Amazon Linux 2023

Questo argomento descrive come eseguire la migrazione dell'applicazione da una ramificazione della piattaforma Amazon Linux 2 a una ramificazione della piattaforma Amazon Linux 2023.

Differenze e compatibilità

Tra le piattaforme Elastic Beanstalk e AL2 AL2 023

Esiste un alto grado di compatibilità tra le piattaforme Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Sebbene ci siano alcune differenze da notare:

- Instance Metadata Service Version 1 (IMDSv1): l'impostazione predefinita dell'IMDSv1opzione Disabilita è impostata su 023 piattaforme. `true` AL2 L'impostazione predefinita è sulle piattaforme. `false` AL2
- strumento di istanza pkg-repo: lo pkg-repo strumento non è disponibile per gli ambienti che eseguono su AL2 piattaforme 023. Tuttavia, è comunque possibile applicare manualmente gli aggiornamenti del pacchetto e del sistema operativo a un'istanza 023. AL2 Per ulteriori informazioni, consulta Gestione dei pacchetti e degli aggiornamenti del sistema operativo nella Guida per l'utente di Amazon Linux 2023
- HTTPd Configurazione Apache: il `httpd.conf` file Apache per le piattaforme AL2 023 presenta alcune impostazioni di configurazione diverse da quelle per: AL2
  - Per impostazione predefinita, nega l'accesso all'intero file system del server. Queste impostazioni sono descritte in Protezione dei file del server per impostazione predefinita nella pagina Suggerimenti per la sicurezza del sito Web Apache.
  - Nega l'accesso alla configurazione `.htaccess` in tutte le directory, ad eccezione di quelle specificamente abilitate. Questa impostazione è descritta in Protezione delle impostazioni di sistema nella pagina Suggerimenti per la sicurezza del sito Web Apache. Nella pagina Tutorial Apache HTTP Server: file .htaccess è riportato che questa impostazione può aiutare a migliorare le prestazioni.
  - Nega l'accesso ai file con lo schema dei nomi `.ht*`. Questa impostazione impedisce ai client Web di visualizzare i file `.htaccess` e `.htpasswd`.

  È possibile modificare qualsiasi impostazione di configurazione precedente per l'ambiente. Per ulteriori informazioni, consulta Configurare HTTPD di Apache.
- Supporto per variabili di ambiente multilinea: le piattaforme AL2 023 supportano valori multilinea per variabili di ambiente e segreti nelle configurazioni dei servizi systemd. Le piattaforme Amazon Linux 2 non supportano valori di variabili di ambiente multilinea. Questo miglioramento consente di utilizzare segreti e valori di configurazione multilinea su piattaforme 023. AL2 Per ulteriori informazioni sull'utilizzo delle variabili e dei segreti di ambiente, vedere. the section called “Valori multilinea”
- CloudWatch inoltro di log personalizzato: l'agente (`awslogs`pacchetto) obsoleto di CloudWatch Logs non è disponibile sulle piattaforme 023. AL2 Se disponi di configurazioni di inoltro dei log personalizzate che installano e utilizzano l'agente obsoleto, devi aggiornare i file di configurazione

per utilizzare l'`awslogs` CloudWatch agente unificato durante la migrazione da Amazon Linux 2 a 023. AL2 Per ulteriori informazioni, consulta the section called “Streaming di file di log personalizzati”.

Differenze specifiche della piattaforma

Oltre alle differenze del sistema operativo di base, esistono differenze specifiche della piattaforma tra le piattaforme di runtime Amazon Linux 2 e AL2 023:

- Ramificazione della piattaforma .NET: la strategia di ramificazione della piattaforma .NET differisce tra Amazon Linux 2 e 023. AL2 Su Amazon Linux 2, la piattaforma .NET Core mantiene una finestra rotante delle versioni principali.NET all'interno di un unico ramo della piattaforma. Nella versione AL2 023, ogni ramo della piattaforma è associato a una versione principale.NET specifica (ad esempio, .NET 9, .NET 10).

  Se distribuisci applicazioni dipendenti dal framework (applicazioni che si basano sul runtime .NET installato sulla piattaforma), devi selezionare un ramo della piattaforma che corrisponda alla versione.NET di destinazione dell'applicazione. Se distribuisci applicazioni autonome (applicazioni che raggruppano il proprio runtime .NET), puoi utilizzare qualsiasi ramo della piattaforma AL2 023 .NET indipendentemente dalla versione.NET dell'applicazione, poiché l'applicazione non dipende dal runtime installato sulla piattaforma. Per ulteriori informazioni, consulta the section called “Raggruppamento delle applicazioni”.
- Selezione della versione di Node.js: la piattaforma Node.js su Amazon Linux 2 supporta la specificazione di una versione Node.js nel `package.json` file dell'applicazione. La piattaforma Node.js su AL2 023 non supporta questa funzionalità. È necessario utilizzare la versione Node.js predefinita fornita dal ramo della piattaforma. Per ulteriori informazioni sulla gestione delle versioni di Node.js, vedere the section called “Configurazione delle dipendenze”.
- Versione del server Ruby Puma: la piattaforma Ruby su Amazon Linux 2 ignora la versione di Puma specificata nel `Gemfile.lock` file dell'applicazione e utilizza la versione Puma predefinita della piattaforma. La piattaforma Ruby su AL2 023 rispetta la versione Puma specificata in, se presente. `Gemfile.lock` Se non viene specificata alcuna versione, la piattaforma installa la versione Puma predefinita della piattaforma.
- Disponibilità dei pacchetti PHP: alcuni pacchetti disponibili sulle piattaforme PHP Amazon Linux 2 non sono disponibili sulle piattaforme PHP AL2 023:
  - Pacchetti client MySQL: *`mysql`* i pacchetti client `mysql-devel` e da riga di comando non sono installati su 023 piattaforme PHP. AL2 Se la tua applicazione richiede la connettività del

database MySQL, usa `mysqli` PHP `pdo_mysql` o le estensioni, disponibili su entrambe le piattaforme.

- Strumenti Compass e Ruby: i `rubygems` pacchetti `ruby-devel` e per il supporto del framework Compass CSS non sono installati sulle piattaforme PHP 023. AL2 Compass è obsoleto. Prendi in considerazione l'utilizzo di moderni strumenti di preelaborazione CSS come alternative.

- Strumenti di controllo della versione Go: il sistema di controllo della versione di Bazaar (`bzr`) non è disponibile sulle piattaforme AL2 023 Go. Bazaar è obsoleto e non è incluso nel repository dei pacchetti 023. AL2 Usa invece Git, Mercurial o Subversion per il controllo della versione, tutti disponibili sulle piattaforme AL2 023 Go.

Tra i sistemi operativi Amazon Linux

Per ulteriori informazioni sulle differenze tra i sistemi operativi Amazon Linux 2 e Amazon Linux 2023, consulta Confronto tra Amazon Linux 2 e Amazon Linux 2023 nella Guida per l'utente di Amazon Linux 2023.

Per ulteriori informazioni su Amazon Linux 2023, consulta Cos'è Amazon Linux 2023? nella Guida per l'utente di Amazon Linux 2023.

Processo di migrazione generale

Quando sei pronto per passare alla produzione, Elastic Beanstalk blue/green richiede una distribuzione per eseguire l'aggiornamento. Di seguito sono riportate le procedure ottimali generali consigliate per la migrazione con una blue/green procedura di distribuzione.

Preparazione al test per la migrazione

Prima di implementare l'applicazione e iniziare il test, consulta le informazioni contenute nella sezione precedente Differenze e compatibilità. Consulta anche il materiale di riferimento citato in tale sezione, Confronto tra Amazon Linux 2 e Amazon Linux 2023 nella Guida per l'utente di Amazon Linux 2023. Prendi nota delle informazioni specifiche di tale sezione che si applicano o possono applicarsi all'applicazione e alla configurazione.

Panoramica delle fasi di migrazione

1. Crea un nuovo ambiente basato su un ramo della piattaforma AL2 023.
2. Distribuisci la tua applicazione nell'ambiente AL2 023 di destinazione.

L'ambiente di produzione esistente rimarrà attivo e inalterato, mentre si esegue l'iterazione attraverso i test e le modifiche apportate al nuovo ambiente.

3. Testa accuratamente l'applicazione nel nuovo ambiente.
4. Quando l'ambiente AL2 023 di destinazione è pronto per essere avviato alla produzione, sostituisci i CNAMEs due ambienti per reindirizzare il traffico verso il nuovo ambiente 023. AL2

Fasi di migrazione dettagliate e best practice

Per una procedura di blue/green distribuzione più dettagliata, consulta. Distribuzioni blu/verde con Elastic Beanstalk

Per indicazioni più specifiche e best practice dettagliate, consulta il metodo blu/verde.

Altri riferimenti per pianificare la migrazione

I seguenti riferimenti possono offrire informazioni aggiuntive per pianificare la migrazione.

- Piattaforme supportate da Elastic Beanstalk in Piattaforme AWS Elastic Beanstalk
- Cronologia dei rami di piattaforma ritirati
- the section called “Piattaforme Linux”
- Domande frequenti sul ritiro della piattaforma

Migrazione da Amazon Linux AMI (AL1) a AL2 o AL2 023

Se l'applicazione Elastic Beanstalk è basata su una ramificazione della piattaforma AMI Amazon Linux, consulta questa sezione per informazioni su come migrare gli ambienti dell'applicazione ad Amazon Linux 2 o Amazon Linux 2023. Le ramificazioni della piattaforma della generazione precedente, basati su AMI Amazon Linux, ora sono obsolete.

Ti consigliamo di eseguire la migrazione ad Amazon Linux 2023, poiché è più recente di Amazon Linux 2. Il sistema operativo Amazon Linux 2 raggiungerà la fine del supporto prima di Amazon Linux 2023, quindi la migrazione ad Amazon Linux 2023 ti consente di beneficiare di un periodo di supporto più lungo.

È importante tenere presente che esiste un alto grado di compatibilità tra le piattaforme Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Sebbene alcune aree presentino delle differenze:

l'opzione Instance Metadata Service Version 1 (IMDSv1) predefinita, il supporto per lo strumento di istanza pkg-repo e alcune configurazioni di Apache. HTTPd Per ulteriori informazioni, consulta Amazon Linux 2023

Differenze e compatibilità

Non è garantito che i rami AL2 della piattaforma basata su AL2 023/ siano retrocompatibili con l'applicazione esistente. È inoltre importante tenere presente che, anche se il codice dell'applicazione viene implementato correttamente nella nuova versione della piattaforma, potrebbe comportarsi o funzionare in modo diverso a causa delle differenze del sistema operativo e del tempo di esecuzione.

Sebbene Amazon Linux AMI e AL2 023/ AL2 condividano lo stesso kernel Linux, si differenziano per i seguenti aspetti: il sistema di inizializzazione, `libc` le versioni, la catena di strumenti del compilatore e vari pacchetti. Per ulteriori informazioni, consulta Amazon Linux 2 FAQs.

Il servizio Elastic Beanstalk ha anche aggiornato versioni specifiche della piattaforma di runtime, strumenti di compilazione e altre dipendenze.

Si consiglia pertanto di dedicare il tempo necessario, testare accuratamente l'applicazione in un ambiente di sviluppo e apportare le modifiche necessarie.

Processo di migrazione generale

Quando sei pronto per passare alla produzione, Elastic Beanstalk blue/green richiede una distribuzione per eseguire l'aggiornamento. Di seguito sono riportate le procedure ottimali generali consigliate per la migrazione con una blue/green procedura di distribuzione.

Preparazione al test per la migrazione

Prima di implementare l'applicazione e iniziare il test, consulta le informazioni contenute nella sezione Considerazioni per tutte le piattaforme Linux seguente. Consulta inoltre le informazioni relative alla piattaforma in uso nella sezione Considerazioni specifiche della piattaforma seguente. Prendi nota delle informazioni specifiche di tale sezione che si applicano o possono applicarsi all'applicazione e alla configurazione.

Panoramica delle fasi di migrazione

1. Crea un nuovo ambiente basato su un ramo della piattaforma AL2 o AL2 023. Ti consigliamo di migrare a un ramo della piattaforma AL2 023.
2. Distribuisci l'applicazione nell'ambiente 023/ di destinazione AL2. AL2

L'ambiente di produzione esistente rimarrà attivo e inalterato, mentre si esegue l'iterazione attraverso i test e le modifiche apportate al nuovo ambiente.

3. Testa accuratamente l'applicazione nel nuovo ambiente.
4. Quando AL2 l'ambiente AL2 023/ di destinazione è pronto per essere avviato alla produzione, sostituisci i due ambienti per reindirizzare il traffico verso il nuovo ambiente. CNAMEs

Fasi di migrazione dettagliate e best practice

Per una procedura di blue/green distribuzione più dettagliata, consulta. Distribuzioni blu/verde con Elastic Beanstalk

Per indicazioni più specifiche e best practice dettagliate, consulta il metodo blu/verde.

Altri riferimenti per pianificare la migrazione

I seguenti riferimenti possono offrire informazioni aggiuntive per pianificare la migrazione.

- Confronto tra Amazon Linux 2 e Amazon Linux 2023 Guida per l'utente di Amazon Linux 2023.
- Cos'è Amazon Linux 2023? nella Guida per l'utente di Amazon Linux 2023
- Piattaforme supportate da Elastic Beanstalk in Piattaforme AWS Elastic Beanstalk
- Cronologia dei rami di piattaforma ritirati
- the section called “Piattaforme Linux”
- Domande frequenti sul ritiro della piattaforma

Considerazioni per tutte le piattaforme Linux

La tabella seguente illustra le considerazioni di cui è necessario tenere conto quando si pianifica la migrazione di un'applicazione alla AL2 versione 023/. AL2 Queste considerazioni si applicano a qualsiasi piattaforma Elastic Beanstalk Linux, indipendentemente da specifici linguaggi di programmazione o server applicativi.

Area	Modifiche e informazioni
File di configura zione	Sulle AL2 piattaforme AL2 023/, è possibile utilizzare i file di configurazione come in precedenza e tutte le sezioni funzionano allo stesso modo. Tuttavia, le impostazi

Area	Modifiche e informazioni
	oni specifiche potrebbero non funzionare allo stesso modo delle piattaforme AMI Amazon Linux precedenti. Esempio: • Alcuni pacchetti software che si installano utilizzando un file di configurazione potrebbero non essere disponibili su AL2 023/ AL2 o i loro nomi potrebbero essere cambiati. • Alcune opzioni di configurazione specifiche della piattaforma sono state spostate dagli spazi dei nomi specifici della piattaforma a spazi dei nomi diversi e indipendenti dalla piattaforma. • I file di configurazione del proxy forniti nella directory `.ebextensions/nginx` devono spostarsi nella directory hook della piattaforma `.platform/nginx` . Per informazioni dettagliate, vedi Configurazione del proxy inverso. Si consiglia di utilizzare gli hook della piattaforma per eseguire codice personali zzato sulle istanze dell'ambiente. È comunque possibile utilizzare comandi e comandi container nei file di configurazione `.ebextensions` , ma non sono così facili da usare. Ad esempio, scrivere script di comando all'interno di un file YAML può essere ingombrante e difficile da testare. È comunque necessario utilizzare i `.ebextensions` file di configurazione per qualsiasi script che necessita di un riferimento a una risorsa AWS CloudFormation .
Hook della piattaforma	AL2 platforms ha introdotto un nuovo modo per estendere la piattaforma dell'ambi ente aggiungendo file eseguibili alle directory di hook sulle istanze dell'ambiente. Con le versioni precedenti della piattaforma Linux, potresti aver usato gli hook della piattaforma personalizzati. Questi hook non erano progettati per piattaforme gestite e non erano supportati, ma potrebbero funzionare in modi utili in alcuni casi. Con le versioni AL2 02/3 AL2 della piattaforma, gli hook di piattaforma personalizzati non funzionano. È necessario eseguire la migrazione di eventuali hook ai nuovi hook della piattaforma. Per dettagli, consulta Hook della piattaforma.

Area	Modifiche e informazioni
Server proxy supportati	AL2Le versioni della AL2 piattaforma 023/ supportano gli stessi server proxy inversi di ogni piattaforma supportata nelle versioni della piattaforma AMI Amazon Linux. Tutte le versioni della piattaforma AL2 023/AL2; utilizzano nginx come server proxy inverso predefinito, ad eccezione delle piattaforme ECS e Docker. Le piattafor me Tomcat, Node.js, PHP e Python supportano anche HTTPD di Apache come alternativa. Tutte le piattaforme abilitano la configurazione del server proxy in modo uniforme, come descritto in questa sezione. Tuttavia, la configurazione del server proxy è leggermente diversa da quella presente nella AMI Amazon Linux. Queste sono le differenze per tutte le piattaforme: • L'impostazione predefinita è nginx: il server proxy predefinito su AL2 tutte le versioni della piattaforma AL2 023/ è nginx. Nelle versioni della piattaforma AMI Amazon Linux di Tomcat, PHP e Python, il server proxy predefinito era HTTPD di Apache. • Spazio dei nomi coerente: tutte le versioni della AL2 piattaforma AL2 023/ utilizzano lo spazio dei nomi per configurare il server proxy. `aws:elast icbeanstalk:environment:proxy` Nelle versioni della piattaforma AMI Amazon Linux questa era una decisione che dipendeva dalla piattaforma e Node.js utilizzava uno spazio dei nomi diverso. • Posizione del file di configurazione: è necessario inserire i file di configurazione del proxy nelle `.platform/httpd` directory `.platform/nginx` and di tutte le versioni della piattaforma 023/. AL2 AL2 Sulle versioni della piattaforma AMI Amazon Linux queste posizioni erano rispettivamente `.ebextensions/ngin x` e `.ebextensions/httpd` . Per le modifiche alla configurazione del proxy specifiche della piattaforma, consulta the section called “Considerazioni specifiche della piattaforma”. Per informazioni sulla configurazione del proxy sulle piattaforme AL2 AL2 023/, vedere. Configura zione del proxy inverso

Area	Modifiche e informazioni
Modifiche alla configurazione del proxy	Esistono modifiche alla configurazione del proxy che si applicano uniformemente a tutte le piattaforme oltre alle modifiche alla configurazione del proxy specifiche per ciascuna piattaforma. È importante fare riferimento a entrambi per configurare con precisione gli ambienti. • Tutte le piattaforme: vedi Configurazione del proxy inverso • Specificità della piattaforma: consulta the section called “Considerazioni specifiche della piattaforma”.
Profilo dell’istanza	AL2023/ AL2 Le piattaforme richiedono la configurazione di un profilo di istanza. La creazione dell'ambiente potrebbe temporaneamente avere esito positivo senza profilo, ma l'ambiente potrebbe mostrare errori subito dopo la creazione quando le azioni che richiedono un profilo di istanza iniziano a fallire. Per informazioni dettagliate, vedi the section called “Profili delle istanze”.
Integrità migliorata	AL2Le versioni della AL2 piattaforma 023/ abilitano l'integrità avanzata per impostazione predefinita. Questa è una modifica se non si utilizza la console Elastic Beanstalk per creare gli ambienti. La console abilita l'integrità avanzata per impostazione predefinita quando possibile, indipendentemente dalla versione della piattaforma. Per informazioni dettagliate, vedi the section called “Monitoraggio e report avanzati sull’integrità”.
AMI personalizzata	Se il tuo ambiente utilizza un'AMI personalizzata, crea una nuova AMI basata su AL2 023/ AL2 per il nuovo ambiente utilizzando una piattaforma Elastic Beanstalk 023/. AL2 AL2
Piattaforme personalizzate	Le versioni gestite AMIs della piattaforma AL2 AL2 023/ non supportano piattaforme personalizzate.

Considerazioni specifiche della piattaforma

In questa sezione vengono illustrate le considerazioni relative alla migrazione per specifiche piattaforme Elastic Beanstalk Linux.

Docker

La famiglia di rami della piattaforma Docker basata su Amazon Linux AMI (AL1) include tre rami di piattaforma. Per ognuno di essi consigliamo un percorso di migrazione diverso.

AL1 Filiale della piattaforma	Percorso di migrazione verso lo AL2 023/ AL2
Docker multi-container gestito da Amazon ECS in esecuzione su Amazon Linux AMI () AL1	Filiali della piattaforma Docker 02/3 basate su ECS AL2 AL2 Le filiali della piattaforma Docker AL2 023/ basate su ECS offrono un percorso di migrazione semplice per gli ambienti in esecuzione sul ramo della AL2 piattaforma Docker multi-container. AL1 • Come la precedente AL1 filiale Multi-container Docker, le AL2 filiali della piattaforma AL2 023/ utilizzano Amazon ECS per coordinare la distribuzione di più contenitori Docker in un cluster Amazon ECS in un ambiente Elastic Beanstalk. • Le filiali della AL2 piattaforma AL2 023/ supportano tutte le funzionalità del precedente ramo Multi-container Docker. AL1 • I rami della AL2 piattaforma AL2 023/ supportano anche lo stesso file v2. `Dockerrun.aws.json` Per ulteriori informazioni sulla migrazione delle applicazioni in esecuzione sul ramo della piattaforma Multi-container Docker Amazon Linux verso un ramo Amazon ECS in esecuzione AL2 sul ramo della piattaforma AL2 023/, consulta. ???
Docker in esecuzione su AMI Amazon Linux () AL1 Docker preconfigurato (Glassfis	Docker in esecuzione su un ramo della piattaforma 02/3 AL2 AL2 Ti consigliamo di migrare le applicazioni in esecuzione su ambienti basati su Docker preconfigurato (Glassfish 5.0) o Docker in esecuzione su Amazon Linux AMI (AL1) verso ambienti basati su Docker Running on Amazon Linux 2 o Docker Running su 023 rami della piattaforma. AL2 Se l'ambiente è basato sul ramo di piattaforma Docker preconfigurato (Glassfish 5.0), consulta the section called “Tutorial - GlassFish su Docker: percorso verso Amazon Linux 2023”.

AL1 Filiale della piattaforma	Percorso di migrazione verso lo AL2 023/ AL2
h 5.0) con AMI Amazon Linux () AL1	La tabella seguente elenca le informazioni sulla migrazione specifiche per il ramo della piattaforma Docker Running su 023/. AL2 AL2

Area	Modifiche e informazioni
Storage	Elastic Beanstalk configura Docker per utilizzare i driver di archiviazione per archiviare le immagini e i dati del container Docker. Su AMI Amazon Linux, Elastic Beanstalk utilizza il driver di storage Device Mapper. Per migliorare le prestazioni, Elastic Beanstalk esegue il provisioning di un volume Amazon EBS aggiuntivo. Nelle versioni della piattaforma AL2 023/ AL2 Docker, Elastic Beanstalk utilizza il driver di archiviazione OverlayFS e raggiunge prestazioni ancora migliori senza richiedere più un volume separato. Con AMI Amazon Linux, se usi l'opzione `BlockDeviceMappings` dello spazio dei nomi `aws:autoscaling:launchconfiguration` per aggiungere volumi di storage personalizzati a un ambiente Docker, è consigliabile aggiungere anche il volume Amazon EBS `/dev/xvdcz` fornito da Elastic Beanstalk. Tuttavia, Elastic Beanstalk non esegue più il provisioning di questo volume ed è pertanto necessario rimuoverlo dai file di configurazione. Per informazioni dettagliate, vedi the section called “Configurazione Docker su AMI Amazon Linux (precedente ad Amazon Linux 2)”.
Autenticazione del repository privato	Quando fornisci un file di autenticazione generato da Docker per connetterti a un repository privato, non devi più convertirlo nel formato precedente richiesto dalle versioni della piattaforma Amazon Linux AMI Docker. AL2023/ Le versioni della piattaforma AL2 Docker supportano il nuovo formato. Per informazioni dettagliate, vedi the section called “Autenticazione con archivi di immagini”.
Server proxy	AL2023/ Le versioni della piattaforma AL2 Docker non supportano contenitori autonomi che non funzionano dietro un server proxy. Nelle versioni della piattaforma Docker AMI Amazon Linux, questo

AL1 Filiale della piattaforma	Percorso di migrazione verso lo AL2 023/ AL2	
	Area	Modifiche e informazioni
		era possibile tramite il valore `none` dell'opzione `ProxyServer` nello spazio dei nomi `aws:elasticbeanstalk:environment:proxy` .

Go

La tabella seguente elenca le informazioni sulla migrazione per le versioni della piattaforma AL2 023/ nella AL2 piattaforma Go.

Area	Modifiche e informazioni
Passaggio della porta	Sulle AL2 piattaforme AL2 023/, Elastic Beanstalk non trasmette un valore di porta al processo applicativo tramite la variabile di ambiente `PORT` È possibile simulare questo comportamento per il processo configurando autonomamente una proprietà di ambiente `PORT`. Tuttavia, se hai più processi e si prevede che Elastic Beanstalk passi valori di porta incrementali ai processi (5000, 5100, 5200 e così via), dovresti modificare l'implementazione. Per dettagli, consulta Configurazione del proxy inverso.

Amazon Corretto

La tabella seguente elenca le informazioni di migrazione per le versioni della piattaforma Corretto nella piattaforma Java SE.

Area	Modifiche e informazioni
Corretto e OpenJDK a confronto	Per implementare la piattaforma Java, Standard Edition (Java SE), le filiali della AL2 piattaforma AL2 023/ utilizzano Amazon Corretto, una AWS distribuzione dell'Open Java Development Kit (OpenJDK). Le versioni precedenti della piattaforma Java SE Elastic Beanstalk utilizzano i pacchetti OpenJDK inclusi con AMI Amazon Linux.

Area	Modifiche e informazioni
Strumenti di compilazione	AL2AL2 Le piattaforme 023/ dispongono di versioni più recenti degli strumenti di compilazione:, e. `gradle` `maven` `ant`
Gestione file JAR	Sulle AL2 piattaforme AL2 023/, se il pacchetto sorgente (file ZIP) contiene un singolo file JAR e nessun altro file, Elastic Beanstalk non rinomina più il file JAR in. `application.jar` È possibile rinominare solo se si invia un file JAR da solo, non all'interno di un file ZIP.
Passaggio della porta	Sulle AL2 piattaforme AL2 023/, Elastic Beanstalk non trasmette un valore di porta al processo applicativo tramite la variabile di ambiente. `PORT` È possibile simulare questo comportamento per il processo configurando autonomamente una proprietà di ambiente `PORT`. Tuttavia, se hai più processi e si prevede che Elastic Beanstalk passi valori di porta incrementali ai processi (5000, 5100, 5200 e così via), dovresti modificare l'implementazione. Per dettagli, consulta Configurazione del proxy inverso.
Java 7	Elastic Beanstalk non AL2 supporta AL2 un ramo della piattaforma 023/Java 7. Se si dispone di un'applicazione Java 7, eseguire la migrazione a Corretto 8 o Corretto 11.

Tomcat

La tabella seguente elenca le informazioni sulla migrazione per le versioni della piattaforma AL2 023/ nella piattaforma TomcatAL2 .

Area	Modifiche e informazioni
Opzioni di configurazione	Nelle versioni della AL2 piattaforma AL2 023/, Elastic Beanstalk supporta solo un sottoinsieme delle opzioni di configurazione e dei valori delle opzioni nel namespace. `aws:elasticbeanstalk:environment:proxy` Ecco le informazioni sulla migrazione per ogni opzione.

<table>
<tr><th>Area</th><th>Modifiche e informazioni</th></tr>
<tr><td></td><td>
<table>
<tr><th>Opzione</th><th>Informazioni sulla migrazione</th></tr>
<tr><td><code>GzipCompression</code></td><td>Non supportato AL2 nelle versioni della AL2 piattaforma 023/.</td></tr>
<tr><td><code>ProxyServer</code></td><td>AL2023/ Le versioni della piattaforma AL2 Tomcat supportano sia i server proxy nginx che Apache HTTPD versione 2.4. Tuttavia, Apache versione 2.2 non è supportato.

Nelle versioni della piattaforma AMI Amazon Linux, il proxy predefinito era Apache 2.4. Se hai utilizzato l'impostazione proxy predefinita e hai aggiunto file di configurazione proxy personalizzati, la configurazione del proxy dovrebbe continuare a funzionare su 023/. AL2 AL2 Tuttavia, se è stato utilizzato il valore dell'opzione <code>apache/2.2</code> , è ora necessario migrare la configurazione proxy ad Apache versione 2.4.</td></tr>
</table>

L'<code>XX:MaxPermSize</code> opzione nel <code>aws:elasticbeanstalk:container:tomcat:jvmoptions</code> namespace non è supportata nelle versioni della piattaforma 023/. AL2 AL2 L'impostazione JVM per modificare la dimensione della generazione permanente si applica solo a Java 7 e versioni precedenti e pertanto non è applicabile alle versioni della piattaforma 023/. AL2 AL2
</td></tr>
<tr><td>Percorso dell'applicazione</td><td>Sulle AL2 piattaforme AL2 023/, il percorso della directory dell'applicazione sulle EC2 istanze Amazon del tuo ambiente è. <code>/var/app/current</code> Era <code>/var/lib/tomcat8/webapps</code> sulle piattaforme AMI Amazon Linux.</td></tr>
</table>

Node.js

La tabella seguente elenca le informazioni sulla migrazione per le versioni della piattaforma AL2 023/ nella AL2 piattaforma Node.js.

Area	Modifiche e informazioni
Versioni Node.js installate	Sulle AL2 piattaforme AL2 023/, Elastic Beanstalk gestisce diversi rami della piattaforma Node.js e installa solo l'ultima versione della versione principale di Node.js corrispondente al ramo della piattaforma su ciascuna versione della piattaforma. Ad esempio, ogni versione della piattaforma nel ramo della piattafor ma Node.js 12 ha solo Node.js 12.x.y installato per impostazione predefinita. Nelle versioni della piattaforma AMI Amazon Linux, abbiamo installato le versioni multiple di Node.js su ciascuna versione della piattaforma e mantenuto solo un singolo ramo di piattaforma. Scegliere il ramo della piattaforma Node.js che corrisponde alla versione principale Node.js di cui ha bisogno l'applicazione.
Nomi dei file di log HTTPD di Apache	Sulle AL2 piattaforme AL2 023/, se si utilizza il server proxy Apache HTTPD, i nomi dei file di registro HTTPD sono `access_log` `eerror_log` , il che è coerente con tutte le altre piattaforme che supportano Apache HTTPD. Nelle versioni della piattaforma AMI Amazon Linux, questi file di log sono stati denominati rispettiv amente `access.log` e `error.log` . Per informazioni dettagliate sui nomi dei file di log e sui percorsi di tutte le piattafor me, consulta the section called “Come Elastic Beanstalk configura i log CloudWatc h”.
Opzioni di configura zione	Sulle AL2 piattaforme AL2 023/, Elastic Beanstalk non supporta le opzioni di configurazione nel namespace. `aws:elasticbeanstalk:contai ner:nodejs` Alcune delle opzioni hanno alternative. Ecco le informazioni sulla migrazione per ogni opzione. | Opzione	Informazioni sulla migrazione
`NodeComm nd`	Utilizzare un `Procfile` o la parola chiave `scripts` in un file `package.json` per specificare lo script iniziale.
`NodeVers on`	Utilizzare la parola chiave`engines` in un file `package.json` per specificare la versione di Node.js. Tenere presente che è possibile specificare solo una versione di Node.js che corrisponde al ramo della piattaforma. Ad esempio, se si utilizza il ramo della piattaforma

Area	Modifiche e informazioni	
	Opzione	Informazioni sulla migrazione
		Node.js 12, è possibile specificare solo una versione 12.x.y Node.js. Per informazioni dettagliate, vedi the section called “Specifica delle dipendenze Node.js con un file package.json”.
	`GzipCompression`	Non supportato nelle versioni della AL2 piattaforma AL2 023/.

	`ProxyServer`	Nelle versioni della piattaforma AL2 023/ AL2 Node.js, questa opzione è stata spostata nello spazio dei nomi. `aws:elasticbeanstalk:environment:proxy` È possibile scegliere tra `nginx` (impostazione predefinita) e `apache`.
		
		AL2Le versioni della piattaforma 023/ AL2 Node.js non supportano applicazioni autonome che non funzionano dietro un server proxy. Nelle versioni della piattaforma Node.js AMI Amazon Linux, questo era possibile tramite il valore `none` dell'opzione `ProxyServer` nello spazio dei nomi `aws:elasticbeanstalk:container:nodejs`. Se l'ambiente esegue un'applicazione autonoma, aggiorna il codice per ascoltare la porta a cui li server proxy (nginx o Apache) inoltra il traffico.
		
		`var port = process.env.PORT || 5000;`
		
		`app.listen(port, function() {`
		`  console.log('Server running at http://127.0.0.1:%s', port);`
		`});`

PHP

La tabella seguente elenca le informazioni sulla migrazione per le versioni della piattaforma AL2 023/ nella AL2 piattaforma PHP.

Area	Modifiche e informazioni
Elaborazione di file PHP	Sulle AL2 piattaforme AL2 023/, i file PHP vengono elaborati utilizzando PHP-FPM (un gestore di processi CGI). Sulle piattaforme AMI Amazon Linux è stato usato mod_php (un modulo Apache).
Server proxy	AL2023/ Le versioni della piattaforma AL2 PHP supportano sia i server proxy nginx che Apache HTTPD. Il valore predefinito è nginx. Le versioni della piattaforma PHP AMI Amazon Linux supportano solo HTTPD di Apache. Se sono stati aggiunti file di configurazione Apache personalizzati, è possibile impostare l'opzione `ProxyServer` nello spazio dei nomi `aws:elasticbeanstalk:environment:proxy` su `apache`.

Python

La tabella seguente elenca le informazioni sulla migrazione per le versioni della AL2 piattaforma AL2 023/ nella piattaforma Python.

Area	Modifiche e informazioni
Server WSGI	Sulle AL2 piattaforme AL2 023/, Gunicorn è il server WSGI predefinito. Per impostazione predefinita, Gunicorn ascolta sulla porta 8000. La porta potrebbe essere diversa da quella utilizzata dall'applicazione sulla piattaforma AMI Amazon Linux. Se stai impostando l'opzione `WSGIPath` dello spazio dei nomi `aws:elasticbeanstalk:container:python`, sostituisci il valore con la sintassi di Gunicorn. Per informazioni dettagliate, vedi the section called “Spazio dei nomi di configurazione di Python”. In alternativa, puoi utilizzare `Procfile` a per specificare e configurare il server WSGI. Per informazioni dettagliate, vedi the section called “Procfile”.
Percorso dell'applicazione	Sulle AL2 piattaforme AL2 023/, il percorso della directory dell'applicazione sulle EC2 istanze Amazon del tuo ambiente è. `/var/app/current` Era `/opt/python/current/app` sulle piattaforme AMI Amazon Linux.
Server proxy	AL2023/ Le versioni della piattaforma AL2 Python supportano sia i server proxy nginx che Apache HTTPD. Il valore predefinito è nginx.

Area	Modifiche e informazioni
	Le versioni della piattaforma Python AMI Amazon Linux supportano solo HTTPD di Apache. Se sono stati aggiunti file di configurazione Apache personalizzati, è possibile impostare l'opzione `ProxyServer` nello spazio dei nomi `aws:elasticbeanstalk:environment:proxy` su `apache`.

Ruby

La tabella seguente elenca le informazioni sulla migrazione per le versioni della piattaforma AL2 023/ nella piattaforma RubyAL2 .

Area	Modifiche e informazioni
Versioni di Ruby installate	Sulle AL2 piattaforme AL2 023/, Elastic Beanstalk installa solo l'ultima versione di una singola versione di Ruby, corrispondente al ramo della piattaforma, su ciascuna versione della piattaforma. Ad esempio, ogni versione della piattaforma nel ramo della piattaforma Ruby 2.6 ha solo Ruby 2.6.x installato. Nelle versioni della piattaforma AMI Amazon Linux, vengono installate le versioni più recenti di più versioni di Ruby, ad esempio 2.4.x, 2.5.x e 2.6.x. Se l'applicazione utilizza una versione di Ruby che non corrisponde al ramo della piattaforma in uso, ti consigliamo di passare a un ramo della piattaforma con la versione di Ruby corretta per l'applicazione.
Server applicazioni	Sulle AL2 piattaforme AL2 023/, Elastic Beanstalk installa il server delle applicazioni Puma solo su tutte le versioni della piattaforma Ruby. È possibile utilizzare un `Procfile` per avviare un server delle applicazioni diverso e un `Gemfile` per installarlo. Sulla piattaforma AMI Amazon Linux, sono supportate due versioni di rami di piattaforma per ogni versione di Ruby: uno con il server delle applicazioni Puma e l'altro con il server delle applicazioni Passenger. Se l'applicazione utilizza Passenger, è possibile configurare l'ambiente Ruby per installare e utilizzare Passenger. Per maggiori informazioni ed esempi, consulta the section called “Piattaforma Ruby”.

Domande frequenti sul ritiro della piattaforma

> **Note**
>
> Elastic Beanstalk ha ritirato tutti i rami della piattaforma basati su Amazon Linux AL1 AMI () il 18 luglio 2022.

Le risposte contenute in questa domanda frequente fanno riferimento ai seguenti argomenti:

- Policy di supporto per la piattaforma Elastic Beanstalk
- Cronologia dei rami di piattaforma ritirati
- Piattaforme supportate da Elastic Beanstalk in Piattaforme AWS Elastic Beanstalk
- Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2
- Amazon Linux 2 FAQs.

1. Cosa significa il ritiro di un ramo di piattaforma?

Dopo la data di ritiro annunciata per un ramo di piattaforma, non sarai più in grado di creare un nuovo ambiente basato sul ramo di piattaforma ritirata, a meno che tu non disponga già di un ambiente attivo basato su tale ramo di piattaforma. Per ulteriori informazioni, consulta Domanda frequente n. 11. Elastic Beanstalk smetterà di offrire nuovi aggiornamenti di manutenzione per questi rami di piattaforma. Un ramo di piattaforma ritirato non è raccomandato per l'utilizzo in ambienti di produzione. Per ulteriori informazioni, consulta Domanda frequente n. 5.

2. Perché ha AWS ritirato le filiali delle piattaforme AL1 basate su piattaforme?

Elastic Beanstalk ritira i rami di piattaforma quando i componenti della piattaforma vengono deprecati o ritirati dai loro fornitori. In questo caso, l'AMI Amazon Linux (AL1) ha terminato il supporto standard il 31 dicembre 2020. Sebbene Elastic Beanstalk abbia continuato AL1 a offrire piattaforme basate fino al 2022, AL2 da allora abbiamo AL2023 rilasciato piattaforme basate sulle funzionalità più recenti. Affinché i clienti possano continuare a beneficiare della sicurezza e delle funzionalità più recenti in futuro, è fondamentale che i clienti migrino alle nostre AL2 piattaforme o basate su di esse. AL2023

3. Quali rami di piattaforma vengono ritirate?

Per un elenco dei componenti della piattaforma e dei rami della piattaforma che sono stati ritirati, consulta Cronologia dei rami di piattaforma ritirati.

4. Quali piattaforme sono attualmente supportate?

Consulta Piattaforme supportate da Elastic Beanstalk nella sezione Piattaforme AWS Elastic Beanstalk .

5. Elastic Beanstalk rimuoverà o terminerà qualsiasi componente del mio ambiente dopo il ritiro?

La nostra politica per le filiali di piattaforme ritirate non rimuove l'accesso agli ambienti né elimina le risorse. Tuttavia, un ambiente basato su un ramo di piattaforma ritirato può finire in una situazione imprevedibile poiché Elastic Beanstalk non è in grado di offrire aggiornamenti di sicurezza, supporto tecnico o aggiornamenti rapidi per rami di piattaforma ritirati poiché il fornitore contrassegna il componente come fine del ciclo di vita. Ad esempio, una vulnerabilità di sicurezza dannosa e critica potrebbe emergere in un ambiente in esecuzione su un ramo di piattaforma ritirata. Oppure un'azione API EB potrebbe smettere di funzionare per l'ambiente se nel tempo diventa incompatibile con il servizio Elastic Beanstalk. La possibilità di incorrere in questi tipi di rischi aumenta più a lungo rimane attivo un ambiente basato su un ramo di piattaforma ritirato.

Se la tua applicazione dovesse riscontrare problemi durante l'esecuzione su una filiale della piattaforma dismessa e non riesci a migrarla su una piattaforma supportata, dovrai prendere in considerazione altre alternative. Le soluzioni alternative includono l'incapsulamento dell'applicazione in un'immagine Docker per eseguirlo come container Docker. Ciò consentirebbe a un cliente di utilizzare qualsiasi delle nostre soluzioni Docker, come le nostre piattaforme AL2023 Elastic AL2 Beanstalk/Docker, o altri servizi basati su Docker come Amazon ECS o Amazon EKS. Le alternative non Docker includono il nostro AWS CodeDeploy servizio, che consente la personalizzazione completa dei runtime desiderati.

6. Posso inviare una richiesta di proroga della data di ritiro?

No. Dopo la data di ritiro, gli ambienti esistenti continueranno a funzionare. Tuttavia, Elastic Beanstalk non fornirà più aggiornamenti di manutenzione e sicurezza della piattaforma. Pertanto, è fondamentale migrare verso AL2 o AL2023 se le applicazioni sono ancora in esecuzione su una piattaforma basata su di essa. AL1 Per ulteriori informazioni su rischi e soluzioni alternative, consulta Domanda frequente n. 5.

7. Quali sono le soluzioni alternative se non riesco a completare la migrazione AL2 o AL2023 la migrazione in tempo?

I clienti possono continuare a gestire l'ambiente, anche se ti incoraggiamo a pianificare la migrazione di tutti gli ambienti Elastic Beanstalk a una versione della piattaforma supportata. In questo modo si riduce al minimo il rischio e si ottengono continui vantaggi derivanti dagli importanti miglioramenti di sicurezza, prestazioni e funzionalità offerti nelle versioni più recenti. Per ulteriori informazioni su rischi e soluzioni alternative, consulta Domanda frequente n. 5.

8. Qual è la procedura consigliata per migrare verso le nostre piattaforme? AL2 AL2023

Per istruzioni complete AL1 su AL2023 AL2 /migrazione, consultaMigrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2. Questo argomento spiega che Elastic Beanstalk blue/green richiede una distribuzione per eseguire l'aggiornamento.

9. Se ho un ambiente in esecuzione su una piattaforma ritirata, quale sarebbe l'impatto?

Un ambiente basato su un ramo di piattaforma ritirato può finire in una situazione imprevedibile poiché Elastic Beanstalk non offre aggiornamenti di sicurezza, supporto tecnico o aggiornamenti rapidi per rami della piattaforma ritirati poiché il fornitore contrassegna il componente come fine del ciclo di vita. Ad esempio, una vulnerabilità di sicurezza dannosa e critica potrebbe emergere in un ambiente in esecuzione su un ramo di piattaforma ritirata. Oppure un'azione API EB potrebbe smettere di funzionare per l'ambiente se nel tempo diventa incompatibile con il servizio Elastic Beanstalk. La possibilità di incorrere in questi tipi di rischi aumenta più a lungo rimane attivo un ambiente su una piattaforma ritirato. Per ulteriori informazioni, consulta Domanda frequente n. 5.

10. Cosa succede 90 giorni dopo la data di ritiro?

La nostra politica per le filiali di piattaforme ritirate non rimuove l'accesso agli ambienti né elimina le risorse. Tuttavia, tieni presente che un ambiente basato su un ramo di piattaforma ritirato può finire in una situazione imprevedibile poiché Elastic Beanstalk non offre aggiornamenti di sicurezza, supporto tecnico o aggiornamenti rapidi per rami di piattaforma ritirati poiché il fornitore contrassegna il componente come fine del ciclo di vita. Ad esempio, una vulnerabilità di sicurezza dannosa e critica potrebbe emergere in un ambiente in esecuzione su un ramo di piattaforma ritirata. Oppure un'azione API EB potrebbe smettere di funzionare per l'ambiente se nel tempo diventa incompatibile con il servizio Elastic Beanstalk. La possibilità di incorrere in questi tipi di rischi aumenta più a lungo rimane

attivo un ambiente su una piattaforma ritirato. Per ulteriori informazioni, consulta Domanda frequente n. 5 .

11. Posso creare un nuovo ambiente basato su una piattaforma ritirata?

È possibile creare un nuovo ambiente basato su un ramo di piattaforma ritirato, se hai già utilizzato quel ramo di piattaforma per creare un ambiente esistente utilizzando lo stesso account e nella stessa regione. Il ramo della piattaforma ritirato non sarà disponibile nella console Elastic Beanstalk. Tuttavia, per i clienti che dispongono di ambienti esistenti basati su un ramo di piattaforma ritirato, sarà disponibile tramite l'interfaccia a riga di comando EB, l'API EB e AWS CLI. Inoltre, i clienti esistenti possono utilizzare le console Clone Environment (Clona ambiente) e Rebuild environment (Ricrea ambiente). Tuttavia, tieni presente che un ambiente basato su un ramo di piattaforma ritirato può finire in una situazione imprevedibile. Per ulteriori informazioni, consulta Domanda frequente n. 5.

12. Se dispongo di un ambiente esistente in esecuzione su un ramo della piattaforma ritirato, fino a quando posso creare un nuovo ambiente basato sul ramo della piattaforma ritirato? È possibile farlo utilizzando la console, la CLI o l'API?

È possibile creare l'ambiente dopo la data di pensionamento. Tuttavia, tieni presente che un ramo di piattaforma ritirato può finire in una situazione imprevedibile. Più nel tempo un ambiente di questo tipo viene creato o attivo, maggiore è il rischio per l'ambiente di affrontare problemi imprevisti. Per ulteriori informazioni sulla creazione di un nuovo ambiente, consulta Domanda frequente n. 11.

13. Posso clonare o ricreare il mio ambiente basato sulla piattaforma ritirata?

Sì. Puoi farlo utilizzando le console Clone Environment (Clona ambiente) e Rebuild environment (Ricrea ambiente). È inoltre possibile utilizzare EB CLI, EB API e. AWS CLI Per ulteriori informazioni sulla creazione di un nuovo ambiente, consulta Domanda frequente n. 11.

Tuttavia, ti incoraggiamo a pianificare la migrazione di tutti gli ambienti Elastic Beanstalk a una versione della piattaforma supportata. In questo modo si riduce al minimo il rischio e si ottengono continui vantaggi derivanti dagli importanti miglioramenti di sicurezza, prestazioni e funzionalità offerti nelle versioni più recenti. Per ulteriori informazioni su rischi e soluzioni alternative, consulta Domanda frequente n. 5.

14. Dopo la data di pensionamento, cosa accadrebbe alle AWS risorse del mio ambiente Elastic Beanstalk basato su una filiale della piattaforma dismessa? Ad esempio, se l'istanza EC2 in esecuzione viene terminata, Elastic Beanstalk sarebbe in grado di lanciare una nuova istanza EC2 basata su AL1 per mantenere la capacità?

Le risorse dell'ambiente rimarrebbero attive e continuerebbero a funzionare. E sì, Elastic Beanstalk eseguirà il dimensionamento automatico per le istanze AL1 EC2 nell'ambiente. Tuttavia, Elastic Beanstalk smetterà di fornire nuovi aggiornamenti di manutenzione della piattaforma all'ambiente, il che può portare l'ambiente a finire in una situazione imprevedibile nel tempo. Per ulteriori informazioni, consulta Domanda frequente n. 5.

15. Quali sono le differenze principali tra i sistemi operativi AL2023/AL2 e Amazon Linux AMI (AL1)? In che modo vengono influenzati i rami AL2023 Elastic AL2 Beanstalk/piattaforma?

Sebbene Amazon Linux AMI e AL2023/AL2 condividano lo stesso kernel Linux, differiscono nel sistema di inizializzazione, nelle `libc` versioni, nella catena di strumenti del compilatore e in vari pacchetti. Per ulteriori informazioni, consulta Amazon Linux 2 FAQs.

Il servizio Elastic Beanstalk ha anche aggiornato versioni specifiche della piattaforma di runtime, strumenti di compilazione e altre dipendenze. Non è garantito che i rami della piattaforma AL2 basati su AL2023/siano retrocompatibili con l'applicazione esistente. Inoltre, anche se il codice dell'applicazione viene distribuito correttamente nella nuova versione della piattaforma, potrebbe comportarsi o funzionare in modo diverso a causa delle differenze del sistema operativo e del tempo di esecuzione. Per un elenco e una descrizione delle configurazioni e delle personalizzazioni che dovrai esaminare e testare, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Annullamento degli aggiornamenti della configurazione dell'ambiente e delle distribuzioni delle applicazioni

È possibile annullare gli aggiornamenti in corso attivati da modifiche di configurazione dell'ambiente. È anche possibile annullare la distribuzione di una nuova versione dell'applicazione in corso. Ad esempio, è possibile annullare un aggiornamento se decidi che desideri continuare a utilizzare la configurazione dell'ambiente esistente invece di applicare nuove impostazioni di configurazione dell'ambiente. Oppure, potresti renderti conto che la nuova versione dell'applicazione che stai

distribuendo ha problemi che causeranno un avvio o un'esecuzione non corretti. Annullando un aggiornamento dell'ambiente o della versione dell'applicazione, è possibile evitare di attendere finché il processo di aggiornamento o di distribuzione viene completato prima di avviare un nuovo tentativo per aggiornare l'ambiente o la versione dell'applicazione.

> **Note**
>
> Durante la fase di pulitura in cui le risorse che non sono più necessarie vengono rimosse, dopo l'aggiornamento dell'ultimo batch di istanze, non è più possibile annullare l'aggiornamento.

Elastic Beanstalk esegue il rollback nello stesso modo in cui ha eseguito l'ultimo aggiornamento di successo. Ad esempio, se hai abilitato aggiornamenti in sequenza basati sul tempo nel tuo ambiente, Elastic Beanstalk attenderà il tempo di pausa specificato tra i rollback delle modifiche in un batch di istanze prima di annullare le modifiche nel batch successivo. In alternativa, se hai attivato di recente gli aggiornamenti ricorrenti, ma l'ultima volta che le impostazioni di configurazione dell'ambiente sono state aggiornate era senza aggiornamenti in sequenza, Elastic Beanstalk eseguirà il rollback su tutte le istanze contemporaneamente.

Non è possibile impedire a Elastic Beanstalk di ripristinare la configurazione dell'ambiente precedente una volta avviato l'annullamento dell'aggiornamento. Il processo di rollback prosegue fino a quando tutte le istanze nell'ambiente hanno la configurazione dell'ambiente precedente o fino a quando il processo di rollback dà esito negativo. Per le distribuzioni delle versioni dell'applicazione, l'annullamento della distribuzione interrompe semplicemente la distribuzione; alcune istanze avranno la nuova versione dell'applicazione, mentre altre continueranno a eseguire l'attuale versione dell'applicazione. È possibile distribuire la stessa o un'altra versione dell'applicazione in un secondo momento.

Per ulteriori informazioni sugli aggiornamenti in sequenza, consulta Aggiornamenti in sequenza della configurazione dell'ambiente Elastic Beanstalk. Per ulteriori informazioni sulle distribuzione delle versioni delle applicazioni in batch, consulta Impostazioni e policy di distribuzione.

Per annullare un aggiornamento

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

3. Nella pagina Environment overview (Panoramica dell'ambiente), scegli Actions (Operazioni), quindi seleziona Abort current operation (Interrompi operazione corrente).

Ricompilazione degli ambienti Elastic Beanstalk

L' AWS Elastic Beanstalk ambiente può diventare inutilizzabile se non si utilizza la funzionalità Elastic Beanstalk per modificare o terminare le risorse sottostanti dell'ambiente. AWS In questo caso, puoi ricostruire l'ambiente per tentare di ripristinarlo a uno stato attivo. Ricreare un ambiente significa terminare tutte le risorse e sostituirle con nuove risorse con la stessa configurazione.

È anche possibile ricostruire gli ambienti terminati entro sei settimane (42 giorni) dalla loro cessazione. In fase di ricompilazione, Elastic Beanstalk tenta di creare un nuovo ambiente con nome, ID e configurazione invariati.

Ricompilazione di un ambiente in esecuzione

Puoi ricompilare un ambiente utilizzando la console Elastic Beanstalk o utilizzando l'API `RebuildEnvironment`.

> **Warning**
>
> Se l'ambiente dispone di un database accoppiato, questo verrà eliminato durante la ricostruzione e il nuovo database nell'ambiente ricostruito non conterrà i dati precedenti. Se desideri conservare il database o scattare un'istantanea, assicurati che la politica di eliminazione del database sia configurata correttamente per i risultati desiderati dopo la ricostruzione. Per ulteriori informazioni, consulta Ciclo di vita del database.

Per ricreare un ambiente in esecuzione (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Rebuild environment (Ricrea ambiente).
4. Seleziona Rebuild (Ricrea).

Per ricostruire un ambiente in esecuzione con l'API Elastic Beanstalk, `RebuildEnvironment`usa l'azione con o l'SDK. AWS CLI AWS

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Ricompilazione di un ambiente terminato

Puoi ricompilare e ripristinare un ambiente terminato utilizzando la console Elastic Beanstalk, l'interfaccia a riga di comando EB o l'API `RebuildEnvironment`.

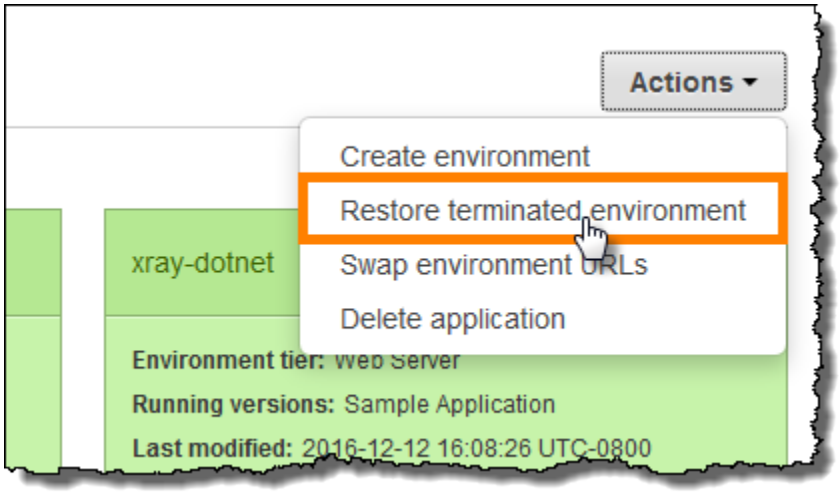
> **Note**
>
> A meno che tu non stia utilizzando un tuo nome di dominio personalizzato per l'ambiente terminato, l'ambiente utilizza un sottodominio di elasticbeanstalk.com. Questi sottodomini sono condivisi all'interno di una regione Elastic Beanstalk. Pertanto, possono essere utilizzati da qualsiasi ambiente creato da qualsiasi cliente nella stessa regione. Sebbene il tuo ambiente sia stato terminato, un altro ambiente potrebbe utilizzare il suo sottodominio. In questo caso, la ricreazione non andrebbe a buon fine.
> È possibile evitare questo problema utilizzando un dominio personalizzato. Per informazioni dettagliate, vedi Nome di dominio dell'ambiente Elastic Beanstalk.

Gli ambienti terminati di recente appaiono nella panoramica delle applicazioni per massimo un'ora. Durante tale periodo, puoi visualizzare gli eventi dell'ambiente nel pannello di controllo e utilizzare l'operazione Restore environment (Ripristina ambiente)??? per ricrearlo.

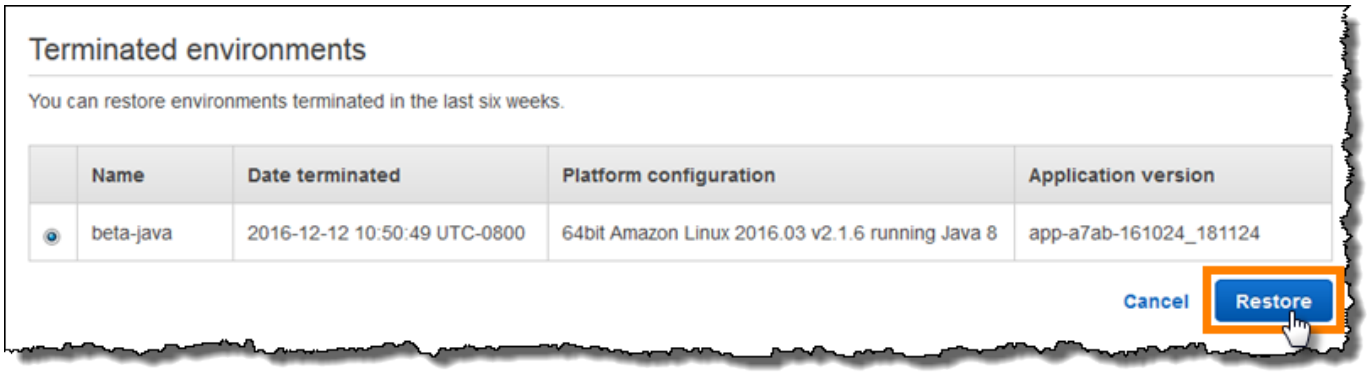
Per ricreare un ambiente che non è più visibile, utilizza l'opzione Restore terminated environment (Ripristina ambiente terminato) dalla pagina delle applicazioni.

Per ricreare un ambiente terminato (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Seleziona Actions (Operazioni), quindi Restore terminated environment (Ripristina ambiente terminato).



4. Seleziona un ambiente terminato.
5. Scegli Restore (Ripristina).



Elastic Beanstalk tenta di creare un nuovo ambiente con nome, ID e configurazione invariati. Se un ambiente con lo stesso nome o un URL esiste già quando tenti di ricostruirlo, la ricostruzione ha esito negativo. Anche l'eliminazione della versione dell'applicazione distribuita nell'ambiente porterà al fallimento della ricostruzione.

Se utilizzi l'interfaccia a riga di comando EB per la gestione dell'ambiente, usa il comando eb restore per ricostruire un ambiente terminato.

```
$ eb restore e-vdnftxubwq
```

Per ulteriori informazioni, consulta eb restore.

Per ricostruire un ambiente terminato con l'API Elastic Beanstalk, usa `RebuildEnvironment`l'azione con o l'SDK. AWS CLI AWS

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Tipi di ambiente

Nel AWS Elastic Beanstalk, puoi creare un ambiente scalabile con bilanciamento del carico o un ambiente a istanza singola. Il tipo di ambiente che richiedi dipende dall'applicazione che distribuisci. Ad esempio, è possibile sviluppare e testare un'applicazione in un ambiente a singola istanza per risparmiare sui costi, quindi effettuare l'aggiornamento di quell'ambiente a un ambiente con bilanciamento del carico e scalabile quando l'applicazione è pronta per la produzione.

> **Note**
>
> Il livello dell'ambiente lavoratore per un'applicazione Web che elabora attività in background non include un sistema di bilanciamento del carico. Tuttavia, un ambiente worker (operatore) viene dimensionato in modo efficace aggiungendo le istanze al gruppo Auto Scaling per elaborare i dati dalla coda Amazon SQS quando il carico lo richiede.

Ambiente scalabile e con bilanciamento del carico

Un ambiente scalabile e con bilanciamento del carico utilizza i servizi Elastic Load Balancing e Amazon Auto EC2 Scaling per fornire le EC2 istanze Amazon necessarie per l'applicazione distribuita. Amazon EC2 Auto Scaling avvia automaticamente istanze aggiuntive per far fronte all'aumento del carico sull'applicazione. Se il carico sull'applicazione diminuisce, Amazon EC2 Auto Scaling interrompe le istanze ma lascia sempre in esecuzione il numero minimo di istanze specificato. Se l'applicazione richiede la scalabilità con l'opzione di esecuzione in più zone di disponibilità, utilizza un ambiente di bilanciamento del carico e scalabile. Se non sei certo del tipo di ambiente da selezionare, è possibile selezionarne uno e, se necessario, cambiarlo in un secondo momento.

Ambiente di istanza singola

Un ambiente a istanza singola contiene un' EC2 istanza Amazon con un indirizzo IP elastico. Un ambiente a singola istanza non dispone di un sistema di bilanciamento del carico, il che può aiutare a ridurre i costi rispetto a un ambiente con bilanciamento del carico e scalabile. Sebbene un ambiente a istanza singola utilizzi il servizio Amazon EC2 Auto Scaling, le impostazioni per il numero minimo di istanze, il numero massimo di istanze e la capacità desiderata sono tutte impostate

su 1. Di conseguenza, le nuove istanze non vengono avviate per supportare l'aumento del carico dell'applicazione.

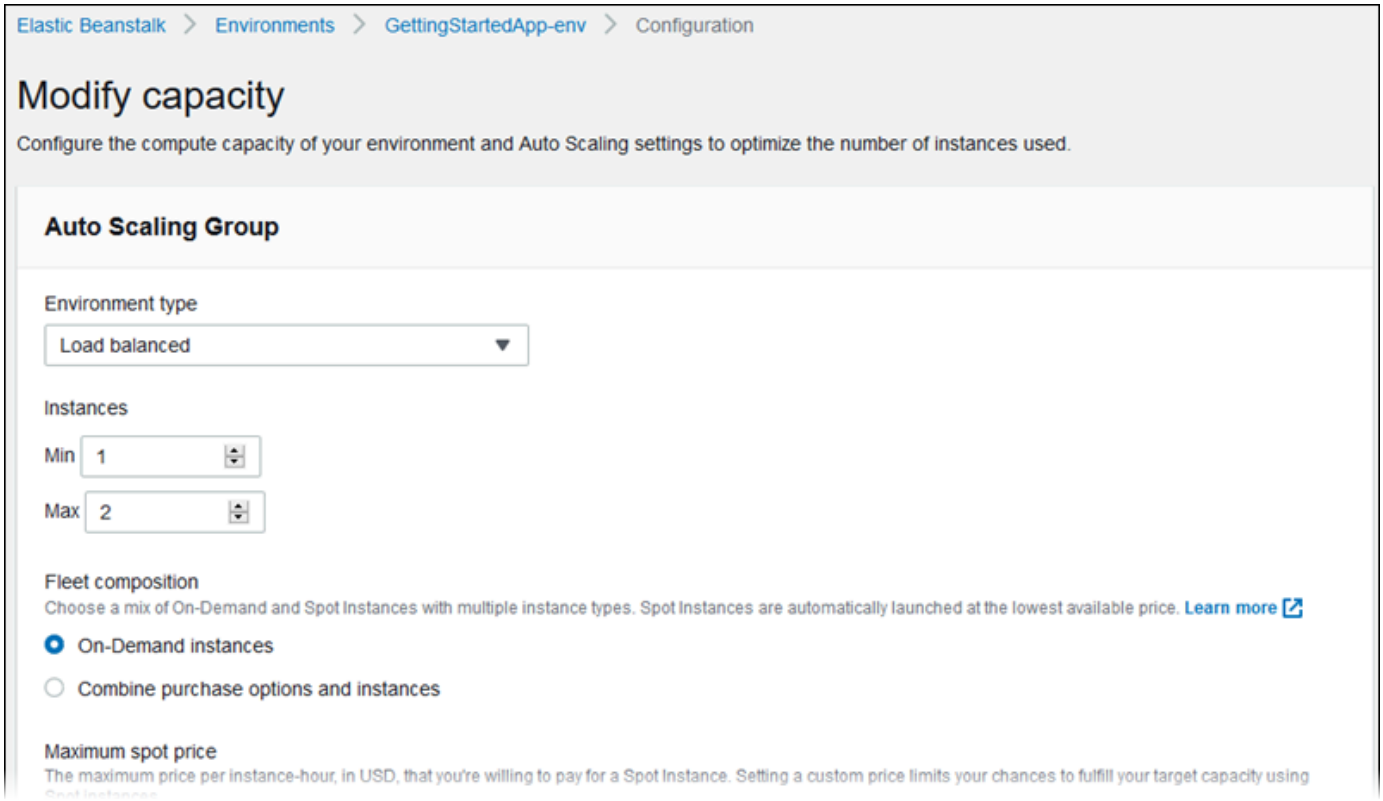
Utilizza un ambiente di istanza singola se prevedi che l'applicazione in produzione abbia un traffico ridotto o se stai eseguendo lo sviluppo in remoto. Se non sei certo del tipo di ambiente da selezionare, è possibile selezionarne uno e, se necessario, cambiarlo in un secondo momento. Per ulteriori informazioni, consulta Modifica del tipo di ambiente.

Modifica del tipo di ambiente

È possibile modificare il tipo di ambiente per sceglierne uno a istanza singola o un ambiente con bilanciamento del carico e scalabile modificando la configurazione dell'ambiente. In alcuni casi, può essere opportuno cambiare il tipo di ambiente passando da un tipo a un altro. Ad esempio, supponiamo che tu abbia sviluppato e testato un'applicazione in un ambiente a istanza singola per ridurre i costi. Quando l'applicazione è pronta per la produzione, puoi modificare il tipo di ambiente con un ambiente con bilanciamento del carico e scalabile in modo che sia possibile soddisfare le richieste dei clienti.

Per modificare il tipo di ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria Capacity (Capacità), scegliere Edit (Modifica).
5. Dall'elenco Environment Type (Tipo di ambiente), selezionare il tipo di ambiente che desideri.



6. Scegli Save (Salva).

   L'aggiornamento dell'ambiente può richiedere diversi minuti mentre Elastic Beanstalk effettua il provisioning delle risorse AWS .

Se il tuo ambiente è in un VPC, seleziona le sottoreti in cui inserire le istanze Elastic Load Balancing e Amazon. EC2 Ogni zona di disponibilità in cui viene eseguita la tua applicazione deve avere entrambi. Per informazioni dettagliate, vedi Utilizzo di Elastic Beanstalk con Amazon VPC.

Ambienti worker Elastic Beanstalk

Se l' AWS Elastic Beanstalk applicazione esegue operazioni o flussi di lavoro che richiedono molto tempo per essere completati, è possibile trasferire tali attività in un ambiente di lavoro dedicato. Il disaccoppiamento del front-end della tua applicazione Web da un processo che esegue le operazioni di blocco è un modo comune per assicurare che la tua applicazione rimanga reattiva sotto carico.

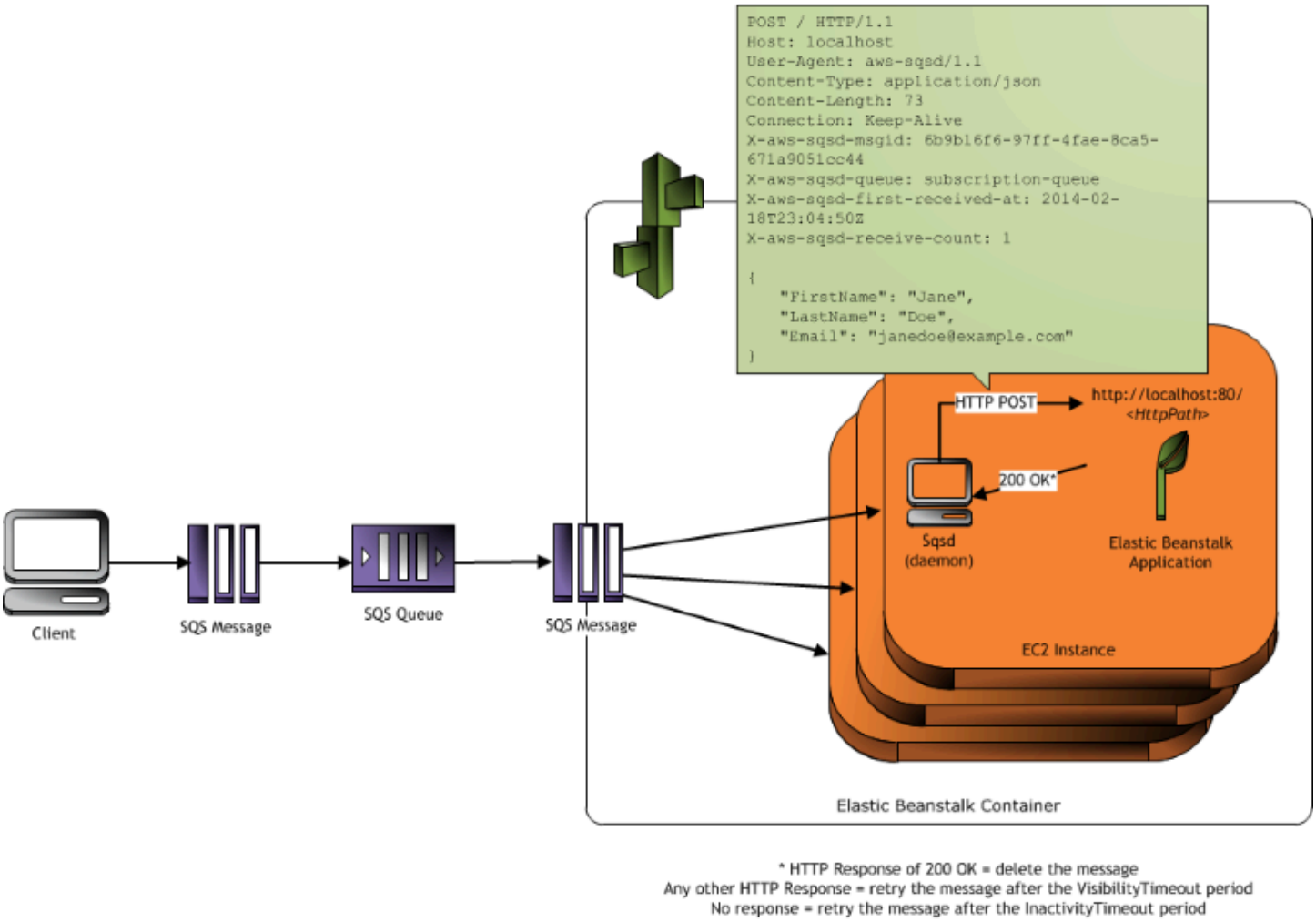
Un'attività con esecuzione lunga è tutto ciò che sostanzialmente aumenta il tempo necessario per completare una richiesta, ad esempio l'elaborazione di immagini o video, l'invio di e-mail oppure la generazione di un archivio ZIP. Queste operazioni possono richiedere solo un secondo o due per

essere completate, ma alcuni secondi di ritardo è molto per una richiesta Web che altrimenti verrebbe completata in meno di 500 millisecondi.

Un'opzione è generare un processo lavoratore in locale, restituire una risposta positiva ed elaborare le attività in modo asincrono. Questo funziona se la tua istanza è in grado di eseguire tutte le operazioni inviate. In presenza di un carico elevato, tuttavia, un'istanza può essere sovraccaricata da attività in background e non rispondere alle richieste di priorità superiore. Se i singoli utenti sono in grado di generare più attività, l'aumento del carico potrebbe non corrispondere a un aumento degli utenti, rendendo più complesso un ridimensionamento efficiente a livello di server Web.

Per evitare di eseguire attività di lunga durata a livello locale, puoi utilizzare l' AWS SDK per il tuo linguaggio di programmazione per inviarle a una coda Amazon Simple Queue Service (Amazon SQS) ed eseguire il processo che le esegue su un set separato di istanze. Le istanze lavoratore vengono quindi progettate per richiedere solo gli elementi dalla coda quando hanno la capacità di eseguirli, evitando così di sovraccaricarle.

Gli ambienti worker Elastic Beanstalk semplificano questo processo gestendo la coda Amazon SQS ed eseguendo un processo daemon su ciascuna istanza che legge dalla coda automaticamente. Quando il daemon preleva un elemento dalla coda, invia una richiesta HTTP POST in locale per `http://localhost/` sulla porta 80 con i contenuti del messaggio della coda nel corpo. Tutto quello di cui ha bisogno l'applicazione è eseguire l'attività di lunga esecuzione in risposta al POST. È possibile configurare il daemon per pubblicare su un altro percorso, utilizzare un tipo MIME diverso dall'applicazione/JSON, connettersi a una coda esistente o personalizzare le connessioni (numero massimo di richieste simultanee), il timeout e i tentativi.



Con le attività periodiche, è anche possibile configurare il daemon lavoratore per accodare i messaggi in base a una pianificazione cron Ogni attività periodica può essere resa disponibile su un percorso diverso. Abilita le attività periodiche includendo un file YAML nel tuo codice sorgente che definisce la pianificazione e il percorso per ogni attività.

> **Note**
>
> .NET sulla piattaforma Windows Server non supporta gli ambienti lavoratore.

Sections

- Daemon SQS dell'ambiente lavoratore
- Code di lettere morte
- Attività periodiche

- Usa Amazon CloudWatch per la scalabilità automatica nei livelli di ambiente di lavoro
- Configurazione degli ambienti lavoratore

Daemon SQS dell'ambiente lavoratore

Gli ambienti worker eseguono un processo daemon fornito da Elastic Beanstalk. Questo daemon è aggiornato regolarmente per aggiungere funzionalità e risolvere i bug. Per ottenere la versione più recente del daemon, esegui l'aggiornamento all'ultima versione della piattaforma.

Quando l'applicazione nell'ambiente worker restituisce una risposta `200 OK` per dichiarare che ha ricevuto ed elaborato correttamente la richiesta, il daemon invia una chiamata `DeleteMessage` alla coda Amazon SQS per eliminare il messaggio dalla coda. Se l'applicazione restituisce una risposta diversa da `200 OK`, prima di reinserire il messaggio nella coda Elastic Beanstalk attende che termini l'intervallo `ErrorVisibilityTimeout` configurato. Se non c'è risposta, prima di reinserire il messaggio nella coda Elastic Beanstalk attende che l'intervallo `InactivityTimeout` termini, in modo che il messaggio sia disponibile per un altro tentativo di elaborazione.

> **Note**
>
> Le proprietà delle code Amazon SQS (ordine dei messaggi, at-least-once consegna e campionamento dei messaggi) possono influire sul modo in cui si progetta un'applicazione Web per un ambiente di lavoro. Per maggiori informazioni, consulta l'argomento relativo alle proprietà delle code distribuite nella Guida per gli sviluppatori di Amazon Simple Queue Service.

Amazon SQS elimina automaticamente i messaggi che sono stati in una coda per un periodo superiore al `RetentionPeriod` configurato.

Il daemon imposta le seguenti intestazioni HTTP.

Intestazioni HTTP

Nome	Valore
`User-Agent`	`aws-sqsd`
	`aws-sqsd/1.1 1`

Intestazioni HTTP	
`X-Aws-Sqsd-Msgid`	ID messaggio SQS, utilizzato per rilevare le tempeste di messaggi (un numero insolitamente elevato di nuovi messaggi).
`X-Aws-Sqsd-Queue`	Nome della coda SQS.
`X-Aws-Sqsd-First-Received-At`	Orario UTC, in formato ISO 8601, quando il messaggio è stato ricevuto.
`X-Aws-Sqsd-Receive-Count`	Conteggio messaggi SQS ricevuti.
`X-Aws-Sqsd-Attr-` *`message-attribute-name`*	Attributi dei messaggi personalizzati assegnati al messaggio in fase di elaborazione. Il messaggio `message-attribute-name` è il nome dell'attributo effettivo del messaggio. Tutti gli attributi di stringa e di messaggio numero vengono aggiunti all'intestazione. Gli attributi binari verranno eliminati e non saranno inclusi nell'intestazione.
`Content-Type`	Configurazione di tipo mime; per impostazione predefinita, `application/json` .

Code di lettere morte

Gli ambienti worker Elastic Beanstalk supportano le code DLQ di Amazon Simple Queue Service (Amazon SQS). Una coda DLQ è una coda a cui altre code (origini) possono inviare messaggi che, per qualche motivo, non sono stati elaborati correttamente. Uno dei principali vantaggi dell'utilizzo di una coda DLQ è la possibilità di mettere da parte e isolare i messaggi non elaborati in modo corretto. È quindi possibile analizzare i messaggi inviati alla coda DLQ per cercare di stabilire il motivo per cui non sono stati elaborati correttamente.

Per impostazione predefinita, una coda DLQ viene abilitata per un ambiente worker se specifichi una coda Amazon SQS generata automaticamente al momento della creazione di un livello dell'ambiente worker. Se scegli una coda SQS esistente per il tuo ambiente lavoratore, è necessario utilizzare SQS

per configurare una coda DLQ in modo indipendente. Per informazioni su come utilizzare SQS per configurare una coda DLQ, consulta Using Amazon SQS Dead Letter Queues.

Non è possibile disabilitare le code DLQ. I messaggi che non possono essere distribuiti vengono infine inviati a una coda DLQ. È possibile, tuttavia, disattivare questa funzione in modo efficiente impostando l'opzione `MaxRetries` sul massimo valore valido di 100.

Se una coda DLQ non è configurata per la coda Amazon SQS dell'ambiente worker, Amazon SQS mantiene i messaggi nella coda fino alla scadenza del periodo di conservazione. Per informazioni dettagliate sulla configurazione del periodo di conservazione, consulta the section called "Configurazione degli ambienti lavoratore".

> **Note**
>
> L'opzione `MaxRetries` di Elastic Beanstalk equivale all'opzione `MaxReceiveCount` di SQS. Se l'ambiente lavoratore non utilizza una coda SQS autogenerata, utilizza l'opzione `MaxReceiveCount` in SQS per disabilitare correttamente la coda DLQ. Per ulteriori informazioni, consulta Using Amazon SQS Dead Letter Queues.

Per ulteriori informazioni sul ciclo di vita di un messaggio SQS, consulta l'argomento relativo al ciclo di vita dei messaggi.

Attività periodiche

È possibile definire le attività periodiche in un file denominato `cron.yaml` nel tuo bundle di origine per aggiungere automaticamente processi alla coda dell'ambiente lavoratore a intervalli regolari.

Ad esempio, il file `cron.yaml` seguente crea due attività periodiche. La prima viene eseguita ogni 12 ore, mentre la seconda tutti i giorni alle 23.00.

Example cron.yaml

```
version: 1
cron:
 - name: "backup-job"
   url: "/backup"
   schedule: "0 */12 * * *"
 - name: "audit"
   url: "/audit"
```

```
  schedule: "0 23 * * *"
```

Il **name** deve essere univoco per ogni attività. L'URL è il percorso su cui viene inviata la richiesta POST per attivare l'attività. La pianificazione è un'espressione CRON che stabilisce quando viene eseguita l'operazione.

Quando viene eseguita un'attività, il daemon pubblica un messaggio nella coda SQS dell'ambiente con un'intestazione che indica l'attività che deve essere eseguita. Ogni istanza dell'ambiente può prendere il messaggio ed elaborare l'attività.

> **Note**
>
> Se configuri l'ambiente worker con una coda SQS esistente e scegli una coda FIFO Amazon SQS, le attività periodiche non sono supportate.

Elastic Beanstalk impiega l'elezione dei leader per stabilire quale istanza nell'ambiente worker sposta in coda l'attività periodica. Ogni istanza tenta di diventare leader scrivendo su una tabella Amazon DynamoDB. La prima istanza che va a buon fine è quella leader che deve continuare a scrivere sulla tabella per mantenere lo stato di leader. Se l'istanza leader è inutilizzabile, un'altra istanza prende rapidamente il suo posto.

Per le attività periodiche, il daemon lavoratore imposta le seguenti intestazioni aggiuntive.

Intestazioni HTTP

Nome	Valore
`X-Aws-Sqsd-Taskname`	Per le attività periodiche, il nome dell'attività da eseguire.
`X-Aws-Sqsd-Scheduled-At`	L'orario in cui è stata programmata l'attività periodica
`X-Aws-Sqsd-Sender-Id`	AWS numero di conto del mittente del messaggio

Usa Amazon CloudWatch per la scalabilità automatica nei livelli di ambiente di lavoro

Insieme, Amazon EC2 Auto CloudWatch Scaling monitora l'utilizzo della CPU delle istanze in esecuzione nell'ambiente di lavoro. Il modo in cui configuri il limite della scalabilità automatica per capacità di CPU determina il numero di istanze che il gruppo Auto Scaling esegue per gestire correttamente il throughput di messaggi nella coda Amazon SQS. Ogni istanza EC2 pubblica i propri parametri di utilizzo della CPU su. CloudWatch Amazon EC2 Auto Scaling CloudWatch recupera l'utilizzo medio della CPU in tutte le istanze dell'ambiente di lavoro. È possibile configurare il limite superiore e inferiore e il numero di istanze da aggiungere o terminare in base alla capacità della CPU. Quando Amazon EC2 Auto Scaling rileva che è stato raggiunto il limite massimo specificato per la capacità della CPU, Elastic Beanstalk crea nuove istanze nell'ambiente worker. Le istanze vengono eliminate quando il carico della CPU scende al di sotto della soglia.

> **Note**
>
> I messaggi che non sono stati elaborati nel momento in cui un'istanza viene terminata vengono restituiti alla coda dove possono essere elaborati da un altro daemon su un'istanza che è ancora in esecuzione.

Puoi anche impostare altri CloudWatch allarmi, se necessario, utilizzando la console Elastic Beanstalk, la CLI o il file delle opzioni. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon CloudWatch e l'argomento relativo alla creazione di un gruppo Auto Scaling con policy di dimensionamento di fasi.

Configurazione degli ambienti lavoratore

È possibile gestire la configurazione di un ambiente worker modificando la categoria Worker nella pagina Configuration (Configurazione) della console di gestione dell'ambiente.

Elastic Beanstalk > Environments > GettingStartedApp-env > Configuration

Modify worker

You can create a new Amazon SQS queue for your worker application or pull work items from an existing queue. The worker daemon on the instances in your environment pulls an item from the queue and relays it in the body of a POST request to a local HTTP path relative to localhost.

Queue

Worker queue

Autogenerated queue

SQS queue from which to read work items.

Messages

HTTP path

/

The daemon pulls items from the Amazon SQS queue and posts them locally to this path.

MIME type

application/json

Change the MIME type of the POST requests that the worker daemon sends to your application.

HTTP connections

50

Maximum number of concurrent connections to the application.

Visibility timeout

300 seconds

The amount of time to lock an incoming message for processing before returning it to the queue.

Error visibility timeout

seconds

The amount of time to wait before resending a message after an error response from the application.

Advanced options

The following settings control advanced behavior of the worker tier daemon. Learn more

Max retries

10

Maximum number of retries after which the message is discarded.

Connection timeout

5

Inactivity timeout

> **Note**
>
> È possibile configurare il percorso URL per la pubblicazione di messaggi della coda di lavoro, ma non è possibile configurare la porta IP. Elastic Beanstalk pubblica sempre i messaggi della coda operatore sulla porta 80. L'applicazione dell'ambiente di lavoro o il relativo proxy devono essere in ascolto sulla porta 80.

Per configurare il daemon lavoratore

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Worker scegliere Edit (Modifica).

La pagina di configurazione Modify worker (Modifica worker) presenta le opzioni seguenti.

Nella sezione Queue (Coda):

- Worker queue (Coda operatore): specifica la coda Amazon SQS da cui legge il daemon. Se disponibile, puoi scegliere una cosa esistente. Se scegli Autogenerated queue (Coda generata automaticamente), Elastic Beanstalk crea una nuova coda Amazon SQS e un Worker queue URL (URL coda operatore) corrispondente.

  > **Note**
  >
  > Quando scegli Autogenerated queue (Coda generata automaticamente), la coda creata da Elastic Beanstalk è una coda standard Amazon SQS. Quando scegli una coda esistente, puoi fornire una coda standard o FIFO Amazon SQS. Se si fornisce una coda FIFO, le attività periodiche non sono supportate.

- Worker queue URL (URL coda operatore): se scegli una Worker queue (Coda operatore) esistente, questa impostazione visualizza l'URL associato a quella coda Amazon SQS.

Nella sezione Messages (Messaggi):

- HTTP path (Percorso HTTP): specifica il percorso relativo dell'applicazione che riceverà i dati dalla coda Amazon SQS. I dati vengono inseriti nel corpo di un messaggio POST HTTP. Il valore predefinito è `/`.
- MIME type (Tipo MIME): indica il tipo MIME impiegato dal messaggio HTTP POST. Il valore predefinito è `application/json`. Tuttavia, qualsiasi valore è valido perché è possibile creare e quindi specificare un tipo MIME proprio.
- HTTP connections (Connessioni HTTP): specifica il numero massimo di connessioni che il daemon può effettuare simultaneamente per qualsiasi applicazione in un'istanza Amazon EC2. Il valore predefinito è **`50`**. Puoi specificare un valore da **`1`** a **`100`**.
- Visibility timeout (Timeout visibilità): il periodo di tempo espresso in secondi durante il quale un messaggio in ingresso da una coda Amazon SQS è bloccato per l'elaborazione. Trascorso il periodo di tempo impostato, il messaggio è di nuovo visibile nella coda per la lettura da parte di un altro daemon. Scegli un valore di tempo più lungo rispetto a quello che prevedi la tua applicazione utilizzerà per elaborare messaggi, fino a **`43200`** secondi.
- Error visibility timeout (Timeout visibilità errore): il periodo di tempo espresso in secondi che decorre prima che Elastic Beanstalk restituisca un messaggio alla coda Amazon SQS dopo un tentativo di elaborazione non riuscito con errore esplicito. Puoi specificare un valore da **`0`** a **`43200`** secondi.

Nella sezione Opzioni avanzate:

- Max retries (Numero massimo di tentativi): specifica il numero massimo di volte che Elastic Beanstalk tenta di inviare il messaggio alla coda Amazon SQS prima di spostare il messaggio alla coda DLQ. Il valore predefinito è **`10`**. Puoi specificare un valore da **`1`** a **`100`**.

  > **Note**
  >
  > L'opzione Max retries (Numero massimo di tentativi) si applica solo alle code Amazon SQS configurate con una coda DLQ. Per tutte le code Amazon SQS che non sono configurate con una coda DLQ, Amazon SQS conserva i messaggi nella coda e li elabora fino al raggiungimento del periodo specificato dall'opzione Retention period (Periodo di conservazione).

- Connection timeout (Timeout connessione): il periodo di attesa espresso in secondi delle connessioni con esito positivo a un'applicazione. Il valore predefinito è **`5`**. Puoi specificare un valore da **`1`** a **`60`** secondi.

- Inactivity timeout (Timeout inattività): il periodo di attesa espresso in secondi della risposta di una connessione esistente a un'applicazione. Il valore predefinito è **`180`**. Puoi specificare un valore da **`1`** a **`36000`** secondi.
- Retention period (Periodo di conservazione): il periodo di tempo espresso in secondi durante il quale un messaggio è valido e viene elaborato attivamente. Il valore predefinito è **`345600`**. Puoi specificare un valore da **`60`** a **`1209600`** secondi.

Se utilizzi una coda Amazon SQS esistente, le impostazioni configurate al momento della creazione di un ambiente worker possono entrare in conflitto con le impostazioni configurate direttamente in Amazon SQS. Ad esempio, se configuri un ambiente worker con un valore `RetentionPeriod` superiore al valore `MessageRetentionPeriod` impostato in Amazon SQS, Amazon SQS elimina il messaggio quando supera il `MessageRetentionPeriod`.

Al contrario, se il valore `RetentionPeriod` configurato nelle impostazioni dell'ambiente worker è inferiore al valore `MessageRetentionPeriod` impostato in Amazon SQS, il daemon elimina il messaggio prima di Amazon SQS. Per `VisibilityTimeout`, il valore che viene configurato per il daemon nelle impostazioni dell'ambiente worker ignora l'impostazione `VisibilityTimeout` di Amazon SQS. Assicurati che i messaggi vengano eliminati correttamente confrontando le impostazioni di Elastic Beanstalk con le impostazioni di Amazon SQS.

Creazione di collegamenti tra gli ambienti Elastic Beanstalk

Poiché la tua applicazione cresce in termini di dimensioni e complessità, è possibile che tu voglia suddividerla in componenti che hanno cicli di vita di sviluppo e operativi differenti. Eseguendo servizi più piccoli che interagiscono tra loro mediante un'interfaccia ben definita, i team possono lavorare in modo indipendente e le distribuzioni possono presentare un rischio più ridotto. AWS Elastic Beanstalk consente di collegare gli ambienti per condividere le informazioni tra i componenti interdipendenti.

> **Note**
>
> Elastic Beanstalk supporta attualmente collegamenti tra ambienti per tutte le piattaforme ad eccezione di Multicontainer Docker.

Mediante i collegamenti tra ambienti, puoi specificare le connessioni tra gli ambienti dei componenti dell'applicazione come riferimenti designati. Quando crei un ambiente che definisce un collegamento,

Elastic Beanstalk imposta una variabile di ambiente con lo stesso nome del collegamento. Il valore della variabile è l'endpoint che puoi utilizzare per la connessione a un altro componente, ovvero un server Web o un ambiente lavoratore.

Ad esempio, se la tua applicazione è composta da un front-end che raccoglie indirizzi e-mail e un lavoratore che invia un messaggio di benvenuto agli indirizzi e-mail raccolti dal front-end, puoi creare un collegamento al lavoratore nel tuo front-end di modo che rilevi automaticamente l'endpoint (URL coda) per il lavoratore.

Definisci i collegamenti ad altri ambienti in un manifest di ambiente, un file in formato YAML denominato `env.yaml` nella radice dell'origine dell'applicazione. Il seguente manifest definisce un collegamento a un ambiente denominato lavoratore:

`~/workspace/my-app/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentLinks:
  "WORKERQUEUE": "worker"
```

Quando crei un ambiente con una versione dell'applicazione che include il suddetto manifest di ambiente, Elastic Beanstalk cerca un ambiente denominato `worker` appartenente alla stessa applicazione. Se tale ambiente esiste, Elastic Beanstalk crea una proprietà di ambiente denominata `WORKERQUEUE`. Il valore di `WORKERQUEUE` è l'URL coda di Amazon SQS. L'applicazione front-end può leggere questa proprietà come variabile di ambiente. Per informazioni dettagliate, vedi Manifest di ambiente (`env.yaml`).

Per utilizzare i collegamenti all'ambiente, aggiungi un manifesto di ambiente all'origine dell'applicazione e caricalo con l'EB CLI AWS CLI o un SDK. Se utilizzi AWS CLI o un SDK, imposta il `process` flag quando chiami: `CreateApplicationVersion`

```
$ aws elasticbeanstalk create-application-version --process --application-name
 my-app --version-label frontend-v1 --source-bundle S3Bucket="amzn-s3-demo-
bucket",S3Key="front-v1.zip"
```

Questa opzione indica a Elastic Beanstalk di convalidare il manifest di ambiente e i file di configurazione nel bundle di origine quando crei la versione dell'applicazione. L'interfaccia a riga di comando EB imposta questo flag automaticamente quando un manifest di ambiente è presente nella directory di progetto.

Crea i tuoi ambienti normalmente utilizzando qualsiasi client. Quando devi terminare gli ambienti, esegui questa operazione dapprima con il collegamento. Se un ambiente è collegato a un altro ambiente, Elastic Beanstalk impedisce la terminazione dell'ambiente collegato. Per sovrascrivere questa protezione, utilizza il flag `ForceTerminate`. Questo parametro è disponibile nell' AWS CLI come `--force-terminate`:

```
$ aws elasticbeanstalk terminate-environment --force-terminate --environment-name
 worker
```

Ripristino dell'ambiente Elastic Beanstalk da uno stato non valido

Questo argomento fornisce alcune informazioni e risorse di base che spiegano come risolvere i problemi di un ambiente Elastic Beanstalk in uno stato non valido.

Risolvere l'errore

Le operazioni standard su un ambiente in uno stato non valido non verranno completate correttamente. L'operazione non riuscita restituirà un errore che include il testo seguente:

```
The stack stack_id associated with environment environment-ID is in stack-status state.
```

Per risolvere questo errore, consulta l'articolo del Knowledge Center Perché il mio ambiente Elastic Beanstalk è nello stato non valido? .

> **Note**
>
> Prima del 16 dicembre 2024, l'operazione non riuscita restituiva invece il seguente errore: `Environment is in an invalid state for this operation. Must be ready.` in questo caso era necessario contattare l' AWS assistenza per ripristinare lo stato dell'ambiente dopo aver completato le azioni correttive.
> Oggi è ancora necessario risolvere i problemi relativi allo stack seguendo le istruzioni contenute nell'articolo di riferimento del Knowledge Center. Tuttavia, una volta completate con successo le azioni correttive, Elastic Beanstalk aggiorna automaticamente lo stato dell'ambiente da non valido a disponibile e puoi riprendere le operazioni standard sull'ambiente senza ulteriori ritardi.

Perché si verifica l'errore

Quando distribuisci un'applicazione in Elastic Beanstalk, il servizio crea uno stack sottostante. AWS CloudFormation Elastic Beanstalk CloudFormation chiama il servizio per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione.

Se Elastic Beanstalk esegue un'operazione su un ambiente senza avere accesso a una risorsa richiesta, lo stack sottostante dell'ambiente può entrare in uno stato CloudFormation di errore. Anche altri problemi possono portare a questo stato, sebbene i problemi di autorizzazione siano la causa principale. A causa dello stato di errore dello stack, CloudFormation impedisce alle richieste operative di Elastic Beanstalk di eseguire ulteriori aggiornamenti dello stack, causando il fallimento delle operazioni di Elastic Beanstalk, come e. UpdateEnvironment RetrieveEnvironmentInfo

A questo punto è necessario innanzitutto correggere la causa principale del problema sottostante per porre rimedio allo stack. CloudFormation Il servizio Elastic Beanstalk CloudFormation rileva quindi la modifica dello stato dello stack e procede al ripristino dello stato disponibile dell'ambiente. A questo punto, ulteriori operazioni possono essere completate con successo.

I problemi di autorizzazione in genere causano questo effetto sullo CloudFormation stack e sull'ambiente Elastic Beanstalk out-of-band, sebbene anche le modifiche possano causare problemi.

> **Important**
>
> Per evitare interruzioni dell'ambiente, consigliamo vivamente di avviare le operazioni di gestione e configurazione dell'ambiente solo dal servizio Elastic Beanstalk. La modifica delle risorse utilizzando la console, i comandi CLI o l'SDK di un servizio diverso da Elastic Beanstalk è una modifica che causa la deriva delle risorse. out-of-band La deriva delle risorse influisce sullo stato dello CloudFormation stack, il che a sua volta fa sì che l'ambiente Elastic Beanstalk entri in uno stato non valido.
> Per ulteriori informazioni sulla deriva delle risorse, vedi Cos'è la deriva? nella Guida per l'AWS CloudFormation utente.

Configurazione degli ambienti Elastic Beanstalk

Questo argomento si concentra sulle opzioni di configurazione disponibili nella console Elastic Beanstalk. AWS Elastic Beanstalk offre un'ampia gamma di opzioni per personalizzare le risorse dell'ambiente, oltre al comportamento di Elastic Beanstalk e alle impostazioni della piattaforma.

I seguenti argomenti mostrano come configurare l'ambiente nella console. Inoltre, descrive gli spazi dei nomi sottostanti che corrispondono alle opzioni della console da utilizzare con i file di configurazione o con le opzioni di configurazione delle API. Per informazioni sui metodi di configurazione avanzati, consulta Configurazione degli ambienti (avanzata).

Argomenti

- Risorse fornite
- Configurazione dell'ambiente utilizzando la console Elastic Beanstalk
- Le EC2 istanze Amazon per il tuo ambiente Elastic Beanstalk
- Auto Scaling delle istanze dell'ambiente Elastic Beanstalk
- Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk
- Aggiunta di un database all'ambiente Elastic Beanstalk
- Tuo AWS Elastic Beanstalk sicurezza dell'ambiente
- Tagging delle risorse negli ambienti Elastic Beanstalk
- Variabili di ambiente e altre impostazioni software
- Notifiche dell'ambiente Elastic Beanstalk con Amazon SNS
- Configurazione di Amazon Virtual Private Cloud (Amazon VPC) con Elastic Beanstalk
- Nome di dominio dell'ambiente Elastic Beanstalk

Risorse fornite

Quando crei un ambiente server Web, Elastic Beanstalk crea più risorse per supportare il funzionamento dell'applicazione. Questo capitolo descrive come personalizzare queste risorse per l'ambiente Elastic Beanstalk.

- Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.

- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).

Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Configurazione dell'ambiente utilizzando la console Elastic Beanstalk

Questo argomento descrive le opzioni di configurazione disponibili tramite la console Elastic Beanstalk e spiega come navigare nelle pagine di configurazione.

Per visualizzare un riepilogo della configurazione dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).

Pagina di configurazione

La pagina Configuration overview (Panoramica di configurazione) mostra un gruppo di categorie di configurazione. Ogni categoria di configurazione raggruppa una serie di opzioni correlate.

Accesso al servizio

Le opzioni di questa categoria selezionano il ruolo di servizio e il profilo dell'istanza EC2 che Elastic Beanstalk utilizza per gestire l'ambiente. Opzionalmente, scegli una coppia di key pair EC2 per accedere in modo sicuro alle tue istanze EC2.

Rete e database

Le opzioni di questa categoria configurano le impostazioni VPC e le sottoreti per le istanze EC2 e il sistema di bilanciamento del carico dell'ambiente. Offrono anche la possibilità di configurare un database Amazon RDS integrato con il tuo ambiente.

Traffico e scalabilità delle istanze

Queste opzioni personalizzano la capacità, la scalabilità e il bilanciamento del carico per le istanze EC2 dell'ambiente. Puoi anche configurare Elastic Load Balancing per acquisire log con informazioni dettagliate sulle richieste inviate al load balancer.

Per la configurazione sono disponibili anche le seguenti opzioni per le istanze EC2:

- Tipo di volume principale, dimensione, velocità input/output operativa (IOPS) e velocità effettiva.
- Attivazione del servizio di metadati delle istanze (IMDS).
- Selezione di gruppi di sicurezza EC2 per controllare il traffico delle istanze.
- CloudWatch intervallo di monitoraggio delle metriche.
- Intervallo di tempo per la registrazione delle metriche.

Aggiornamenti, monitoraggio e registrazione

Questa categoria configura le seguenti opzioni:

- Reportistica sullo stato dell'ambiente, inclusa la possibilità di selezionare la segnalazione avanzata sullo stato di salute.
- Aggiornamenti gestiti della piattaforma che definiscono quando e come Elastic Beanstalk implementa le modifiche all'ambiente.
- Attivazione del X-Ray servizio per raccogliere dati sul comportamento dell'applicazione per identificare problemi e opportunità di ottimizzazione.
- Opzioni specifiche della piattaforma, tra cui il server proxy e le proprietà dell'ambiente del sistema operativo.

Navigazione nella pagina di configurazione

Scegliete Modifica in una categoria di configurazione per avviare la pagina di configurazione associata, in cui è possibile visualizzare i valori completi delle opzioni e apportare modifiche.

Navigazione in una categoria di configurazione

Naviga in una pagina di categoria di configurazione con una delle seguenti azioni:

- Annulla: torna alla pagina di panoramica della configurazione senza applicare le modifiche alla configurazione. Selezionando Cancel (Annulla), la console perde eventuali modifiche in sospeso eseguite su qualsiasi categoria di configurazione.

Puoi anche annullare le modifiche alla configurazione scegliendo un altro elemento nella pagina di navigazione a sinistra, come Eventi o Registri.

- Continue (Continua): torna alla pagina Configuration overview (Panoramica configurazione). È quindi possibile continuare ad apportare modifiche o applicare quelle in sospeso.
- Applica: applica le modifiche apportate in una qualsiasi delle categorie di configurazione al tuo ambiente. In alcuni casi viene richiesto di confermare la conseguenza di una delle tue decisioni di configurazione.

Navigazione su Configurazione pagina di panoramica

Scegli Modifica in una categoria di configurazione per avviare una pagina di configurazione correlata, in cui è possibile visualizzare i valori completi delle opzioni e apportare modifiche. Dopo aver visualizzato e modificato le opzioni, puoi scegliere una delle seguenti azioni dalla pagina di panoramica della configurazione:

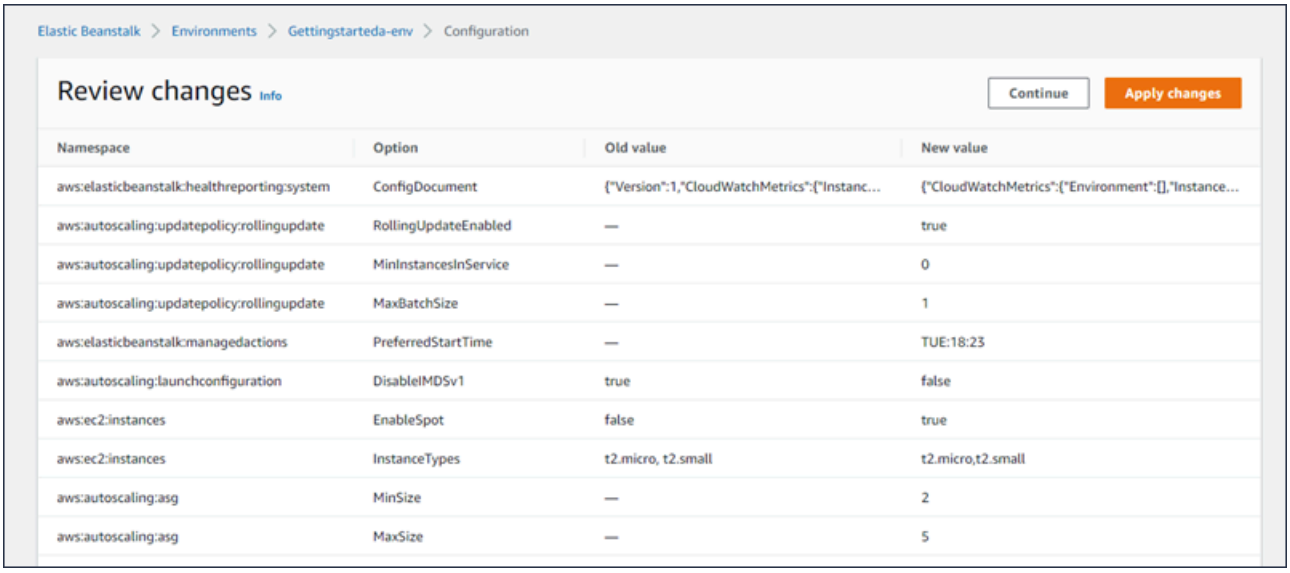
- Cancel (Annulla): torna al pannello di controllo dell'ambiente senza applicare le modifiche di configurazione. Selezionando Cancel (Annulla), la console perde eventuali modifiche in sospeso eseguite su qualsiasi categoria di configurazione.

  Puoi anche annullare le modifiche alla configurazione scegliendo un altro elemento nella pagina di navigazione a sinistra, come Eventi o Registri.
- Review changes (Rivedi modifiche): visualizza un riepilogo di tutte le modifiche apportate in sospeso su qualsiasi categoria di configurazione. Per informazioni dettagliate, vedi Pagina Review Changes (Controlla modifiche).
- Apply changes (Applica modifiche): applica al tuo ambiente le modifiche apportate nelle categorie di configurazione. In alcuni casi viene richiesto di confermare la conseguenza di una delle tue decisioni di configurazione.

Pagina Review Changes (Controlla modifiche)

La pagina Review Changes (Controlla modifiche) mostra una tabella in cui sono riportate tutte modifiche delle opzioni in sospeso apportate in una qualsiasi delle schede di configurazione e che non sono state ancora applicate all'ambiente.

Le tabelle elencano ciascuna opzione come una combinazione di Namespace (Spazio dei nomi) e Option Name (Nome opzione) con cui viene identificata da Elastic Beanstalk. Per informazioni dettagliate, vedi Opzioni di configurazione.



Namespace	Option	Old value	New value
aws:elasticbeanstalk:healthreporting:system	ConfigDocument	{"Version":1,"CloudWatchMetrics":{"Instanc...	{"CloudWatchMetrics":{"Environment":[],"Instance...
aws:autoscaling:updatepolicy:rollingupdate	RollingUpdateEnabled	—	true
aws:autoscaling:updatepolicy:rollingupdate	MinInstancesInService	—	0
aws:autoscaling:updatepolicy:rollingupdate	MaxBatchSize	—	1
aws:elasticbeanstalk:managedactions	PreferredStartTime	—	TUE:18:23
aws:autoscaling:launchconfiguration	DisableIMDSv1	true	false
aws:ec2:instances	EnableSpot	false	true
aws:ec2:instances	InstanceTypes	t2.micro, t2.small	t2.micro,t2.small
aws:autoscaling:asg	MinSize	—	2
aws:autoscaling:asg	MaxSize	—	5

Al termine della verifica delle modifiche, è possibile scegliere una delle seguenti operazioni:

- Continue (Continua): torna alla pagina Configuration overview (Panoramica configurazione). È quindi possibile continuare ad apportare modifiche o applicare quelle in sospeso.
- Apply changes (Applica modifiche): applica al tuo ambiente le modifiche apportate nelle categorie di configurazione. In alcuni casi viene richiesto di confermare la conseguenza di una delle tue decisioni di configurazione.

Le EC2 istanze Amazon per il tuo ambiente Elastic Beanstalk

Questo argomento spiega le EC2 istanze Amazon e le opzioni di configurazione che influiscono sul tuo ambiente Elastic Beanstalk.

Quando crei un ambiente server Web, AWS Elastic Beanstalk crea una o più macchine virtuali Amazon Elastic Compute Cloud (Amazon EC2), note come istanze.

Le istanze dell'ambiente sono configurate per eseguire app Web sulla piattaforma scelta. È possibile apportare modifiche a varie proprietà e comportamenti delle istanze dell'ambiente durante la creazione dell'ambiente o dopo che è già in esecuzione. In alternativa, è possibile apportare subito queste variazioni modificando il codice sorgente distribuito nell'ambiente. Per ulteriori informazioni, consulta the section called “Opzioni di configurazione”.

> **Note**
>
> Il gruppo Auto Scaling nel tuo ambiente gestisce le EC2 istanze Amazon che eseguono la tua applicazione. Quando apporti le modifiche alla configurazione descritte in questo argomento, cambia anche la configurazione di avvio. La configurazione di avvio è un modello di EC2 lancio di Amazon o una risorsa di configurazione di avvio del gruppo Auto Scaling. Questa modifica richiede la sostituzione di tutte le istanze. Innesca anche un aggiornamento in sequenza o un aggiornamento immutabile, a seconda di quale è configurato.

EC2 opzioni di acquisto delle istanze

Elastic Beanstalk supporta EC2 diverse opzioni di acquisto di istanze Amazon:

- On-Demand: un'istanza On-Demand è una pay-as-you-go risorsa: non è richiesto alcun impegno a lungo termine per utilizzarla.
- Riservata: un'istanza riservata è uno sconto di fatturazione preacquistato applicato automaticamente alle istanze On-Demand corrispondenti nel tuo ambiente.
- Spot: un'istanza Spot è un' EC2 istanza Amazon inutilizzata disponibile a un prezzo inferiore al prezzo On-Demand. Puoi abilitare e configurare l'allocazione delle istanze Spot nel tuo ambiente. Per ulteriori informazioni, consulta Auto Scaling delle istanze dell'ambiente Elastic Beanstalk.

Argomenti

- Tipi di EC2 istanze Amazon
- Configurazione delle EC2 istanze Amazon utilizzando la console Elastic Beanstalk
- Gestione dei gruppi di sicurezza EC2
- Configurazione dei gruppi di sicurezza e dei tipi di istanze di Amazon EC2 utilizzando AWS CLI
- Configurazione delle EC2 istanze Amazon con opzioni di namespace
- Configurazione dell'IMDS sulle istanze dell'ambiente Elastic Beanstalk

Tipi di EC2 istanze Amazon

Questo argomento spiega il termine tipo di istanza. Quando crei un nuovo ambiente, Elastic Beanstalk effettua il EC2 provisioning di istanze Amazon basate sui tipi di istanze Amazon che EC2 scegli. I tipi di istanza scelti determinano l'hardware host su cui vengono eseguite le istanze. EC2 i tipi

di istanza possono essere classificati in base all'architettura di processore su cui si basa ciascuna. Elastic Beanstalk supporta tipi di istanze basati sulle seguenti AWS architetture di processore: architettura Graviton 64-bit Arm (arm64), architettura a 64 bit (x86) e architettura a 32 bit (i386). Elastic Beanstalk seleziona l'architettura del processore x86 per impostazione predefinita quando crei un nuovo ambiente.

> **Note**
>
> L'architettura a 32 bit i386 non è più supportata dalla maggior parte delle piattaforme Elastic Beanstalk. Consigliamo di scegliere invece i tipi di architettura x86 o arm64. Elastic Beanstalk fornisce opzioni di configurazione per i tipi di istanza del processore i386 nello spazio dei nomi `aws:ec2:instances`.

Tutti i tipi di istanza nella configurazione per un determinato ambiente Elastic Beanstalk devono avere lo stesso tipo di architettura del processore. Supponiamo di aggiungere un nuovo tipo di istanza a un ambiente esistente che ha già un tipo di istanza t2.medium, basato sull'architettura x86. È possibile aggiungere solo un altro tipo di istanza della stessa architettura, come ad esempio t2.small. Se desideri sostituire i tipi di istanza esistenti con quelli di un'architettura diversa, puoi farlo. Assicurati però che tutti i tipi di istanza inclusi nel comando siano basati sullo stesso tipo di architettura.

Elastic Beanstalk aggiunge regolarmente il supporto per nuovi tipi di istanze compatibili dopo che Amazon li ha introdotti. EC2 Per informazioni sui tipi di istanza disponibili, consulta Tipi di istanza nella Amazon EC2 User Guide.

> **Note**
>
> Elastic Beanstalk ora offre supporto per Graviton su tutte le più recenti piattaforme Amazon Linux 2 in tutte le regioni supportate da Graviton. AWS Per ulteriori informazioni sulla creazione di un ambiente Elastic Beanstalk con tipi di istanze basate su arm64, consulta Configurazione delle EC2 istanze Amazon utilizzando la console Elastic Beanstalk.
> Crea nuovi ambienti che eseguono EC2 istanze Amazon sull'architettura arm64 e migra le applicazioni esistenti su di essi con le opzioni di distribuzione di Elastic Beanstalk.
> Per saperne di più sui processori basati su Graviton arm64, consulta queste risorse: AWS
>
> - Vantaggi: il AWS processore Graviton

- Guida introduttiva e altri argomenti, come considerazioni specifiche sulla lingua — Articolo introduttivo su Graviton AWS GitHub

Configurazione delle EC2 istanze Amazon utilizzando la console Elastic Beanstalk

Puoi creare o modificare la configurazione dell'istanza Amazon del tuo ambiente Elastic Beanstalk nella console Elastic EC2 Beanstalk.

> **Note**
>
> Sebbene la console Elastic Beanstalk non offra la possibilità di modificare l'architettura del processore di un ambiente esistente, puoi farlo con. AWS CLI Ad esempio, i comandi, vedi. Configurazione dei gruppi di sicurezza e dei tipi di istanze di Amazon EC2 utilizzando AWS CLI

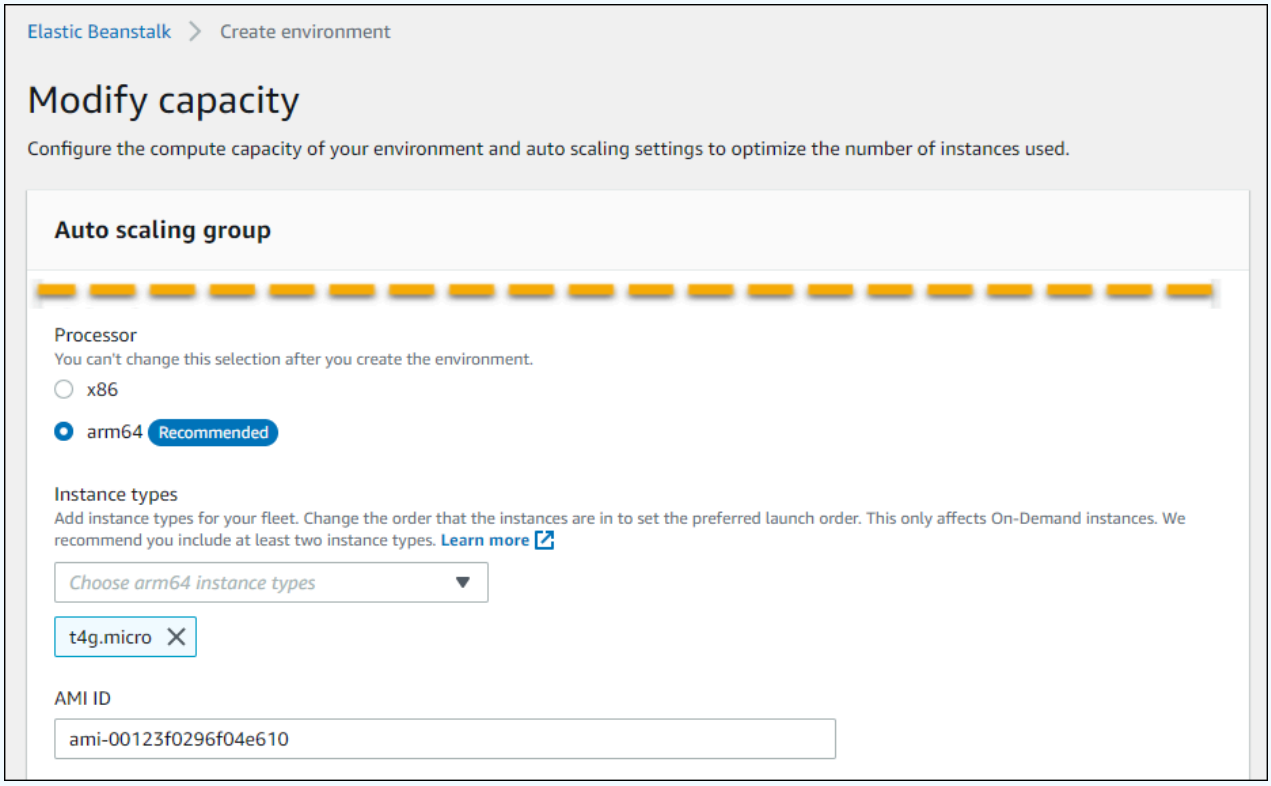
Per configurare le EC2 istanze Amazon nella console Elastic Beanstalk durante la creazione dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegliere Create a new environment (Crea un nuovo ambiente) per iniziare a creare l'ambiente.
4. Nella pagina principale della procedura guidata, prima di scegliere Create environment (Crea ambiente), scegli Configure more options (Configura più opzioni).
5. Nella categoria di configurazione Instances (Istanze), scegliere Edit (Modifica). Apportare modifiche alle impostazioni in questa categoria e quindi scegliere Applica. Per le descrizioni delle impostazioni, vedere la sezione the section called “Impostazioni della categoria istanze” di questa pagina.
6. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica). Apportare modifiche alle impostazioni in questa categoria e quindi scegliere Continua. Per le descrizioni delle impostazioni, vedere la sezione the section called “Impostazioni della categoria capacità” di questa pagina.

Selezione dell'architettura del processore

Scorri verso il basso fino a Processore per selezionare un'architettura di processore per le tue istanze. EC2 La console elenca le architetture dei processori supportate dalla piattaforma che hai scelto in precedenza nel pannello Create environment (Crea dell'ambiente).

Se l'architettura del processore desiderata non viene visualizzata, torna all'elenco delle categorie di configurazione per selezionare una piattaforma che la supporta. Dal pannello Modify Capacity (Modifica capacità), scegliCancel (Annulla). Quindi, scegli Change platform version (Modifica versione piattaforma) per scegliere nuove impostazioni della piattaforma. Poi, nella categoria di configurazione Capacity (Capacità) scegli Edit (Modifica) per visualizzare di nuovo le scelte dell'architettura del processore.



7. Scegli Save (Salva) e apporta eventuali altre modifiche di configurazione richieste dal tuo ambiente.
8. Seleziona Create environment (Crea ambiente).

Per configurare le EC2 istanze Amazon di un ambiente in esecuzione nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Instances (Istanze), scegliere Edit (Modifica). Apportare modifiche alle impostazioni in questa categoria e quindi scegliere Applica. Per le descrizioni delle impostazioni, vedere la sezione the section called “Impostazioni della categoria istanze” di questa pagina.
5. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica). Apportare modifiche alle impostazioni in questa categoria e quindi scegliere Continua. Per le descrizioni delle impostazioni, vedere la sezione the section called “Impostazioni della categoria capacità” di questa pagina.

Impostazioni della categoria istanze

Le seguenti impostazioni relative alle EC2 istanze Amazon sono disponibili nella categoria Configurazione istanze.

Opzioni

- Intervallo di monitoraggio
- Volume root (dispositivo di avvio)
- Servizio di metadati dell'istanza
- EC2 gruppi di sicurezza

Elastic Beanstalk > Environments > Gettingstarted-env > Configuration

Modify instances

Amazon CloudWatch monitoring

The time interval between when metrics are reported from the EC2 instances.

Monitoring interval

5 minute

Root volume (boot device)

Root volume type

(Container default)

Size

The number of gigabytes of the root volume attached to each instance.

GB

IOPS

Input/output operations per second for a provisioned IOPS (SSD) volume.

100 IOPS

Throughput

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

MiB/s

Instance metadata service (IMDS)

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. Learn more

Disable IMDSv1

With the current setting, the environment enables both IMDSv1 and IMDSv2.

☑ Disabled

EC2 security groups

	Group name	Group ID	Name
☐	awseb-e-awppgphwta-stack-AWSEBLoadBalancerSecurityGroup-LUAOUHKL3SNI	sg-027aafe45182f171f	WinTest-dev
☐	awseb-e-awppgphwta-stack-AWSEBSecurityGroup-109O5QSLX6UCC	sg-020e30e60b3e80c5b	WinTest-dev
☐	awseb-e-m5yhre5nuj-stack-AWSEBLoadBalancerSecurityGroup-PIICIFO0QHGG	sg-03879e31c4ebe98ea	Gettingstarted-env
☑	awseb-e-m5yhre5nuj-stack-AWSEBSecurityGroup-12122MOSKFTC4	sg-05b1982101cf211ef	Gettingstarted-env
☐	default	sg-3527cd14	

Cancel Continue Apply

Intervallo di monitoraggio

Per impostazione predefinita, le istanze del tuo ambiente pubblicano parametri sanitari di base su Amazon CloudWatch a intervalli di cinque minuti senza costi aggiuntivi.

Per report più dettagliati, puoi impostare l'intervallo di monitoraggio su 1 minuto per aumentare la frequenza con cui le risorse del tuo ambiente pubblicano i parametri sanitari di base su at. CloudWatch CloudWatch si applicano i costi di servizio per le metriche con intervallo di un minuto. Per ulteriori informazioni, consulta Amazon CloudWatch.

Volume root (dispositivo di avvio)

Ogni istanza nell'ambiente viene configurata con un volume radice. Il volume radice è il dispositivo a blocchi Amazon EBS collegato all'istanza per archiviare il sistema operativo, le librerie, gli script e il codice sorgente dell'applicazione. Per impostazione predefinita, tutte le piattaforme utilizzano i dispositivi a blocchi SSD di uso generale per lo storage.

È possibile modificare il Root volume type (Tipo di volume radice) per utilizzare lo storage magnetico o i tipi di volume provisioned IOPS SSD e aumentare le dimensioni del volume, se necessario. Per i volumi Provisioned IOPS, è necessario selezionare anche il numero di IOPS per i quali effettuare il provisioning. La velocità effettiva è applicabile solo ai tipi di volume SSD gp3. È possibile inserire la velocità effettiva desiderata per il provisioning. Può variare tra 125 e 1.000 mebibyte al secondo (MIB/s). Seleziona il tipo di volume che meglio soddisfa i tuoi requisiti in termini di prestazioni e prezzi.

> **Important**
>
> L'impostazione delle `RootVolumeType` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Per ulteriori informazioni, consulta Amazon EBS Volume Types nella Amazon EC2 User Guide e i dettagli del prodotto Amazon EBS.

Servizio di metadati dell'istanza

Il servizio di metadati dell'istanza (IMDS) è un componente dell'istanza utilizzato dal codice sull'istanza per accedere in modo sicuro ai metadati dell'istanza. Il codice può accedere ai metadati dell'istanza da un'istanza in esecuzione utilizzando uno dei due metodi. Sono Instance Metadata Service Version 1 (IMDSv1) o Instance Metadata Service Version 2 (). IMDSv2 IMDSv2 è più sicuro. Disabilita IMDSv1 per imporre IMDSv2. Per ulteriori informazioni, consulta the section called “IMDS”.

> **Note**
>
> La sezione IMDS in questa pagina di configurazione viene visualizzata solo per le versioni della piattaforma che supportano. IMDSv2

EC2 gruppi di sicurezza

I gruppi di sicurezza collegati alle istanze determinano il traffico autorizzato a raggiungere e uscire dalle istanze.

Il gruppo EC2 di sicurezza predefinito creato da Elastic Beanstalk consente tutto il traffico in entrata da Internet o dai sistemi di bilanciamento del carico sulle porte standard per HTTP (80) e SSH (22). Puoi anche definire gruppi di sicurezza personalizzati per designare le regole del firewall per le istanze. EC2 I gruppi di sicurezza possono consentire il traffico su altre porte o da altre fonti. Ad esempio, è possibile creare un gruppo di sicurezza per l'accesso SSH che consente il traffico in entrata sulla porta 22 da un intervallo di indirizzi IP limitati. Oppure, per una maggiore sicurezza, puoi crearne uno che consenta il traffico proveniente da un host bastion a cui solo tu puoi accedere.

Puoi scegliere di disattivare il tuo ambiente dal gruppo di EC2 sicurezza predefinito impostando l'`DisableDefaultEC2SecurityGroup`opzione nel aws:autoscaling:launchconfiguration namespace su. `true` Questa opzione non è disponibile nella console, ma è possibile impostarla con. AWS CLI Per ulteriori informazioni, consulta Gestione dei gruppi di sicurezza EC2.

Per ulteriori informazioni sui gruppi di EC2 sicurezza Amazon, consulta Amazon EC2 Security Groups nella Amazon EC2 User Guide.

> **Note**
>
> Per consentire il traffico tra le istanze dell'ambiente A e le istanze dell'ambiente B, è possibile aggiungere una regola al gruppo di sicurezza che Elastic Beanstalk ha collegato

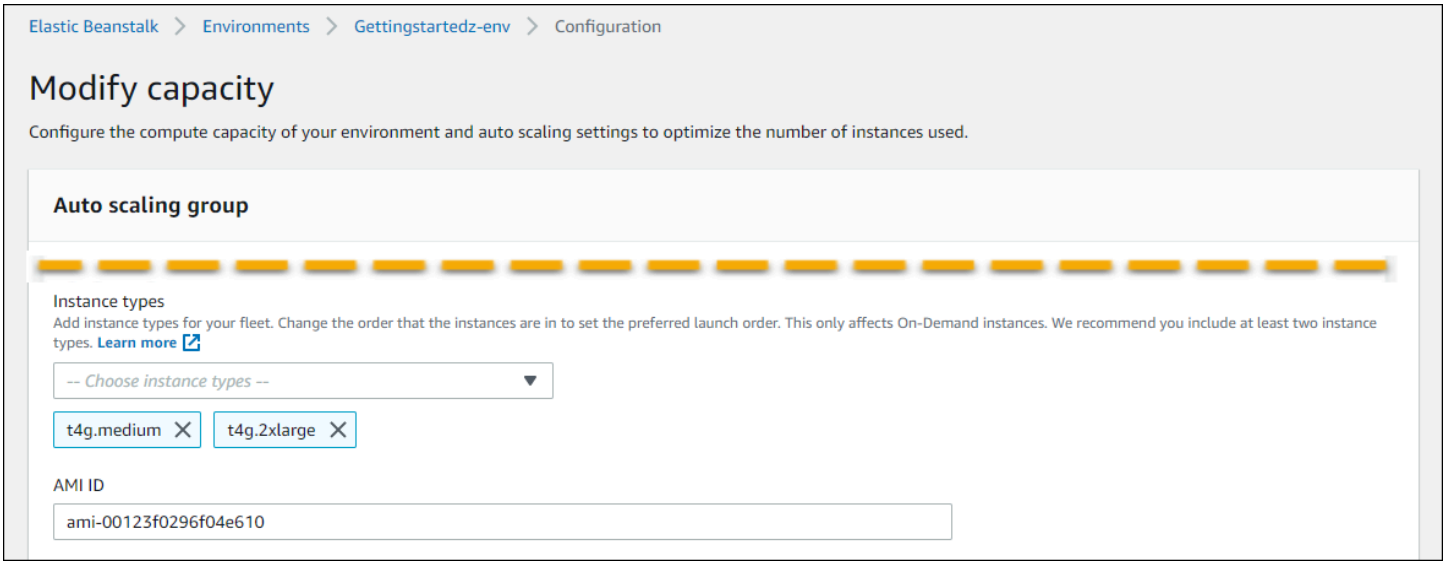
all'ambiente B.Quindi, è possibile specificare il gruppo di sicurezza che Elastic Beanstalk ha collegato all'ambiente A. Questo consente il traffico in entrata o in uscita dalle istanze dell'ambiente A. Tuttavia, in questo modo si crea una dipendenza tra i due gruppi di sicurezza. Successivamente, quando tenti di terminare l'ambiente A, Elastic Beanstalk non è in grado di eliminare il gruppo di sicurezza dell'ambiente, poiché il gruppo di sicurezza dell'ambiente B dipende da esso.
Pertanto, è preferibile creare prima un gruppo di sicurezza separato. Quindi, allegalo all'ambiente A e specificalo in una regola del gruppo di sicurezza dell'ambiente B.

Impostazioni della categoria capacità

Le seguenti impostazioni relative alle EC2 istanze Amazon sono disponibili nella categoria di configurazione Capacity.

Opzioni

- Tipi di istanza
- ID AMI



Tipi di istanza

L'impostazione Instance types determina il tipo di EC2 istanza Amazon che viene lanciata per eseguire l'applicazione. Questa pagina di configurazione mostra un elenco di tipi di istanza. È possibile selezionare uno o più tipi di istanza. La console Elastic Beanstalk mostra solo i tipi di istanza

basati sull'architettura del processore configurata per l'ambiente. Pertanto, è possibile aggiungere solo tipi di istanza della stessa architettura del processore.

> **Note**
>
> Sebbene la console Elastic Beanstalk non offra la possibilità di modificare l'architettura del processore di un ambiente esistente, puoi farlo con. AWS CLI Ad esempio, i comandi, vedi. Configurazione dei gruppi di sicurezza e dei tipi di istanze di Amazon EC2 utilizzando AWS CLI

Scegli un'istanza che sia abbastanza potente per l'esecuzione dell'applicazione sotto carico, ma non così potente da essere inattiva la maggior parte del tempo. Per le attività di sviluppo, la famiglia t2 di istanze fornisce una quantità moderata di potenza con la possibilità di aumentare le prestazioni per brevi periodi di tempo. Per dimensionamenti di grandi dimensioni e applicazioni ad alta disponibilità, utilizza un pool di istanze per essere certo che la capacità non ne risenta molto se una singola istanza non è funzionante. Inizia con un tipo di istanza che si può utilizzare per eseguire cinque istanze con carico moderato durante le ore normali. Se un'istanza ha esito negativo, il resto delle istanze è in grado di assorbire il resto del traffico. Il buffer della capacità consente anche all'ambiente di scalare quando il traffico inizia ad aumentare durante le ore più intense.

Per ulteriori informazioni sulle famiglie e i tipi di EC2 istanze Amazon, consulta Tipi di istanze nella Amazon EC2 User Guide. Per determinare quali tipi di istanze soddisfano i tuoi requisiti e le relative regioni supportate, consulta i tipi di istanze disponibili nella Amazon EC2 User Guide.

ID AMI

L'Amazon Machine Image (AMI) è l'immagine della macchina Amazon Linux o Windows Server utilizzata da Elastic Beanstalk per EC2 avviare istanze Amazon nel tuo ambiente. Elastic Beanstalk fornisce le immagini di macchine che contengono gli strumenti e le risorse necessari per l'esecuzione dell'applicazione.

Elastic Beanstalk seleziona un'AMI predefinita per l'ambiente in base a Regione, versione della piattaforma e architettura del processore scelti. Se hai creato un'AMI personalizzata, sostituisci l'ID dell'AMI di default con il tuo ID di default personalizzato.

Gestione dei gruppi di sicurezza EC2

Quando Elastic Beanstalk crea un ambiente, assegna un gruppo di sicurezza predefinito alle istanze EC2 che vengono avviate con esso. I gruppi di sicurezza collegati alle istanze determinano il traffico autorizzato a raggiungere e uscire dalle istanze.

Il gruppo di sicurezza EC2 predefinito creato da Elastic Beanstalk consente tutto il traffico in entrata da Internet o dai sistemi di bilanciamento del carico sulle porte standard per HTTP (80) e SSH (22). Puoi anche definire i tuoi gruppi di sicurezza personalizzati per designare le regole del firewall per le istanze EC2. I gruppi di sicurezza possono consentire il traffico su altre porte o da altre fonti. Ad esempio, è possibile creare un gruppo di sicurezza per l'accesso SSH che consente il traffico in entrata sulla porta 22 da un intervallo di indirizzi IP limitati. Oppure, per una maggiore sicurezza, puoi crearne uno che consenta il traffico proveniente da un host bastion a cui solo tu puoi accedere.

Puoi scegliere di disattivare il tuo ambiente dal gruppo di sicurezza EC2 predefinito impostando l'`DisableDefaultEC2SecurityGroup`opzione nel aws:autoscaling:launchconfiguration namespace su. `true` Utilizza i file di configurazione AWS CLIo per applicare questa opzione al tuo ambiente e per collegare gruppi di sicurezza personalizzati alle istanze EC2.

Gestione dei gruppi di sicurezza EC2 in ambienti a più istanze

Se crei un gruppo di sicurezza EC2 personalizzato in un ambiente a più istanze, devi anche considerare in che modo i bilanciatori del carico e le regole del traffico in entrata mantengono le tue istanze sicure e accessibili.

Il traffico in entrata verso un ambiente con più istanze EC2 è gestito dal sistema di bilanciamento del carico, che indirizza il traffico in entrata tra tutte le istanze EC2. Quando Elastic Beanstalk crea un gruppo di sicurezza EC2 predefinito, definisce anche regole in entrata che consentono il traffico in entrata dal load balancer. Senza questa regola in entrata nel gruppo di sicurezza, il traffico in entrata non potrà entrare nelle istanze. Questa condizione bloccherebbe essenzialmente le istanze dalle richieste esterne.

Se disabiliti il gruppo di sicurezza EC2 predefinito per un ambiente con bilanciamento del carico, Elastic Beanstalk convalida alcune regole di configurazione. Se la configurazione non soddisfa i controlli di convalida, invia messaggi che indicano all'utente di fornire la configurazione richiesta. I controlli di convalida sono i seguenti:

- Almeno un gruppo di sicurezza deve essere assegnato al load balancer utilizzando l'`SecurityGroups`opzione aws:elbv2:loadbalancer oaws:elb:loadbalancer, a seconda che si tratti

rispettivamente di un application load balancer o di un load balancer classico. Per AWS CLI alcuni esempi, vedere. Configurazione con AWS CLI

- È necessario che esistano regole sul traffico in entrata che consentano alle istanze EC2 di ricevere traffico dal sistema di bilanciamento del carico. Sia i gruppi di sicurezza EC2 che i gruppi di sicurezza del load balancer devono fare riferimento a queste regole in entrata. Per ulteriori informazioni, consultare la sezione seguente Regole per il traffico in entrata.

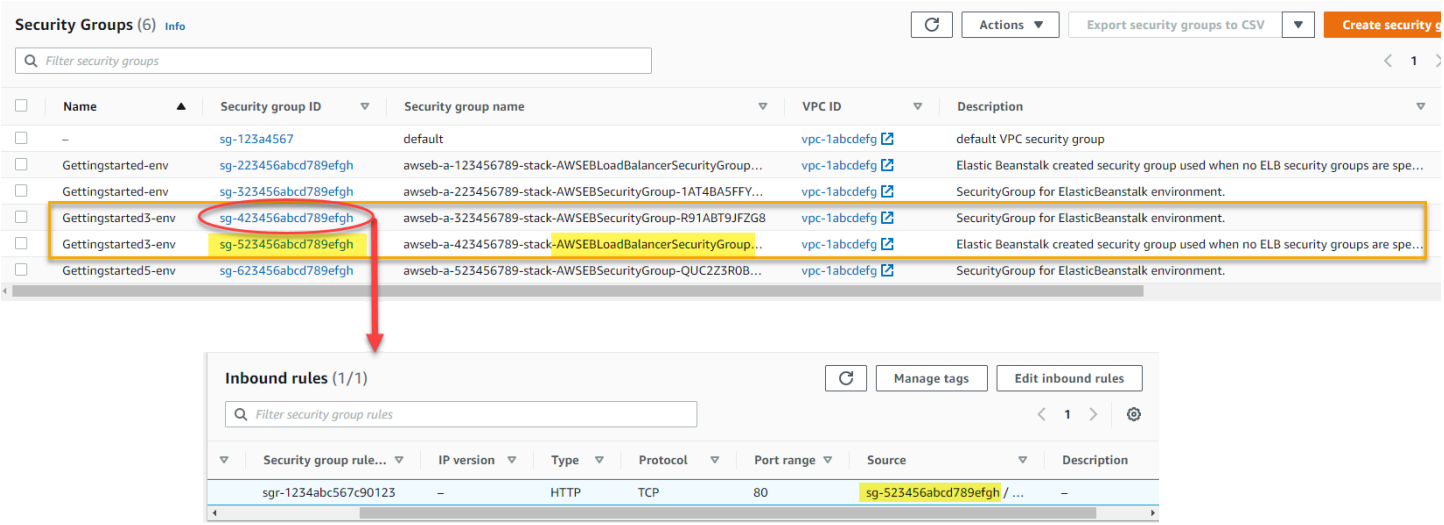
Regole per il traffico in entrata

I gruppi di sicurezza EC2 per un ambiente a più istanze devono includere una regola in entrata che faccia riferimento al gruppo di sicurezza del bilanciamento del carico. Ciò si applica agli ambienti con qualsiasi tipo di bilanciamento del carico, dedicato o condiviso, e con gruppi di sicurezza di bilanciamento del carico personalizzati o predefiniti.

Puoi visualizzare tutti i gruppi di sicurezza collegati ai componenti dell'ambiente nella console EC2. L'immagine seguente mostra l'elenco della console EC2 dei gruppi di sicurezza che Elastic Beanstalk crea di default durante l'operazione di creazione dell'ambiente.

La schermata Security Groups mostra gli ambienti e i gruppi di sicurezza associati. Entrambi GettingStarted3-envsono GettingStarted-envambienti a più istanze con sistemi di bilanciamento del carico dedicati. Ciascuno di questi ambienti presenta due gruppi di sicurezza elencati, uno per le istanze EC2 e l'altro per il bilanciamento del carico. Elastic Beanstalk crea questi gruppi di sicurezza quando crea gli ambienti. GettingStarted5-envnon dispone di un gruppo di sicurezza per il bilanciamento del carico, perché ha solo un'istanza EC2 e quindi nessun sistema di bilanciamento del carico.

La schermata delle regole in entrata approfondisce il gruppo di sicurezza EC2 per le istanze di. GettingStarted3-env Questo esempio definisce le regole in entrata per il gruppo di sicurezza EC2. Nota che la colonna Source nelle regole in entrata elenca l'id del gruppo di sicurezza del gruppo di sicurezza del load balancer elencato nell'immagine precedente. Questa regola consente alle istanze EC2 di GettingStarted3-envricevere traffico in entrata da quello specifico sistema di bilanciamento del carico sulla porta 80.



Per ulteriori informazioni, consulta Modifica dei gruppi di sicurezza per l'istanza e le regole Elastic Load Balancing nella Guida per l'utente di Amazon EC2.

Configurazione dei gruppi di sicurezza e dei tipi di istanze di Amazon EC2 utilizzando AWS CLI

Puoi utilizzare la AWS Command Line Interface (AWS CLI) per configurare le istanze Amazon EC2 nei tuoi ambienti Elastic Beanstalk.

Configurazione dei gruppi di sicurezza EC2 utilizzando AWS CLI

Questo argomento fornisce esempi di diverse configurazioni di gruppi di sicurezza EC2 per ambienti a istanza singola e con carico bilanciato (multiistanza). Per ulteriori informazioni sulle opzioni di questi esempi, consulta aws:autoscaling:launchconfiguration.

> **Note**
>
> L'operazione di creazione dell'ambiente fornisce un gruppo di sicurezza EC2 per impostazione predefinita. Inoltre, per impostazione predefinita, crea un ambiente con un sistema di bilanciamento del carico delle applicazioni.
> L'operazione di aggiornamento dell'ambiente può essere utilizzata per disabilitare o abilitare il gruppo di sicurezza EC2 predefinito per l'ambiente con l'opzione booleana. `DisableDefaultEC2SecurityGroup` L'esempio 5 mostra come riportare l'ambiente alla configurazione di sicurezza predefinita, se l'avevi modificato in precedenza.

Gli esempi seguenti mostrano un comando create-environment che disattiva il gruppo di sicurezza EC2 predefinito e fornisce invece gruppi di sicurezza personalizzati. Poiché l'`DisableDefaultEC2SecurityGroup`opzione è impostata su`true`, il gruppo di sicurezza EC2 predefinito che Elastic Beanstalk normalmente associa alle istanze EC2 non viene creato. Pertanto, è necessario fornire l'opzione ad altri gruppi di sicurezza. `SecurityGroups`

Si noti che l'aws:elasticbeanstalk:environment`EnvironmentType`opzione è impostata su`SingleInstance`. Per creare un ambiente a istanza singola, è necessario specificare questa opzione, poiché `LoadBalanced` è l'impostazione predefinita`EnvironmentType`. Poiché questo ambiente non include un load balancer, non è necessario specificare un gruppo di sicurezza del load balancer.

Example 1 — Nuovo ambiente a istanza singola con gruppi di sicurezza EC2 personalizzati (opzioni di namespace in linea)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 applrunning Node.js 22" \
--option-settings \
Namespace=aws:elasticbeanstalk:environment,OptionName=EnvironmentType,Value=SingleInstance
 \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-
elasticbeanstalk-ec2-role \
Namespace=aws:autoscaling:launchconfiguration,OptionName=DisableDefaultEC2SecurityGroup,Value=t
 \
Namespace=aws:autoscaling:launchconfiguration,OptionName=SecurityGroups,Value=sg-
abcdef01, sg-abcdef02 \
Namespace=aws:autoscaling:launchconfiguration,OptionName=EC2KeyName,Value=my-keypair
```

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga.

Example 2 — Nuovo ambiente a istanza singola con gruppi di sicurezza EC2 personalizzati (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
```

```
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  { "Namespace" : "aws:elasticbeanstalk:environment",
    "OptionName" : "EnvironmentType",
    "Value" : "SingleInstance"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "EC2KeyName",
    "Value": "my-keypair"
  }
]
```

L'esempio seguente crea un ambiente con carico bilanciato. Specifica l'opzione aws:elasticbeanstalk:environment namespace impostata su. `LoadBalancerType` `application` Poiché stiamo disabilitando il gruppo di sicurezza EC2 predefinito con l'`DisableDefaultEC2SecurityGroup`opzione, dobbiamo fornire nuovamente i nostri gruppi di sicurezza personalizzati per le istanze EC2, con l'opzione, come nell'aws:autoscaling:launchconfiguration`SecurityGroups`esempio precedente. Poiché questo ambiente dispone di un sistema di bilanciamento del carico per instradare il traffico, dobbiamo fornire gruppi di sicurezza anche per il sistema di bilanciamento del carico.

Per creare un ambiente con un sistema di bilanciamento del carico classico, ma per il resto con la stessa configurazione, aggiorna la configurazione dell'opzione aws:elasticbeanstalk:environment namespace a. `LoadBalancerType` `classic`

I diversi tipi di load balancer hanno namespace diversi che contengono le opzioni per specificare i gruppi di sicurezza:

- application load balancer — opzione aws:elbv2:loadbalancer `SecurityGroups`
- sistema di bilanciamento del carico classico — opzione aws:elb:loadbalancer `SecurityGroups`
- sistema di bilanciamento del carico di rete: poiché i sistemi di bilanciamento del carico di rete non dispongono di gruppi di sicurezza, configura i gruppi di sicurezza EC2 con identificatori VPC. Per ulteriori informazioni, consulta Aggiornare i gruppi di sicurezza per il Network Load Balancer nella Guida dell'utente per Network Load Balancer.

Example 3 — Nuovo ambiente multiistanza con gruppi di sicurezza EC2 personalizzati (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace" : "aws:elasticbeanstalk:environment",
    "OptionName" : "EnvironmentType",
    "Value" : "LoadBalanced"
  },
  {
  "Namespace" : "aws:elasticbeanstalk:environment",
    "OptionName" : "LoadBalancerType",
    "Value" : "application"
  },
  {
```

```
    "Namespace" : "aws:elbv2:loadbalancer",
    "OptionName" : "SecurityGroups",
    "Value" : "sg-abcdefghikl012345"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "EC2KeyName",
    "Value": "my-keypair"
  }
]
```

È possibile disabilitare il gruppo di sicurezza EC2 predefinito per un ambiente esistente con il comando update-environment. Il comando di esempio seguente disabilita il gruppo di sicurezza EC2 predefinito e assegna gruppi di sicurezza EC2 personalizzati alle istanze EC2 dell'ambiente.

Utilizza i `options.jason` file di esempio negli esempi 4 (a), 4 (b) o 4 (c), a seconda che l'ambiente sia bilanciato dal carico e dal tipo di bilanciamento del carico. Il file di configurazione 4 (a) specifica i gruppi di sicurezza per un ambiente a istanza singola. Poiché non richiede un sistema di bilanciamento del carico, forniamo solo il gruppo di sicurezza per le istanze EC2. I file di configurazione 4 (b) e 4 (c) specificano i gruppi di sicurezza per un sistema di bilanciamento del carico delle applicazioni e un sistema di bilanciamento del carico classico. In questi casi è inoltre necessario specificare i gruppi di sicurezza per il load balancer.

Example 4 — Aggiorna un ambiente esistente per disabilitare il gruppo di sicurezza EC2 predefinito (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk update-environment \
```

```
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example 4 (a) — File di configurazione per un ambiente a istanza singola (nessun sistema di bilanciamento del carico)

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Per aggiornare un ambiente che utilizza un sistema di bilanciamento del carico delle applicazioni, utilizzate lo spazio dei `aws:elbv2:loadbalancer` nomi per specificare i gruppi di sicurezza per il bilanciamento del carico.

Example 4 (b) — File di configurazione per un ambiente con un sistema di bilanciamento del carico delle applicazioni

```
### example options.json ###
[
  {
    "Namespace" : "aws:elbv2:loadbalancer",
    "OptionName" : "SecurityGroups",
    "Value" : "sg-abcdefghikl012345"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
```

```
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Per aggiornare un ambiente che utilizza un sistema di bilanciamento del carico classico, utilizzate lo spazio dei `aws:elb:loadbalancer` nomi per specificare i gruppi di sicurezza per il bilanciamento del carico.

Example 4 (c) — File di configurazione per un ambiente con un sistema di bilanciamento del carico classico

```
### example options.json ###
[
  {
    "Namespace" : "aws:elb:loadbalancer",
    "OptionName" : "SecurityGroups",
    "Value" : "sg-abcdefghikl012345"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",n
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Per riportare l'ambiente al comportamento e alla configurazione predefiniti con il gruppo di sicurezza predefinito assegnato da Elastic Beanstalk, usa il comando update-environment per impostare su. `DisableDefaultEC2SecurityGroup` `false` Per un ambiente a più istanze, Elastic Beanstalk gestisce anche i gruppi di sicurezza e le regole del traffico di rete per il bilanciamento del carico dell'ambiente.

L'esempio seguente si applica sia a un ambiente a istanza singola che a più istanze (carico bilanciato):

Example 5 — Riportare un ambiente all'utilizzo del gruppo di sicurezza predefinito (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v6.5.0 running Node.js 22" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "false"
  }
]
```

Configurazione di EC2 con tipi di istanze utilizzando AWS CLI

Questo argomento fornisce esempi per configurare i tipi di istanze delle istanze EC2 nel proprio ambiente.

I primi due esempi creano un nuovo ambiente. Il comando specifica un tipo di istanza Amazon EC2, t4g.small, basato sull'architettura del processore arm64. Elastic Beanstalk imposta l'ID Immagine (AMI) per le istanze EC2 in base alla regione, alla versione della piattaforma e al tipo di istanza. Il tipo di istanza corrisponde a un'architettura del processore. Il parametro `solution-stack-name` si applica alla versione della piattaforma.

Example 1 — Crea un nuovo ambiente basato su arm64 (opzioni dello spazio dei nomi in riga)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
```

```
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small
```

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga.

Example 2 — Crea un nuovo ambiente basato su arm64 (opzioni dello spazio dei nomi nel file **`options.json`**)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:ec2:instances",
    "OptionName": "InstanceTypes",
    "Value": "t4g.small"
  }
]
```

I due esempi successivi aggiornano la configurazione di un ambiente esistente con il comando update-environment. In questo esempio aggiungiamo un altro tipo di istanza basato ancora sull'architettura del processore arm64. Per gli ambienti esistenti, tutti i tipi di istanza aggiunti devono avere la stessa architettura del processore. Se desideri sostituire i tipi di istanza esistenti con quelli di un'architettura diversa, puoi farlo. Ma assicuratevi che tutti i tipi di istanza del comando abbiano lo stesso tipo di architettura.

Example 3 — Aggiorna un ambiente basato su arm64 esistente (opzioni dello spazio dei nomi in linea)

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t4g.small,t4g.micro
```

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga.

Example 4 — Aggiorna un ambiente basato su arm64 esistente (opzioni dello spazio dei nomi nel file **`options.json`**)

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:ec2:instances",
    "OptionName": "InstanceTypes",
    "Value": "t4g.small, t4g.micro"
  }
]
```

I due esempi successivi mostrano di più comandi create-environment. Questi esempi non forniscono valori per `InstanceTypes`. Quando i valori `InstanceTypes` non sono specificati, Elastic Beanstalk ha come impostazione predefinita l'architettura del processore basata su x86. L'ID immagine (AMI) per le istanze EC2 dell'ambiente verrà impostata in maniera predefinita in base alla regione, alla versione della piattaforma e al tipo di istanza predefinito. Il tipo di istanza corrisponde a un'architettura del processore.

Example 5 — Crea un nuovo ambiente basato su x86 (opzioni dello spazio dei nomi in riga)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-
elasticbeanstalk-ec2-role
```

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga.

Example 6 — Crea un nuovo ambiente basato su x86 (opzioni dello spazio dei nomi nel file **`options.json`**)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.7 running Docker" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
```

```
    "Value": "aws-elasticbeanstalk-ec2-role"
  }
]
```

Configurazione delle EC2 istanze Amazon con opzioni di namespace

È possibile utilizzare le opzioni di configurazione nello spazio dei nomi `aws:autoscaling:launchconfiguration` per configurare le istanze del tuo ambiente, incluse le opzioni aggiuntive che non sono disponibili nella console.

> **Important**
>
> L'impostazione dell'`BlockDeviceMappings`opzione `DisableIMDSv1RootVolumeType`, o può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Il seguente esempio di file di configurazione utilizza le opzioni di configurazione di base illustrate in questo argomento. Per vedere esempi di opzioni di configurazione aggiuntive quando è necessario specificare gruppi di sicurezza per i sistemi di bilanciamento del carico, consulta. Configurazione con AWS CLI

```
option_settings:
  aws:autoscaling:launchconfiguration:
    SecurityGroups: my-securitygroup
    MonitoringInterval: "1 minute"
    DisableIMDSv1: false
    DisableDefaultEC2SecurityGroup: true
    SecurityGroups: "sg-abcdef01, sg-abcdef02"
    EC2KeyName: my-keypair
    IamInstanceProfile: "aws-elasticbeanstalk-ec2-role"
    BlockDeviceMappings: "/dev/sdj=:100,/dev/sdh=snap-51eef269,/dev/sdb=ephemeral0"
  aws:elasticbeanstalk:environment:
    EnvironmentType: SingleInstance
```

I `DisableDefaultEC2SecurityGroup` e non `BlockDeviceMappings` sono disponibili nella console.

È possibile utilizzare `BlockDeviceMappings` per configurare i dispositivi a blocchi aggiuntivi per le istanze. Per ulteriori informazioni, consulta Block Device Mapping nella Amazon EC2 User Guide.

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Configurazione dell'IMDS sulle istanze dell'ambiente Elastic Beanstalk

Questo argomento descrive l'Instance Metadata Service (IMDS).

I metadati dell'istanza sono dati correlati a un'istanza Amazon Elastic Compute Cloud EC2 (Amazon) che le applicazioni possono utilizzare per configurare o gestire l'istanza in esecuzione. Il servizio di metadati dell'istanza (IMDS) è un componente dell'istanza utilizzato dal codice sull'istanza per accedere in modo sicuro ai metadati dell'istanza. Questo codice può essere il codice della piattaforma Elastic Beanstalk sulle istanze dell'ambiente AWS , l'SDK che l'applicazione potrebbe utilizzare o persino il codice proprio dell'applicazione. Per ulteriori informazioni, consulta Metadati dell'istanza e dati utente nella Amazon EC2 User Guide.

Il codice può accedere ai metadati dell'istanza da un'istanza in esecuzione utilizzando uno dei due metodi: Instance Metadata Service Version 1 (IMDSv1) o Instance Metadata Service Version 2 (). IMDSv2 IMDSv2 utilizza richieste orientate alla sessione e mitiga diversi tipi di vulnerabilità che potrebbero essere utilizzate per tentare di accedere all'IMDS. Per informazioni su questi due metodi, consulta Configurazione del servizio di metadati dell'istanza nella Amazon EC2 User Guide.

Argomenti

- Supporto della piattaforma per IMDS
- Scelta dei metodi IMDS
- Configurazione di IMDS tramite la console Elastic Beanstalk
- Spazio dei nomi aws:autoscaling:launchconfiguration

Supporto della piattaforma per IMDS

Le piattaforme Elastic Beanstalk in esecuzione su Amazon Linux 2 e Amazon Linux 2023 e Windows Server supportano tutte e due le piattaforme. IMDSv1 IMDSv2 Per ulteriori informazioni, consulta Configurazione di IMDS tramite la console Elastic Beanstalk

Scelta dei metodi IMDS

Quando si prende una decisione sui metodi IMDS che si desidera che l'ambiente supporti , considerare i seguenti casi d'uso:

- AWS SDK: se la tua applicazione utilizza un AWS SDK, assicurati di utilizzare la versione più recente dell'SDK. AWS SDKsEffettua chiamate IMDS e utilizza le versioni SDK più recenti quando possibile. IMDSv2 Se disattivi IMDSv1 o se l'applicazione utilizza una vecchia versione SDK, le chiamate IMDS potrebbero non riuscire.
- Il codice dell'applicazione: se l'applicazione effettua chiamate IMDS, valuta la possibilità di utilizzare l' AWS SDK in modo da poter effettuare le chiamate anziché effettuare richieste HTTP dirette. In questo modo, non è necessario apportare modifiche al codice per passare tra i metodi IMDS. L' AWS SDK lo utilizza IMDSv2 quando possibile.
- Codice della piattaforma Elastic Beanstalk: il nostro codice effettua chiamate IMDS AWS tramite l'SDK e IMDSv2 quindi lo utilizza su tutte le versioni della piattaforma di supporto. Se il codice utilizza un up-to-date AWS SDK ed effettua tutte le chiamate IMDS tramite l'SDK, puoi disattivarlo in tutta sicurezza. IMDSv1

Configurazione di IMDS tramite la console Elastic Beanstalk

Puoi modificare la configurazione dell'istanza Amazon del tuo ambiente Elastic Beanstalk nella console Elastic EC2 Beanstalk.

> **Important**
>
> L'impostazione delle `DisableIMDSv1` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non

riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Per configurare IMDS sulle tue EC2 istanze Amazon nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Instance traffic and scaling (Traffico e dimensionamento delle istanze), scegli Edit (Modifica).
5. Imposta Disabilita IMDSv1 per imporre. IMDSv2 IMDSv1Deseleziona Disabilita per abilitare entrambi IMDSv1 e IMDSv2.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Spazio dei nomi aws:autoscaling:launchconfiguration

È possibile utilizzare una opzione di configurazione nello spazio dei nomi `aws:autoscaling:launchconfiguration` per configurare IMDS nelle istanze dell'ambiente.

> **Important**
>
> L'impostazione delle `DisableIMDSv1` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Il seguente esempio di file di configurazione disabilita IMDSv1 l'utilizzo dell'`DisableIMDSv1`opzione.

```
option_settings:
  aws:autoscaling:launchconfiguration:
```

```
    DisableIMDSv1: true
```

Impostare Disable IMDSv1 su `true` per disabilitare IMDSv1 e applicare IMDSv2.

Imposta Disabilita `false` su IMDSv1 per abilitare entrambi IMDSv1 e IMDSv2.

Auto Scaling delle istanze dell'ambiente Elastic Beanstalk

Questo argomento descrive come personalizzare le funzionalità di Auto Scaling per gestire il carico di lavoro dell'ambiente Elastic Beanstalk. Puoi configurare Auto Scaling per il tuo ambiente utilizzando la console Elastic Beanstalk, le opzioni di configurazione dello spazio dei nomi o l'EB CLI. AWS CLI

Ambienti con carico bilanciato o a istanza singola

Il tuo AWS Elastic Beanstalk ambiente include un gruppo Auto Scaling che gestisce le istanze Amazon EC2 nel tuo ambiente. In un ambiente con singola istanza, il gruppo Auto Scaling garantisce che esista sempre un'istanza in esecuzione. In un ambiente con bilanciamento del carico, è possibile configurare il gruppo con un'ampia gamma di istanze per l'esecuzione e Auto Scaling consente di aggiungere o rimuovere le istanze in base alle esigenze del carico.

Configurazione dell'istanza EC2

Il gruppo Auto Scaling applica anche le tue scelte di configurazione per effettuare il provisioning e gestire le istanze EC2 nel tuo ambiente. Puoi modificare la configurazione EC2 per cambiare il tipo di istanza, la coppia di chiavi, lo storage Amazon Elastic Block Store (Amazon EBS) e altre impostazioni che possono essere configurate solo all'avvio di un'istanza.

Istanze On-Demand e Spot

Come opzione, Elastic Beanstalk può includere istanze Spot nell'ambiente e gestirle in combinazione con istanze On-Demand. Puoi configurare Amazon EC2 Auto Scaling per monitorare e rispondere automaticamente alle modifiche che influiscono sulla disponibilità delle istanze Spot abilitando il Ribilanciamento della capacità. Puoi anche configurare Strategia di allocazione spot ciò che il servizio Auto Scaling utilizza per fornire istanze Spot al tuo ambiente.

Autorizzazioni richieste per l'abilitazione delle istanze Spot

Per abilitare le richieste di istanze Spot è necessario utilizzare i modelli di avvio Amazon EC2. Quando si configura tale caratteristica durante la creazione dell'ambiente o gli aggiornamenti, Elastic

Beanstalk tenta di configurare l'ambiente in modo da utilizzare i modelli di avvio Amazon EC2 (se l'ambiente non li utilizza già). In questo caso, se la policy utente non dispone delle autorizzazioni necessarie, la creazione dell'ambiente o gli aggiornamenti potrebbero non riuscire. Pertanto, si consiglia di utilizzare la policy utente gestita o di aggiungere le autorizzazioni necessarie alle policy personalizzate. Per informazioni dettagliate sulle autorizzazioni necessarie, consulta Autorizzazioni richieste per i modelli di avvio.

Trigger di Auto Scaling

Il gruppo Auto Scaling utilizza due CloudWatch allarmi Amazon per attivare le operazioni di scalabilità. I trigger predefiniti eseguono il dimensionamento quando la media del traffico di rete in uscita da ciascuna istanza è superiore a 6 MiB o inferiore a 2 MiB per un intervallo di tempo di cinque minuti. Per utilizzare Auto Scaling in modo efficace, configura i trigger appropriati per applicazione, tipo di istanza e requisiti del servizio. È possibile effettuare il dimensionamento in base a diverse statistiche, tra cui latenza, I/O su disco, uso della CPU e numero di richieste.

Pianifica le azioni di Auto Scaling

Per ottimizzare l'uso delle istanze Amazon EC2 nell'ambiente durante i periodi con picchi di traffico prevedibili, configura il gruppo Auto Scaling per modificare il numero di istanze in base a una pianificazione. È possibile pianificare le modifiche alla configurazione del gruppo che si verificano giornalmente o settimanalmente, oppure pianificare solo le modifiche una tantum per la preparazione agli eventi di marketing che guideranno molto traffico verso il tuo sito.

Controllo dello stato di Auto Scaling

Auto Scaling monitora l'integrità di ciascuna istanza Amazon EC2 avviata. Se un'istanza viene terminata inaspettatamente, Auto Scaling rileva l'evento e avvia un'istanza di sostituzione. Per configurare il gruppo da utilizzare per il meccanismo di controllo dell'integrità del sistema di bilanciamento del carico, consulta Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk.

Argomenti

- Migrazione dell'ambiente Elastic Beanstalk ai modelli di avvio
- Supporto di istanze Spot per il tuo ambiente Elastic Beanstalk
- Trigger di Auto Scaling per il tuo ambiente Elastic Beanstalk
- Azioni di Auto Scaling pianificate per gli ambienti Elastic Beanstalk
- Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk

Migrazione dell'ambiente Elastic Beanstalk ai modelli di avvio

A partire dal 1° ottobre 2024, Amazon EC2 Auto Scaling non supporta più le configurazioni di avvio per nuovi account. Gli account creati prima di tale data potrebbero avere configurazioni di avvio.

Ti consigliamo di passare ai modelli di lancio per i seguenti vantaggi:

- Maggiore disponibilità per le tue applicazioni
- Migliore ottimizzazione dei carichi di lavoro nei gruppi di Auto Scaling
- Accesso alle più recenti funzionalità di EC2 e Auto Scaling

Per ulteriori informazioni, consulta le configurazioni di avvio di Auto Scaling nella Guida per l'utente di Amazon EC2 Auto Scaling.

Impostazioni delle opzioni per i modelli di avvio

Per migrare il tuo ambiente dalle configurazioni di avvio ai modelli di avvio, imposta una delle seguenti opzioni di configurazione:

- `RootVolumeType`opzione impostata su gp3. È possibile impostare questa opzione con la console o il namespace.
- `BlockDeviceMappings`l'opzione contiene gp3. È possibile impostare questa opzione con la console o il namespace.
- `DisableIMDSv1`opzione impostata su true. Si consiglia di impostare questa opzione utilizzando lo spazio dei nomi.
- `EnableSpot`opzione impostata su true. Per ulteriori informazioni, consulta Abilitazione delle istanze Spot.

> **Important**
>
> Dopo che un ambiente inizia a utilizzare i modelli di avvio, Elastic Beanstalk non torna alle configurazioni di avvio, anche se rimuovi le opzioni di configurazione che originariamente hanno attivato l'uso dei modelli di avvio.

Verifica se il tuo ambiente dispone di configurazioni di lancio o modelli di avvio

Puoi confermare se il tuo ambiente utilizza già modelli di avvio o se utilizza configurazioni di avvio esaminando il modello dello CloudFormation stack.

Per ispezionare il modello di stack del tuo ambiente CloudFormation

1. Apri la AWS CloudFormation console in https://console.aws.amazon.com /cloudformation.
2. Nella barra di navigazione nella parte superiore dello schermo, scegli la AWS regione in cui hai creato l'ambiente.
3. Nella pagina Stacks della CloudFormation console, controlla la colonna Descrizione.

   Individua e seleziona lo stack per l'ambiente Elastic Beanstalk. CloudFormation visualizza i dettagli dello stack per l'ambiente.
4. Nei dettagli dello stack, seleziona la scheda Modello.

   Utilizzando la ricerca nelle pagine del browser, puoi cercare launchtemplate o launchconfiguration nel testo del modello.

Per ulteriori informazioni, consulta Visualizza le informazioni sullo stack nella Guida per l'utente.AWS CloudFormation

Autorizzazioni richieste per i modelli di avvio

La politica predefinita del AWSElasticBeanstalkManagedUpdatesCustomerRolePolicyruolo del servizio gestito di Elastic Beanstalk fornisce le autorizzazioni necessarie per creare e gestire i modelli di avvio. Elastic Beanstalk deve gestire i modelli di avvio per completare molte operazioni ambientali, inclusa la creazione di ambienti.

Se colleghi policy personalizzate a un ruolo di servizio Elastic Beanstalk, verifica che il ruolo di servizio includa le seguenti autorizzazioni per la creazione di modelli di avvio. Queste autorizzazioni consentono a Elastic Beanstalk di creare e aggiornare correttamente gli ambienti nel tuo account:

Autorizzazioni richieste per i modelli di avvio Amazon EC2

- `ec2:RunInstances`
- `ec2:CreateLaunchTemplate`
- `ec2:CreateLaunchTemplateVersions`

- `ec2:DeleteLaunchTemplate`
- `ec2:DeleteLaunchTemplateVersions`
- `ec2:DescribeLaunchTemplate`
- `ec2:DescribeLaunchTemplateVersions`

Il seguente esempio di politica IAM include queste autorizzazioni.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersions",
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions",
        "ec2:DescribeLaunchTemplate",
        "ec2:DescribeLaunchTemplateVersions"
      ],
      "Resource": [
                "*"
            ]
     }
  ]
}
```

Per ulteriori informazioni, consultare Gestione dei ruoli del servizio Elastic Beanstalk e Gestione delle policy utente Elastic Beanstalk.

Ulteriori informazioni sui modelli di lancio

Per ulteriori informazioni sui modelli di lancio, consulta i modelli di lancio di Auto Scaling nella Guida per l'utente di Amazon EC2 Auto Scaling.

Per saperne di più sulla AWS transizione ai modelli di lancio e sui vantaggi che offrono, consulta Amazon EC2 Auto Scaling non aggiungerà più il supporto per le nuove funzionalità EC2 a Launch AWS Configurations nel Compute Blog.

> **Important**
>
> Non è necessario seguire la procedura a cui si fa riferimento in questo articolo del blog per trasferire un ambiente precedente ai modelli di lancio. Per migrare un ambiente Elastic Beanstalk esistente ai modelli di avvio, imposta una delle opzioni elencate in. Impostazioni delle opzioni per i modelli di avvio

Supporto di istanze Spot per il tuo ambiente Elastic Beanstalk

Questo argomento descrive le opzioni di configurazione disponibili per gestire la capacità e il bilanciamento del carico delle istanze Spot nell'ambiente Elastic Beanstalk. Fornisce inoltre dettagli ed esempi sui metodi che è possibile utilizzare per configurare queste opzioni. È possibile utilizzare la console Elastic Beanstalk, le opzioni di configurazione del namespace o l'EB AWS CLICLI per gestire le opzioni di configurazione.

Riduci al minimo le interruzioni delle istanze Spot con Capacity Rebal

Per ridurre al minimo l'impatto delle interruzioni delle istanze Spot sulla tua applicazione, puoi abilitare l'opzione Capacity Rebalancing inclusa in Amazon EC2 Auto Scaling.

> **Important**
>
> La richiesta di istanze Spot può variare significativamente da un momento all'altro e anche la disponibilità di istanze Spot può variare significativamente a seconda di quante istanze Amazon EC2 inutilizzate sono disponibili. È sempre possibile che l'istanza Spot venga interrotta.

Quando abiliti il ribilanciamento della capacità, EC2 tenta automaticamente di sostituire le istanze Spot in un gruppo di Auto Scaling prima che vengano interrotte. Per abilitare questa caratteristica, usa la console Elastic Beanstalk per configurare il gruppo Auto Scaling. In alternativa, puoi impostare l'opzione di configurazione Elastic Beanstalk `EnableCapacityRebalancing` a `true` nello spazio dei nomi aws:autoscaling:asg.

Per ulteriori informazioni, consulta Capacity Rebalancing nella Amazon EC2 Auto Scaling User Guide e Spot Instance Interruptions nella Amazon EC2 User Guide.

Tipi di istanze precedenti e supporto per istanze Spot

Alcuni AWS account meno recenti potrebbero fornire a Elastic Beanstalk tipi di istanze predefiniti che non supportano le istanze Spot. Se abiliti le richieste di istanze Spot e visualizzi l'errore Nessuno dei tipi di istanza che hai specificato supporta Spot, aggiorna la configurazione con i tipi di istanza che supportano le istanze Spot. Per scegliere i tipi di istanze Spot, utilizza Spot Instance Advisor.

Argomenti

- Abilitazione delle istanze Spot per il tuo ambiente
- Strategia di allocazione delle istanze Spot
- Gestione delle istanze On-Demand e delle istanze Spot
- Configurazione della capacità per l'ambiente Elastic Beanstalk

Abilitazione delle istanze Spot per il tuo ambiente

Per sfruttare le istanze Spot di Amazon EC2, imposta l'`EnableSpot`opzione per il tuo ambiente. Il gruppo Auto Scaling dell'ambiente combina quindi le opzioni di acquisto Amazon EC2 e mantiene un mix di istanze on demand e Spot.

Puoi utilizzare la console Elastic Beanstalk, le opzioni di configurazione del namespace o l'EB CLI per abilitare AWS CLIle richieste di istanze Spot per il tuo ambiente.

Prima di abilitare le istanze Spot per il tuo ambiente, acquisisci familiarità con le opzioni di configurazione di Auto Scaling, capacità e bilanciamento del carico disponibili. I requisiti dell'applicazione relativi al carico di lavoro, all'impatto delle interruzioni delle istanze e ai prezzi sono tutte considerazioni importanti nella pianificazione dell'attivazione delle istanze Spot.

Gli argomenti che seguono forniscono dettagli sulle opzioni di Auto Scaling e di gestione della capacità e su come il loro utilizzo combinato influisce sull'ambiente. Esistono procedure ed esempi di configurazioni per informare e guidare l'utente sulle varie opzioni e su come configurarle. Offriamo anche strumenti e funzionalità per aiutarti a gestire la configurazione e rispondere agli eventi. Puoi pianificare modifiche automatiche alla configurazione in base a periodi di traffico prevedibili, configurare i trigger per rispondere a fattori come il volume del traffico e configurare il monitoraggio e i controlli di integrità dell'Auto Scaling.

Per informazioni più dettagliate sulle istanze Spot, inclusa la spiegazione dei concetti chiave e delle best practice, consulta le istanze Spot nella Amazon EC2 User Guide.

> **Important**
>
> L'impostazione delle `EnableSpot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Strategia di allocazione delle istanze Spot

Puoi selezionare una qualsiasi delle strategie di allocazione elencate in questo argomento per il tuo ambiente Elastic Beanstalk. Utilizza la console Elastic Beanstalk, le opzioni di configurazione dello spazio dei nomi AWS CLIo, per impostare e configurare la strategia di allocazione delle istanze Spot e gli attributi correlati per il tuo ambiente.

Amazon EC2 applica una strategia di allocazione per gestire e fornire istanze Spot per il tuo ambiente. Ogni strategia di allocazione ottimizza le istanze allocate in base a come sono definite per gestire la capacità disponibile, il prezzo e la selezione dei tipi di istanze.

Amazon EC2 Auto Scaling fornisce le seguenti strategie di allocazione per le istanze Spot.

- Capacità ottimizzata (impostazione predefinita)
  - Richiede istanze Spot dal pool, con una capacità ottimale per il numero di istanze in fase di avvio.
  - Questa strategia è ideale per i carichi di lavoro in cui è necessario ridurre al minimo la possibilità di interruzione del servizio.
- Capacità di prezzo ottimizzata
  - Richiede istanze Spot dai pool che hanno il minor rischio di interruzione e il prezzo più basso possibile.
  - Questa è la scelta preferibile per la maggior parte dei carichi di lavoro Spot.
- Capacità ottimizzata con priorità
  - Richiede innanzitutto le istanze Spot in base alla disponibilità della capacità, rispettando al contempo la priorità del tipo di istanza da voi prescelto. Puoi fornire un elenco di tipi di istanze, ordinati per priorità, quando configuri le opzioni delle istanze Spot per Elastic Beanstalk.

  - Questa strategia è utile per i carichi di lavoro che richiedono un'interruzione minima del servizio e la prioritizzazione di un tipo di istanza specifico è importante.
- Prezzo più basso
  - Richiede istanze Spot dal pool con il prezzo più basso con istanze disponibili.
  - È importante prendere precauzioni quando si utilizza questa strategia, poiché considera solo il prezzo delle istanze e non la disponibilità della capacità, il che si tradurrà in tassi di interruzione elevati.

Per maggiori dettagli su ciascuna strategia di allocazione, consulta Strategie di allocazione per più tipi di istanze nella Guida per l'utente di Amazon EC2 Auto Scaling.

Per aiutarti a capire quale strategia di allocazione è più adatta a soddisfare i requisiti del tuo ambiente, consulta Scegli la strategia di allocazione Spot appropriata nella Guida per l'utente di Amazon EC2.

Gestione delle istanze On-Demand e delle istanze Spot

È possibile avviare e scalare automaticamente un parco di istanze on demand e istanze spot all’interno di un singolo gruppo con un singolo gruppo Auto Scaling. Le seguenti opzioni possono essere utilizzate insieme per configurare il modo in cui il servizio Auto Scaling gestisce le istanze Spot e le istanze On-Demand nel proprio ambiente.

Puoi configurare queste opzioni per il tuo ambiente utilizzando la console Elastic Beanstalk, le opzioni di configurazione dei namespace o l'EB CLI. AWS CLI

Queste opzioni fanno parte dello spazio dei nomi aws:ec2:instances:

- `EnableSpot`- Se impostata su `true` questa impostazione abilita le richieste di istanze Spot per il tuo ambiente.
- `SpotFleetOnDemandBase`- Imposta il numero minimo di istanze On-Demand che il gruppo Auto Scaling fornisce prima di prendere in considerazione le istanze Spot man mano che l'ambiente si espande.
- `SpotFleetOnDemandAboveBasePercentage`- La percentuale di istanze On-Demand come parte della capacità aggiuntiva fornita dal gruppo Auto Scaling oltre alle istanze. `SpotOnDemandBase`

Le opzioni elencate in precedenza sono correlate alle seguenti opzioni nello spazio dei nomi aws:autoscaling:asg:

- `MinSize`- Il numero minimo di istanze che desideri inserire nel gruppo Auto Scaling.
- `MaxSize`- Il numero massimo di istanze che desideri nel tuo gruppo Auto Scaling.

> **Important**
>
> L'impostazione delle `EnableSpot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Applicazione di entrambi i set di opzioni dello spazio dei nomi

I punti seguenti descrivono in che modo la combinazione di queste impostazioni di opzione influisce sulla scalabilità dell'ambiente.

- Solo `MinSize` determina la capacità iniziale dell'ambiente, ovvero il numero di istanze che desideri eseguire al minimo.
- `SpotFleetOnDemandBase` non influisce sulla capacità iniziale. Quando Spot è abilitato, questa opzione determina quante istanze On-Demand vengono fornite prima che vengano prese in considerazione le istanze Spot.
- Prendilo in considerazione quando `SpotFleetOnDemandBase` è minore di `MinSize`. Otterrai comunque esattamente `MinSize` istanze, come capacità iniziale. Almeno `SpotFleetOnDemandBase` di queste devono essere istanze on demand.
- Prendilo in considerazione quando `SpotFleetOnDemandBase` è maggiore di `MinSize`. Via via che l'ambiente si ridimensiona, avrai la garanzia di ottenere un numero aggiuntivo di istanze uguali alla differenza tra i due valori. In altre parole, otterrai sicuramente almeno altre `(SpotFleetOnDemandBase - MinSize)` istanze on demand, prima di soddisfare il requisito `SpotFleetOnDemandBase`.

Ambienti di istanza singola

Negli ambienti di produzione, le istanze Spot sono particolarmente utili come parte di un ambiente dimensionabile con bilanciamento del carico. Si sconsiglia di utilizzare Spot in un ambiente a istanza singola. Se le istanze Spot non sono disponibili, è possibile perdere l'intera capacità (una singola istanza) dell'ambiente. È comunque possibile utilizzare un'istanza Spot in un ambiente a istanza singola per lo sviluppo o il test. Quando lo fai, assicurati di impostare entrambi `SpotFleetOnDemandBase` e `SpotFleetOnDemandAboveBasePercentage` su zero. Tutte le altre impostazioni determinano un'istanza su richiesta.

Esempi di impostazioni delle opzioni di ridimensionamento

Negli esempi seguenti vengono illustrati diversi scenari di impostazione delle varie opzioni di ridimensionamento. Tutti gli esempi presuppongono un ambiente con bilanciamento del carico con richieste di istanze Spot abilitate.

Example 1: On-Demand e Spot come parte della capacità iniziale

Impostazioni opzioni

Opzione	Spazio dei nomi	Valore
`MinSize`	`aws:autoscaling:asg`	`10`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

In questo esempio, l'ambiente inizia con dieci istanze, di cui sette sono su richiesta (quattro base e 50% dei sei sopra base) e tre sono Spot. L'ambiente può scalare fino a 24 istanze. Man mano che si ridimensiona, la parte di On-Demand nella parte della flotta al di sopra delle quattro istanze di base su richiesta è mantenuta al 50%, fino a un massimo di 24 istanze complessive, di cui 14 sono On-Demand (quattro base e 50% delle 20 sopra base) e dieci sono Spot.

Example 2: Tutta la capacità iniziale su richiesta

Impostazioni opzioni

Opzione	Spazio dei nomi	Valore
`MinSize`	`aws:autoscaling:asg`	`4`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

In questo esempio, l'ambiente inizia con quattro istanze, tutte su richiesta. L'ambiente può scalare fino a 24 istanze. Man mano che si ridimensiona, la parte di On-Demand nella parte della flotta al di sopra delle quattro istanze di base su richiesta è mantenuta al 50%, fino a un massimo di 24 istanze complessive, di cui 14 sono On-Demand (quattro base e 50% delle 20 sopra base) e dieci sono Spot.

Example 3: Base aggiuntiva su richiesta oltre la capacità iniziale

Impostazioni opzioni

Opzione	Spazio dei nomi	Valore
`MinSize`	`aws:autoscaling:asg`	`3`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

In questo esempio, l'ambiente inizia con tre istanze, tutte su richiesta. L'ambiente può scalare fino a 24 istanze. La prima istanza aggiuntiva sopra le tre iniziali è su richiesta, per completare le quattro istanze su richiesta di base. Man mano che si ridimensiona ulteriormente, la parte di On-Demand

nella parte della flotta al di sopra delle quattro istanze di base su richiesta è mantenuta al 50%, fino a un massimo di 24 istanze complessive, di cui 14 sono On-Demand (quattro base e 50% dei 20 sopra base) e dieci sono Spot.

Configurazione della capacità per l'ambiente Elastic Beanstalk

Questo argomento descrive i diversi approcci per configurare la capacità di Auto Scaling per l'ambiente Elastic Beanstalk. Puoi utilizzare la console Elastic Beanstalk, l'EB AWS CLI CLI, o le opzioni dello spazio dei nomi.

> **Important**
>
> L'impostazione delle `EnableSpot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Configurazione con la console

Puoi configurare la gestione della capacità di un gruppo Auto Scaling modificando Capacity nella pagina di configurazione dell'ambiente nella console Elastic Beanstalk.

Per configurare la capacità del gruppo Auto Scaling nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Auto Scaling group (Gruppo Auto Scaling), configurare le impostazioni seguenti.
   - Environment type (Tipo di ambiente): selezionare Load balanced (Carico bilanciato).

- Min instances (Numero minimo di istanze): numero minimo di istanze EC2 che il gruppo deve contenere in qualsiasi momento. Il gruppo inizia con il numero minimo di istanze e ne aggiunge altre quando viene soddisfatta la condizione del trigger di scalabilità verticale.
- Max instances (Numero massimo di istanze): numero massimo di istanze EC2 che il gruppo deve contenere in qualsiasi momento.

> **Note**
>
> Se utilizzi gli aggiornamenti in sequenza, accertati che il numero massimo di istanze sia superiore all'impostazione Minimum instances in service (Istanze minime in servizio) per gli aggiornamenti in sequenza.

- Composizione del parco istanze: per impostazione predefinita è Istanze on demand. Per abilitare le richieste di istanze Spot, seleziona Opzioni e istanze d'acquisto combinate.

> **Important**
>
> L'impostazione delle `EnableSpot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Se sceglierai di abilitare le richieste di istanze Spot, verranno abilitate le seguenti opzioni:

- Strategia di allocazione spot: determina il metodo utilizzato per gestire e fornire le istanze Spot nell'ambiente in uso, in base alla capacità disponibile, al prezzo e alla selezione dei tipi di istanze. Scegli tra Capacità ottimizzata (impostazione predefinita), Ottimizzazione della capacità di prezzo, Capacità ottimizzata con priorità o Prezzo più basso. Per una descrizione di ogni strategia di allocazione e ulteriori informazioni, vedere. the section called “Strategia di allocazione spot”
- Prezzo spot massimo: per consigli sulle opzioni di prezzo massimo per le istanze Spot, consulta la cronologia dei prezzi delle istanze Spot nella Guida per l'utente di Amazon EC2.

- On demand di base: il numero minimo di istanze on demand di cui il gruppo Auto Scaling effettua il provisioning prima di considerare le istanze Spot durante il dimensionamento dell'ambiente.
- On demand sopra la base: la percentuale di istanze on demand come parte della capacità aggiuntiva di cui il gruppo Auto Scaling effettua il provisioning oltre alle istanze on demand di base.

> **Note**
>
> Le opzioni On-Demand base (On demand di base) e On-Demand above base (On demand sopra la base) sono correlate alle opzioni di istanze Min e Max elencate in precedenza. Per ulteriori informazioni su queste opzioni ed esempi, consulta the section called “Supporto per istanze Spot”.

- Ribilanciamento della capacità: questa opzione è rilevante solo quando nel gruppo Auto Scaling è presente almeno un'istanza Spot. Quando questa funzionalità è abilitata, EC2 tenta automaticamente di sostituire le istanze Spot nel gruppo Auto Scaling prima che vengano interrotte, riducendo al minimo le interruzioni delle istanze Spot nelle applicazioni. Per ulteriori informazioni, consulta Capacity Rebalancing nella Guida per l'utente di Amazon EC2 Auto Scaling

- Architettura: l'architettura del processore per le tue istanze EC2. L'architettura del processore determina i tipi di istanze EC2 che saranno disponibili nel campo successivo.
- Tipi di istanze: i tipi di istanze Amazon EC2 avviate per eseguire l'applicazione. Per informazioni dettagliate, vedi the section called “Tipi di istanza”.
- AMI ID (ID istanza AMI): l'immagine della macchina utilizzata da Elastic Beanstalk per avviare istanze Amazon EC2 nell'ambiente. Per informazioni dettagliate, vedi the section called “ID AMI”.
- Availability Zones (Zone di disponibilità): scegli il numero di zone di disponibilità in cui distribuire le istanze dell'ambiente. Per impostazione predefinita, il gruppo Auto Scaling avvia le istanze in modo uniforme in tutte le zone utilizzabili. Per concentrare le tue istanze in un numero minore di zone, scegli il numero di zone da utilizzare. Per gli ambienti di produzione, utilizza almeno due zone per essere sicuro che la tua applicazione sia disponibile in caso una zona di disponibilità fosse inutilizzabile.
- Placement (Posizionamento) (opzionale): scegli le zone di disponibilità da utilizzare. Utilizza questa impostazione se le istanze devono connettersi a risorse in determinate zone o se hai acquistato le istanze riservate, che sono specifiche per zona. Se avvii l'ambiente in un VPC

personalizzato, non è possibile configurare questa opzione. In un VPC personalizzato, scegli le zone di disponibilità per le sottoreti che assegni al tuo ambiente.

- Scaling cooldown (Raffreddamento del dimensionamento): tempo di attesa espresso in secondi per avviare o terminare le istanze dopo il dimensionamento, prima di continuare a valutare i trigger. Per ulteriori informazioni, consulta l'articolo relativo ai periodi di attesa del dimensionamento.

6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurazione tramite opzioni di namespace

In Elastic Beanstalk sono disponibili opzioni di configurazione delle impostazioni di Auto Scaling in due spazi dei nomi: `aws:autoscaling:asg` e `aws:ec2:instances`.

Spazio dei nomi aws:autoscaling:asg

Lo spazio dei nomi `aws:autoscaling:asg` fornisce opzioni per la scalabilità complessiva e la disponibilità.

Nell'esempio di file di configurazione riportato di seguito viene configurato il gruppo Auto Scaling in modo da utilizzare da due a quattro istanze, zone di disponibilità specifiche e un periodo di attesa di 12 minuti (720 secondi). Consente il ribilanciamento della capacità per le istanze Spot. Questa `EnableCapacityRebalancing` opzione ha effetto solo se `EnableSpot` è impostata su `true` nello spazio dei `aws:ec2:instances`nomi, come mostrato nell'esempio del file di configurazione che segue questa.

```
option_settings:
  aws:autoscaling:asg:
    Availability Zones: Any
    Cooldown: '720'
    Custom Availability Zones: 'us-west-2a,us-west-2b'
    MaxSize: '4'
    MinSize: '2'
    EnableCapacityRebalancing: true
```

Spazio dei nomi aws:ec2:instances

> **Note**
>
> Quando si aggiorna la configurazione dell'ambiente e si rimuove uno o più tipi di istanza dall'opzione `InstanceTypes`, Elastic Beanstalk termina tutte le istanze Amazon EC2 in

esecuzione su uno qualsiasi dei tipi di istanza rimossi. Il gruppo Auto Scaling dell'ambiente avvia quindi nuove istanze, se necessario per completare la capacità desiderata, utilizzando i tipi di istanza specificati correnti.

Lo spazio dei nomi `aws:ec2:instances` offre opzioni relative alle istanze dell'ambiente, inclusa la gestione dell'istanza Spot. Integra `aws:autoscaling:launchconfiguration` e `aws:autoscaling:asg`.

L'esempio di file di configurazione seguente configura il gruppo Auto Scaling per abilitare le richieste di istanze Spot per l'ambiente. Designa tre possibili tipi di istanza che possono essere utilizzati. Allmeno un'istanza on demand viene utilizzata per la capacità di base, mentre un 33% sostenuto delle istanze on demand viene utilizzato per tutte le capacità aggiuntive.

La configurazione imposta la strategia di allocazione spot su. `capacity-optimized-prioritized` Questa particolare strategia di allocazione dà la priorità agli avvii delle istanze dal pool in base all'ordine dei tipi di istanze specificati nell'opzione. `InstanceTypes` Se non `SpotAllocationStrategy` è specificato, il valore predefinito è. `capacity-optimized`

```
option_settings:
  aws:ec2:instances:
    EnableSpot: true
    InstanceTypes: 't2.micro,t3.micro,t3.small'
    SpotAllocationStrategy: capacity-optimized-prioritized
    SpotFleetOnDemandBase: '1'
    SpotFleetOnDemandAboveBasePercentage: '33'
```

Per scegliere i tipi di istanze Spot, utilizza Spot Instance Advisor.

> **Important**
>
> L'impostazione delle `EnableSpot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Configurazione mediante AWS CLI

Questa sezione fornisce esempi di come utilizzare il comando AWS CLI create-environment per configurare l'ambiente con le opzioni Auto Scaling e Capacity descritte in queste sezioni. Noterai che le impostazioni dello spazio dei nomi per `aws:autoscaling:asg`e `aws:ec2:instances`, come descritto nella precedente sezione sulle opzioni di configurazione dello spazio dei nomi, sono configurate anche con questo esempio.

L'interfaccia a riga di AWS comando fornisce comandi per creare e configurare ambienti Elastic Beanstalk. Con l'`--option-settings`opzione, puoi passare le opzioni dello spazio dei nomi supportate da Elastic Beanstalk. Ciò significa che le opzioni di configurazione dello spazio dei nomi descritte in precedenza possono essere passate ai AWS CLI comandi applicabili per configurare l'ambiente Elastic Beanstalk.

> **Note**
>
> Puoi anche utilizzare il comando update-environment con `--option-settings` per aggiungere o aggiornare le opzioni dello spazio dei nomi. Se devi rimuovere qualsiasi opzione del namespace dal tuo ambiente, usa il comando with. update-environment `--options-to-remove`

L'esempio seguente crea un nuovo ambiente. Fate riferimento all'argomento precedente opzioni di configurazione dello spazio dei nomi per ulteriori informazioni sulle opzioni che vengono passate.

La prima opzione elencata, `IamInstanceProfile` nel aws:autoscaling:launchconfiguration namespace, è il profilo dell'istanza Elastic Beanstalk. È necessario quando si crea un nuovo ambiente.

Example— ambiente di creazione con opzioni Auto Scaling (opzioni dello spazio dei nomi in linea)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v4.3.0 running Python 3.12" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role
Namespace=aws:autoscaling:asg,OptionName=Availability Zones,Value=Any \
```

```
Namespace=aws:autoscaling:asg,OptionName=Cooldown,Value=720 \
Namespace=aws:autoscaling:asg,OptionName=Custom Availability Zones,Value=us-west-2a,us-west-2b \
Namespace=aws:autoscaling:asg,OptionName=MaxSize,Value=4 \
Namespace=aws:autoscaling:asg,OptionName=MinSize,Value=2 \
Namespace=aws:autoscaling:asg,OptionName=EnableCapacityRebalancing,Value=true \
Namespace=aws:ec2:instances,OptionName=EnableSpot,Value=true \
Namespace=aws:ec2:instances,OptionName=InstanceTypes,Value=t2.micro,t3.micro,t3.small \
Namespace=aws:ec2:instances,OptionName=SpotAllocationStrategy,Value=capacity-optimized-prioritized \
Namespace=aws:ec2:instances,OptionName=SpotFleetOnDemandBase,Value=1 \
Namespace=aws:ec2:instances,OptionName=SpotFleetOnDemandAboveBasePercentage,Value=33
```

> **Important**
>
> L'impostazione delle `EnableSpot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga.

Example—create-environment con opzioni Auto Scaling (opzioni dello spazio dei nomi nel file) options.json

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 v4.3.0 running Python 3.12"
--option-settings file://options.json
```

Example

```
### example options.json ###
[
    {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": "aws-elasticbeanstalk-ec2-role"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "Availability Zones",
        "Value": "Any"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "Cooldown",
        "Value": "720"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "Custom Availability Zones",
        "Value": "us-west-2a,us-west-2b"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MaxSize",
        "Value": "4"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MinSize",
        "Value": "2"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "EnableCapacityRebalancing",
        "Value": "true"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "EnableSpot",
        "Value": "true"
    },
```

```
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "InstanceTypes",
        "Value": "t2.micro,t3.micro,t3.small"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "SpotAllocationStrategy",
        "Value": "capacity-optimized-prioritized"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "SpotFleetOnDemandBase",
        "Value": "1"
    },
    {
        "Namespace": "aws:ec2:instances",
        "OptionName": "SpotFleetOnDemandAboveBasePercentage",
        "Value": "33"
    }
]
```

Configurazione tramite l'EB CLI

Quando si crea un ambiente utilizzando il comando eb create, è possibile specificare alcune opzioni relative al gruppo Auto Scaling dell'ambiente. Queste sono alcune delle opzioni que consentono di controllare la capacità dell'ambiente.

`--single`

Crea l'ambiente con un'istanza Amazon EC2 e nessun load balancer. Se non utilizzi questa opzione, un load balancer viene aggiunto all'ambiente che è stato creato.

`--enable-spot`

Abilita le richieste di istanze Spot per l'ambiente in uso.

> **Important**
>
> L'impostazione delle `enable-spot` opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessarie le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate

anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.

Le seguenti opzioni per il comando eb create può essere usato solo con `--enable-spot`.

`--instance-types`

Elenca i tipi di istanza Amazon EC2 che si desidera utilizzare nell'ambiente.

`--spot-max-price`

Il prezzo massimo orario per unità che si è disposti a pagare per un'istanza Spot in dollari americani. Per consigli sulle opzioni di prezzo massimo per le istanze Spot, consulta la cronologia dei prezzi delle istanze Spot nella Guida per l'utente di Amazon EC2.

`--on-demand-base-capacity`

Il numero minimo di istanze on demand di cui il gruppo Auto Scaling effettua il provisioning prima di considerare le istanze Spot durante la scalabilità dell'ambiente.

`--on-demand-above-base-capacity`

La percentuale di istanze on demand come parte della capacità aggiuntiva che il gruppo Auto Scaling fornisce in aggiunta al numero di istanze specificato dall'opzione `--on-demand-base-capacity`.

L'esempio seguente crea un ambiente e configura il gruppo Auto Scaling per abilitare le richieste dell'istanza Spot per il nuovo ambiente, con tre possibili tipi di istanza da utilizzare. Per questo esempio, esistono tre possibili tipi di istanza.

```
$ eb create --enable-spot --instance-types "t2.micro,t3.micro,t3.small"
```

Important

Esiste un'altra opzione con nome simile che si chiama `--instance-type` (senza "s") che la CLI EB riconosce solo durante l'elaborazione delle istanze on demand. Non utilizzare `--instance-type` (senza "s") con l'opzione `--enable-spot`. Se lo fai, la CLI EB lo ignora. Invece, utilizza `--instance-types` (con "s") con l'opzione `--enable-spot`.

Trigger di Auto Scaling per il tuo ambiente Elastic Beanstalk

Il gruppo Auto Scaling nel tuo ambiente Elastic Beanstalk utilizza due CloudWatch allarmi Amazon per attivare le operazioni di scalabilità. I trigger predefiniti eseguono il dimensionamento quando la media del traffico di rete in uscita da ciascuna istanza è superiore a 6 MB o inferiore a 2 MB per un intervallo di tempo di cinque minuti. Per utilizzare Amazon EC2 Auto Scaling in modo efficace, configura i trigger appropriati per la tua applicazione, il tipo di istanza e i requisiti di servizio. È possibile effettuare il dimensionamento in base a diverse statistiche, tra cui latenza, I/O su disco, uso della CPU e numero di richieste.

Per ulteriori informazioni su CloudWatch metriche e allarmi, consulta Amazon CloudWatch Concepts nella Amazon CloudWatch User Guide.

Configurazione dei trigger di Auto Scaling

Puoi configurare i trigger che consentono di regolare il numero di istanze nel gruppo Auto Scaling del tuo ambiente nella console Elastic Beanstalk.

Per configurare i trigger nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Scaling triggers (Trigger di dimensionamento), configurare le impostazioni seguenti:
   - Metric (Parametro): parametro utilizzato per il trigger di Auto Scaling.
   - Statistic (Statistica): calcolo statistico che deve essere utilizzato dal trigger, ad esempio `Average`.
   - Unit (Unità): unità di misura del parametro di trigger, ad esempio Bytes (Byte).
   - Periodo: specifica la frequenza con cui Amazon CloudWatch misura le metriche relative al trigger.
   - Breach duration (Durata utilizzo fuori limite): intervallo di tempo, espresso in minuti, in cui un parametro può trovarsi al di fuori della soglia massima e minima prima di attivare un'operazione di dimensionamento.

- Upper threshold (Soglia massima): se il parametro supera questo valore per la durata dell'utilizzo fuori limite impostata, viene attivata un'operazione di dimensionamento.
- Incremento di scalabilità: il numero di EC2 istanze Amazon da aggiungere quando si esegue un'attività di scalabilità.
- Lower threshold (Soglia minima): se il parametro scende al di sotto di questo valore per la durata dell'utilizzo fuori limite impostata, viene attivata un'operazione di dimensionamento.
- Incremento di scalabilità: il numero di EC2 istanze Amazon da rimuovere durante l'esecuzione di un'attività di scalabilità.

6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Spazio dei nomi aws:autoscaling:trigger

In Elastic Beanstalk sono disponibili opzioni di configurazione delle impostazioni di Auto Scaling nello spazio dei nomi `aws:autoscaling:trigger`. Le impostazioni in questo namespace sono organizzate in base alla risorsa a cui si applicano.

```
option_settings:
  AWSEBAutoScalingScaleDownPolicy.aws:autoscaling:trigger:
    LowerBreachScaleIncrement: '-1'
  AWSEBAutoScalingScaleUpPolicy.aws:autoscaling:trigger:
    UpperBreachScaleIncrement: '1'
  AWSEBCloudwatchAlarmHigh.aws:autoscaling:trigger:
    UpperThreshold: '6000000'
  AWSEBCloudwatchAlarmLow.aws:autoscaling:trigger:
    BreachDuration: '5'
    EvaluationPeriods: '1'
    LowerThreshold: '2000000'
    MeasureName: NetworkOut
    Period: '5'
    Statistic: Average
    Unit: Bytes
```

Azioni di Auto Scaling pianificate per gli ambienti Elastic Beanstalk

Per ottimizzare l'uso delle EC2 istanze Amazon da parte del tuo ambiente durante periodi prevedibili di picco di traffico, configura il tuo gruppo Amazon Auto EC2 Scaling per modificare il numero di istanze in base a una pianificazione. Puoi configurare l'ambiente con un'operazione ricorrente per aumentare quotidianamente il dimensionamento nel corso della mattinata e ridurlo la sera

quando il traffico è ridotto. Ad esempio, se è previsto un evento di marketing che genererà traffico nel tuo sito per un periodo di tempo limitato, puoi pianificare un evento singolo per aumentare il dimensionamento all'inizio dell'evento di marketing e uno per ridurre il dimensionamento al termine dell'evento.

Puoi definire fino a 120 operazioni pianificate attive per ciascun ambiente. Elastic Beanstalk conserva inoltre fino a 150 operazioni pianificate scadute, che puoi riutilizzare aggiornando le impostazioni.

Configurazione delle operazioni pianificate

Nella console Elastic Beanstalk puoi creare operazioni pianificate per il gruppo Auto Scaling dell'ambiente.

Per configurare le operazioni pianificate nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Time-based Scaling (Dimensionamento a tempo), scegliere Add scheduled action (Aggiungi azione pianificata).
6. Inserire le seguenti impostazioni per le azioni programmate:
   - Name (Nome): specificare un nome univoco contenente un massimo di 255 caratteri alfanumerici senza spazi.
   - Instances (Istanze): scegliere il numero minimo e massimo di istanze da applicare al gruppo Auto Scaling.
   - Desired capacity (Capacità desiderata) (facoltativa): impostare la capacità iniziale desiderata per il gruppo Auto Scaling. Dopo l'applicazione dell'azione programmata, i trigger regolano la capacità desiderata in base alle relative impostazioni.
   - Occurrence (Frequenza): scegliere Recurring (Ricorrente) per ripetere l'operazione di dimensionamento in base a una pianificazione.
   - Start time (Ora di inizio): per le operazioni singole, scegliere la data e l'ora di esecuzione dell'operazione.

Per le operazioni ricorrenti, l'ora di inizio è facoltativa. Specificarlo per scegliere il primo orario di esecuzione dell'operazione. Trascorso questo tempo, l'operazione ricorre in base all'espressione Recurrence (Ricorrenza) .

- Recurrence (Ricorrenza): utilizzare un'espressione Cron per specificare la frequenza di occorrenza dell'operazione pianificata. Ad esempio, `30 6 * * 2` esegue l'operazione ogni martedì alle 6:30 UTC.
- End time (Ora di fine) (opzionale): opzionale per le operazioni ricorrenti. Se specificato, l'operazione ricorre in base all'espressione Recurrence (Ricorrenza) e non viene eseguita nuovamente dopo questo periodo di tempo.

  Quando un'operazione pianificata termina, le impostazioni precedenti di Auto Scaling non vengono ripristinate automaticamente. Configurare una seconda operazione pianificata per ripristinare le impostazioni originali di Auto Scaling come necessario.

7. Scegliere Aggiungi.
8. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

> **Note**
>
> Le operazioni programmate non saranno salvate finché non verranno applicate.

Spazio dei nomi aws:autoscaling:scheduledaction

Se devi configurare un numero elevato di operazioni pianificate, puoi utilizzare i file di configurazione o l'API Elastic Beanstalk per applicare le modifiche delle opzioni di configurazione da un file YAML o JSON. Questi metodi consentono inoltre di accedere all'opzione `Suspend` per disattivare temporaneamente un'operazione pianificata ricorrente.

> **Note**
>
> Se utilizzi le opzioni di configurazione delle operazioni pianificate al di fuori della console, utilizza il formato dell'ora ISO 8601 per specificare l'ora di inizio e quella di fine in UTC. Ad esempio, 2015-04-28T04:07:02Z. Per ulteriori informazioni sul formato dell'ora ISO 8601, consulta la pagina relativa ai formati di data e ora. Le date devono essere univoche per tutte le operazioni pianificate.

In Elastic Beanstalk sono disponibili opzioni di configurazione per le impostazioni delle operazioni pianificate nello spazio dei nomi `aws:autoscaling:scheduledaction`. Utilizza il campo `resource_name` per specificare il nome dell'operazione pianificata.

Example S cheduled-scale-up-specific -time-long.config

Questo file di configurazione indica a Elastic Beanstalk di aumentare il dimensionamento da cinque a dieci istanze in data 2015-12-12T00:00:00Z.

```
option_settings:
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: MinSize
    value: '5'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: MaxSize
    value: '10'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: DesiredCapacity
    value: '5'
  - namespace: aws:autoscaling:scheduledaction
    resource_name: ScheduledScaleUpSpecificTime
    option_name: StartTime
    value: '2015-12-12T00:00:00Z'
```

Example S cheduled-scale-up-specific -time.config

Per utilizzare la sintassi abbreviata con la CLI EB o con i file di configurazione, anteponi il nome della risorsa allo spazio dei nomi.

```
option_settings:
  ScheduledScaleUpSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '5'
    MaxSize: '10'
    DesiredCapacity: '5'
    StartTime: '2015-12-12T00:00:00Z'
```

Example S cheduled-scale-down-specific -time.config

Questo file di configurazione indica a Elastic Beanstalk di ridurre il dimensionamento in data 2015-12-12T07:00:00Z.

```
option_settings:
  ScheduledScaleDownSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '1'
    MaxSize: '1'
    DesiredCapacity: '1'
    StartTime: '2015-12-12T07:00:00Z'
```

Example cheduled-periodic-scale-upS .config

Questo file di configurazione indica a Elastic Beanstalk di aumentare il dimensionamento ogni giorno alle 9:00. Questa operazione è pianificata per iniziare il 14 maggio 2015 e terminare il 12 gennaio 2016.

```
option_settings:
  ScheduledPeriodicScaleUp.aws:autoscaling:scheduledaction:
    MinSize: '5'
    MaxSize: '10'
    DesiredCapacity: '5'
    StartTime: '2015-05-14T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
    Recurrence: 0 9 * * *
```

Example S .config cheduled-periodic-scale-down

Questo file di configurazione indica a Elastic Beanstalk di ridurre il dimensionamento fino a zero istanze in esecuzione ogni giorno alle 18:00. Se sai che la tua applicazione è per lo più inattiva al di fuori delle ore lavorative, puoi creare un'azione pianificata analoga. Se l'applicazione non deve essere attiva al di fuori delle ore lavorative, modifica `MaxSize` in `0`.

```
option_settings:
  ScheduledPeriodicScaleDown.aws:autoscaling:scheduledaction:
    MinSize: '0'
    MaxSize: '1'
    DesiredCapacity: '0'
    StartTime: '2015-05-14T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
```

```
    Recurrence: 0 18 * * *
```

Example S .config cheduled-weekend-scale-down

Questo file di configurazione indica a Elastic Beanstalk di ridurre il dimensionamento ogni giorno alle 6:00. Se sai già che l'applicazione non riceve molto traffico durante i weekend, puoi creare un'operazione pianificata simile.

```
option_settings:
  ScheduledWeekendScaleDown.aws:autoscaling:scheduledaction:
    MinSize: '1'
    MaxSize: '4'
    DesiredCapacity: '1'
    StartTime: '2015-12-12T07:00:00Z'
    EndTime: '2016-01-12T07:00:00Z'
    Recurrence: 0 18 * * 5
```

Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk

Amazon EC2 Auto Scaling monitora lo stato di ogni istanza Amazon Elastic Compute Cloud (Amazon EC2) che lancia. Se un'istanza viene terminata inaspettatamente, Auto Scaling rileva l'evento e avvia un'istanza di sostituzione. Per impostazione predefinita, il gruppo Auto Scaling creato per il tuo ambiente utilizza i controlli di EC2 stato di Amazon. Se un'istanza nel tuo ambiente non supera un controllo EC2 dello stato di Amazon, Auto Scaling la disattiva e la sostituisce.

I controlli EC2 dello stato di Amazon riguardano solo lo stato dell'istanza, non lo stato dell'applicazione, del server o di qualsiasi contenitore Docker in esecuzione sull'istanza. Se si verifica un arresto anomalo dell'applicazione, ma l'istanza su cui viene eseguita è ancora funzionante, potrebbe essere eliminata dal sistema di bilanciamento del carico, ma Auto Scaling non la sostituirà automaticamente. Il comportamento predefinito è ideale per la risoluzione dei problemi. Se Auto Scaling sostituisce l'istanza subito dopo l'arresto anomalo dell'applicazione, è possibile che l'utente non si accorga del problema, anche se l'arresto anomalo si è verificato subito dopo l'avvio.

Se desideri che Auto Scaling sostituisca le istanze la cui applicazione ha smesso di rispondere, puoi utilizzare un file di configurazione per configurare il gruppo Auto Scaling affinché utilizzi i controlli dello stato di Elastic Load Balancing. L'esempio seguente imposta il gruppo in modo che utilizzi i controlli di integrità del load balancer, oltre al controllo EC2 dello stato di Amazon, per determinare lo stato di un'istanza.

Example.ebextensions/autoscaling.config

```
Resources:
  AWSEBAutoScalingGroup:
    Type: "AWS::AutoScaling::AutoScalingGroup"
    Properties:
      HealthCheckType: ELB
      HealthCheckGracePeriod: 300
```

Per ulteriori informazioni sulle `HealthCheckGracePeriod` proprietà `HealthCheckType` e, consulta AWS::AutoScaling::AutoScalingGroup in the AWS CloudFormation User Guide e Health Checks for Auto Scaling Instances nella Amazon EC2 Auto Scaling User Guide.

Per impostazione predefinita, il controllo dello stato di Elastic Load Balancing è configurato in modo da tentare una connessione TCP all'istanza tramite la porta 80. Questo conferma che il server Web in esecuzione sull'istanza accetta le connessioni. Puoi comunque personalizzare il controllo dello stato del sistema di bilanciamento del carico per verificare che anche l'applicazione, e non solo il server Web, sia in buono stato. La configurazione del periodo di grazia imposta il numero di secondi in base ai quali un'istanza può fallire il controllo dello stato senza essere terminata e sostituita. Le istanze possono recuperare dopo essere state eliminate dal sistema di bilanciamento del carico, per cui dai al'istanza una quantità di tempo adeguata per la tua applicazione.

Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk

Un load balancer distribuisce il traffico tra le istanze dell'ambiente. Quando abiliti il bilanciamento del carico, AWS Elastic Beanstalk crea un sistema di bilanciamento del carico Elastic Load Balancing dedicato al tuo ambiente. Elastic Beanstalk gestisce completamente questo load balancer dalle impostazioni di sicurezza e alla terminazione del load balancer quando termini l'ambiente.

In alternativa puoi scegliere di condividere un load balancer in diversi ambienti Elastic Beanstalk. Con un load balancer condiviso, è possibile risparmiare sui costi operativi evitando di avere un load balancer dedicato per ogni ambiente. Si assume inoltre una maggiore responsabilità di gestione per il load balancer condiviso utilizzato dagli ambienti.

Elastic Load Balancing supporta i seguenti tipi di load balancer:

- Classic Load Balancer: il load balancer della generazione precedente. Instrada il traffico di richieste HTTP, HTTPS o TCP verso porte diverse delle istanze dell'ambiente.

- Application Load Balancer: un load balancer a livello di applicazione. Instrada il traffico di richieste HTTP o HTTPS verso diverse porte delle istanze dell'ambiente in base al percorso della richiesta.
- Network Load Balancer: un load balancer a livello di rete. Indirizza il traffico delle richieste TCP o TLS verso porte diverse sulle istanze di ambiente. Supporta i controlli dello stato attivi e passivi.

Elastic Beanstalk supporta tutti e tre i tipi di load balancer. Nella tabella seguente vengono illustrati i tipi che è possibile utilizzare con i due modelli di utilizzo:

Nuovo tipo di load balancer	Dedicato	Condiviso
Classic Load Balancer	✓ Sì	× No
Application Load Balancer	✓ Sì	✓ Sì
Network Load Balancer	✓ Sì	× No

> **Note**
>
> L'opzione Classic Load Balancer (CLB) è disattivata nella procedura guidata della console Crea ambiente. Se disponi di un ambiente esistente configurato con Classic Load Balancer, puoi crearne uno nuovo clonando l'ambiente esistente utilizzando la console Elastic Beanstalk o la CLI EB. È inoltre possibile utilizzare l'EB CLI o AWS CLIcreare un nuovo ambiente configurato con un Classic Load Balancer. Questi strumenti da riga di comando creeranno un nuovo ambiente con un CLB anche se non ne esiste già uno nel tuo account.

Per impostazione predefinita, Elastic Beanstalk crea un sistema Application Load Balancer per l'ambiente quando abiliti il load balancer con la console Elastic Beanstalk o l'interfaccia a riga di comando EB. Configura il load balancer in modo che ascolti il traffico HTTP sulla porta 80 e lo inoltri verso istanze sulla stessa porta. È possibile scegliere il tipo di load balancer che l'ambiente utilizza solo durante la creazione dell'ambiente. In seguito potrai modificare le impostazioni per gestire il comportamento del load balancer in esecuzione nell'ambiente, ma non potrai modificarne il tipo.

> **Note**
>
> L'ambiente deve essere in un VPC con sottoreti in almeno due zone di disponibilità per creare un sistema Application Load Balancer. Tutti i nuovi AWS account includono VPC predefiniti che soddisfano questo requisito.

Consulta i seguenti argomenti per informazioni su ciascun tipo di load balancer supportato da Elastic Beanstalk, le relative funzionalità, su come configurarlo e gestirlo in un ambiente Elastic Beanstalk, nonché su come configurare un load balancer per caricare i log di accesso in Amazon S3.

Argomenti

- Configurazione di un Classic Load Balancer
- Configurazione di un Application Load Balancer
- Configurazione di un sistema Application Load Balancer condiviso
- Configurazione di un sistema Network Load Balancer
- Configurazione dei sistemi di bilanciamento del carico Elastic Beanstalk dual-stack
- Configurazione dei log di accesso

Configurazione di un Classic Load Balancer

Quando abiliti il bilanciamento del carico, il tuo AWS Elastic Beanstalk ambiente è dotato di un sistema di bilanciamento del carico Elastic Load Balancing per distribuire il traffico tra le istanze del tuo ambiente. Elastic Load Balancing supporta diversi tipi di load balancer. Per ulteriori informazioni, consulta la Guida per l'utente di Elastic Load Balancing. Elastic Beanstalk può creare un load balancer automaticamente o ti consente di specificare un load balancer condiviso creato da te.

In questo argomento viene descritta la configurazione di un sistema Classic Load Balancer creato da Elastic Beanstalk per il tuo ambiente. Per informazioni sulla configurazione di tutti i tipi di load balancer supportati da Elastic Beanstalk, consulta Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk.

> **Note**
>
> È possibile scegliere il tipo di load balancer che l'ambiente utilizza solo durante la creazione dell'ambiente. In seguito potrai modificare le impostazioni per gestire il comportamento del load balancer in esecuzione nell'ambiente, ma non potrai modificarne il tipo.

Introduzione

Un sistema Classic Load Balancer è il sistema Elastic Load Balancing della generazione precedente. Supporta l'instradamento del traffico di richieste HTTP, HTTPS o TCP verso porte diverse delle istanze dell'ambiente.

Quando l'ambiente utilizza un sistema Classic Load Balancer, Elastic Beanstalk lo configura per impostazione predefinita in modo che ascolti il traffico HTTP sulla porta 80 e lo inoltri alle istanze sulla stessa porta. Sebbene non sia possibile eliminare il listener predefinito della porta 80, è possibile disabilitarlo. In questo modo si ottiene la stessa funzionalità bloccando il traffico. Nota che puoi aggiungere o eliminare altri ascoltatori. Per supportare connessioni sicure, è possibile configurare il load balancer con un listener sulla porta 443 e un certificato TLS.

Il sistema di bilanciamento del carico utilizza un controllo dello stato per determinare se le EC2 istanze Amazon che eseguono l'applicazione sono integre. Il controllo dello stato invia una richiesta a un URL specificato in un intervallo di tempo preconfigurato. Se l'URL restituisce un messaggio di errore, o non restituisce nulla entro un periodo di timeout specificato, il controllo dello stato ha esito negativo.

Se le prestazioni della tua applicazione migliorano con l'elaborazione di più richieste dallo stesso client su un unico server, puoi configurare il load balancer per l'utilizzo di sticky session. Con le sessioni permanenti, il load balancer aggiunge un cookie alle risposte HTTP che identifica l' EC2 istanza Amazon che ha fornito la richiesta. Quando lo stesso client riceve una richiesta successiva, il load balancer utilizza il cookie per inviare la richiesta alla stessa istanza.

Con il bilanciamento del carico su più zone, ogni nodo del load balancer per Classic Load Balancer distribuisce le richieste in modo uniforme su istanze registrate in tutte le zone di disponibilità abilitate. Se il load balancer su più zone non è attivo, ogni nodo di load balancer distribuisce le richieste in modo uniforme sulle istanze registrate solo nella relativa zona di disponibilità.

Quando un'istanza viene rimossa dal load balancer perché non è più integra o l'ambiente si sta ridimensionando, Connection Draining concede all'istanza il tempo necessario per completare le

richieste prima della chiusura della connessione tra l'istanza e il load balancer. È possibile modificare la quantità di tempo concesso alle istanze per l'invio di una risposta o disabilitare completamente Connection Draining.

> **Note**
>
> Connection Draining viene abilitato per impostazione predefinita al momento della creazione di un ambiente tramite la console Elastic Beanstalk o l'interfaccia a riga di comando EB. Per gli altri client, puoi attivarlo utilizzando le opzioni di configurazione.

Puoi utilizzare le impostazioni avanzate del load balancer per configurare i listener su porte arbitrarie, modificare impostazioni aggiuntive per le sessioni permanenti e configurare il load balancer per connettersi alle istanze in modo sicuro. EC2 Queste impostazioni sono disponibili tramite le opzioni di configurazione che puoi impostare impiegando i file di configurazione del codice sorgente o direttamente in un ambiente utilizzando l'API Elastic Beanstalk. Molte di queste impostazioni sono disponibili anche nella console Elastic Beanstalk. Puoi anche configurare un load balancer in modo che carichi i log di accesso in Amazon S3.

Configurazione di un sistema Classic Load Balancer utilizzando la console Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk per configurare le porte, i certificati HTTPS e altre impostazioni di Classic Load Balancer in fase di creazione o di esecuzione dell'ambiente.

> **Note**
>
> L'opzione Classic Load Balancer (CLB) è disattivata nella procedura guidata della console Crea ambiente. Se disponi di un ambiente esistente configurato con Classic Load Balancer, puoi crearne uno nuovo clonando l'ambiente esistente utilizzando la console Elastic Beanstalk o la CLI EB. È inoltre possibile utilizzare l'EB CLI o AWS CLIcreare un nuovo ambiente configurato con un Classic Load Balancer. Questi strumenti da riga di comando creeranno un nuovo ambiente con un CLB anche se non ne esiste già uno nel tuo account.

Per configurare un sistema Classic Load Balancer di un ambiente in esecuzione nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).

> **Note**
>
> Se la categoria di configurazione di Load balancer (Sistema di bilanciamento del carico) non presenta un pulsante Edit (Modifica), il tuo ambiente non dispone di un load balancer. Per informazioni su come impostarne uno, consultare Modifica del tipo di ambiente.

5. Apporta alla configurazione di Classic Load Balancer le modifiche richieste dall'ambiente.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Impostazioni di Classic Load Balancer

- Listener
- Sessioni
- Bilanciamento del carico su più zone
- Connection Draining
- Controllo dell'integrità

Listener

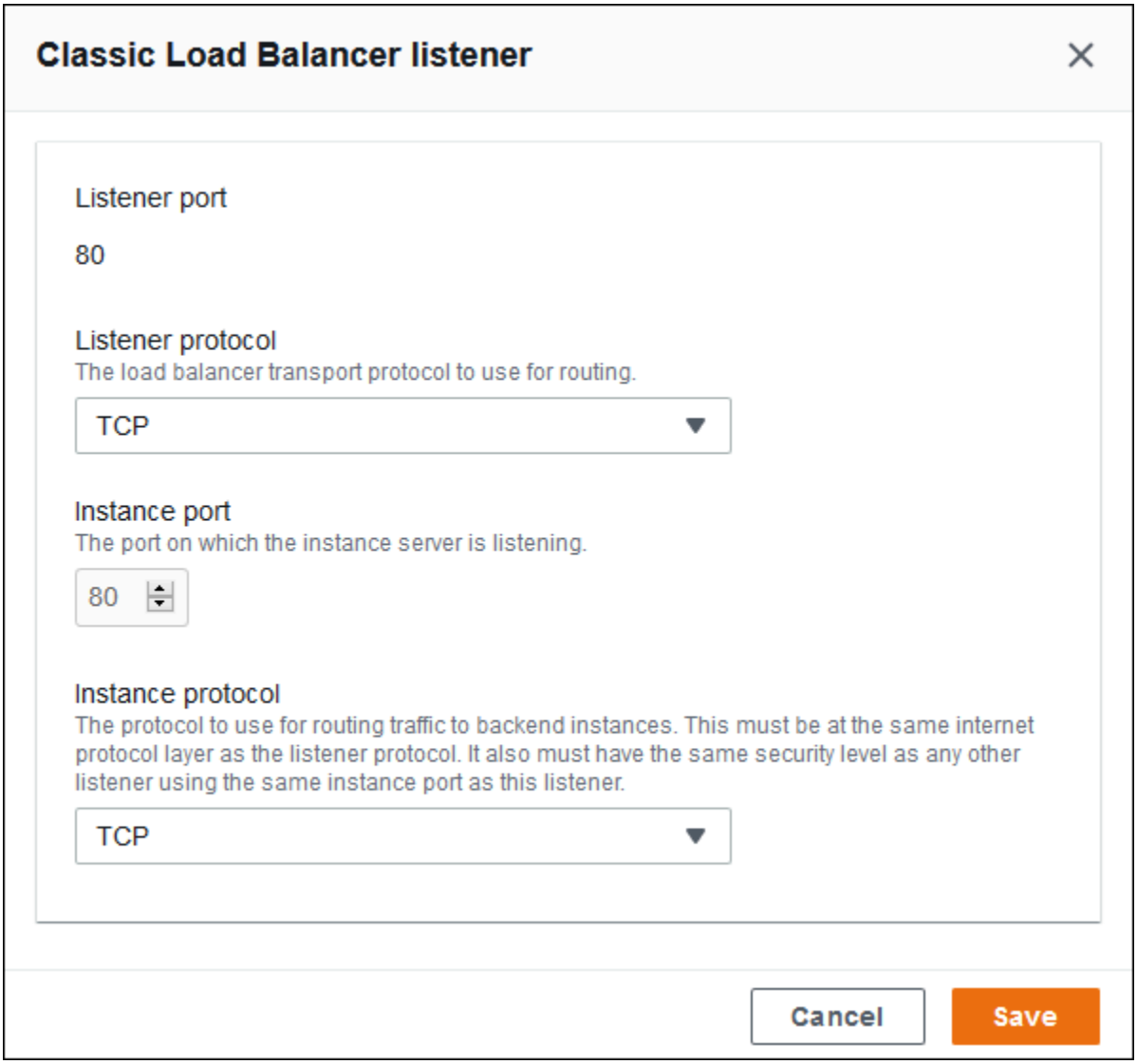
Utilizza questo elenco per specificare i listener per il tuo load balancer. Ogni listener instrada il traffico client in entrata su una determinata porta utilizzando un protocollo sulle tue istanze. Inizialmente, l'elenco mostra il listener predefinito, che instrada il traffico HTTP in entrata sulla porta 80 per il server delle istanze del tuo ambiente che intercetta il traffico HTTP sulla porta 80.

> **Note**
>
> Sebbene non sia possibile eliminare il listener predefinito della porta 80, è possibile disabilitarlo, in modo da ottenere la stessa funzionalità bloccando il traffico.

Configurazione di un listener esistente

1. Seleziona la casella di controllo accanto alla voce della tabella e successivamente Actions (Operazioni) e l'azione desiderata.
2. Se scegli Edit (Modifica), utilizza la finestra di dialogo Classic Load Balancer listener (Listener di Classic Load Balancer) per modificare le impostazioni, quindi seleziona Save (Salva).

Ad esempio, puoi modificare il listener predefinito e modificare il Protocol (Protocollo) da HTTP a TCP se desideri che il load balancer inoltri le richieste così come sono. In questo modo si impedisce al load balancer di riscrivere le intestazioni (incluso `X-Forwarded-For`). La tecnica non funziona con le sticky session.



Aggiunta di un listener

1. Scegli Add listener (Aggiungi listener).
2. Nella finestra di dialogo Classic Load Balancer listener (Listener di Classic Load Balancer), configura le impostazioni e scegli Add (Aggiungi).

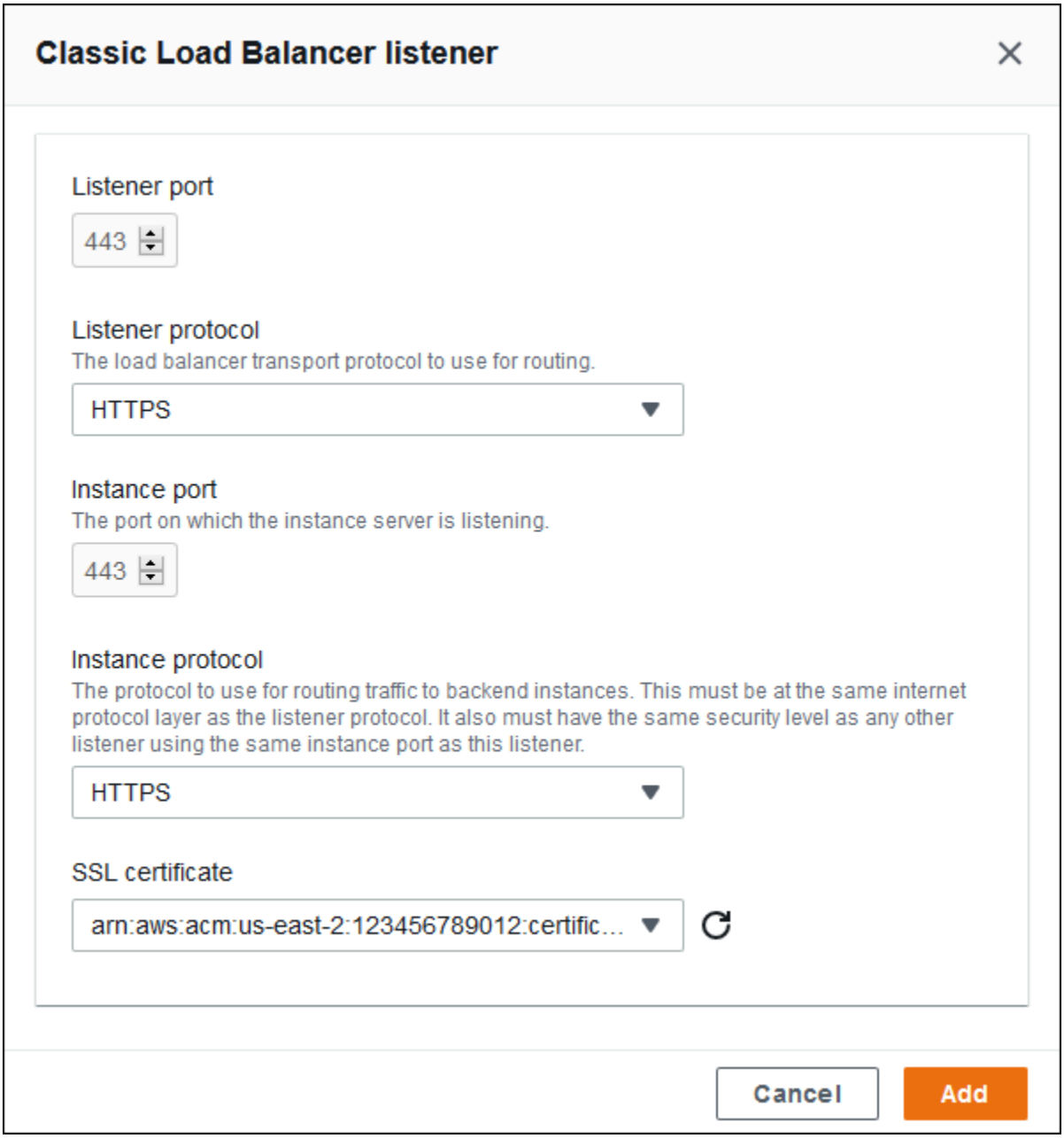
L'aggiunta di un listener sicuro è un caso di utilizzo comune. L'esempio nella seguente immagine aggiunge un listener per il traffico HTTPS sulla porta 443. Il listener instrada il traffico in entrata verso i server delle istanze dell'ambiente che intercettano il traffico HTTPS sulla porta 443.

Prima di configurare un listener HTTPS, assicurati di disporre di un certificato SSL valido. Esegui una delle seguenti operazioni:

- Se AWS Certificate Manager (ACM) è disponibile nella tua AWS regione, crea o importa un certificato utilizzando ACM. Per ulteriori informazioni sulla richiesta di un certificato ACM, consulta Richiesta di un certificato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di certificati nella Guida per l'utente di AWS Certificate Manager .
- Se ACM non è disponibile nella tua AWS regione, carica il certificato e la chiave esistenti su IAM. Per ulteriori informazioni sulla creazione e sul caricamento dei certificati in IAM, consulta l'argomento relativo all'utilizzo dei certificati del server nella Guida per l'utente di IAM.

Per ulteriori informazioni sulla configurazione di HTTPS e sull'utilizzo dei certificati in Elastic Beanstalk, consulta Configurazione del protocollo HTTPS per l'ambiente Elastic Beanstalk.

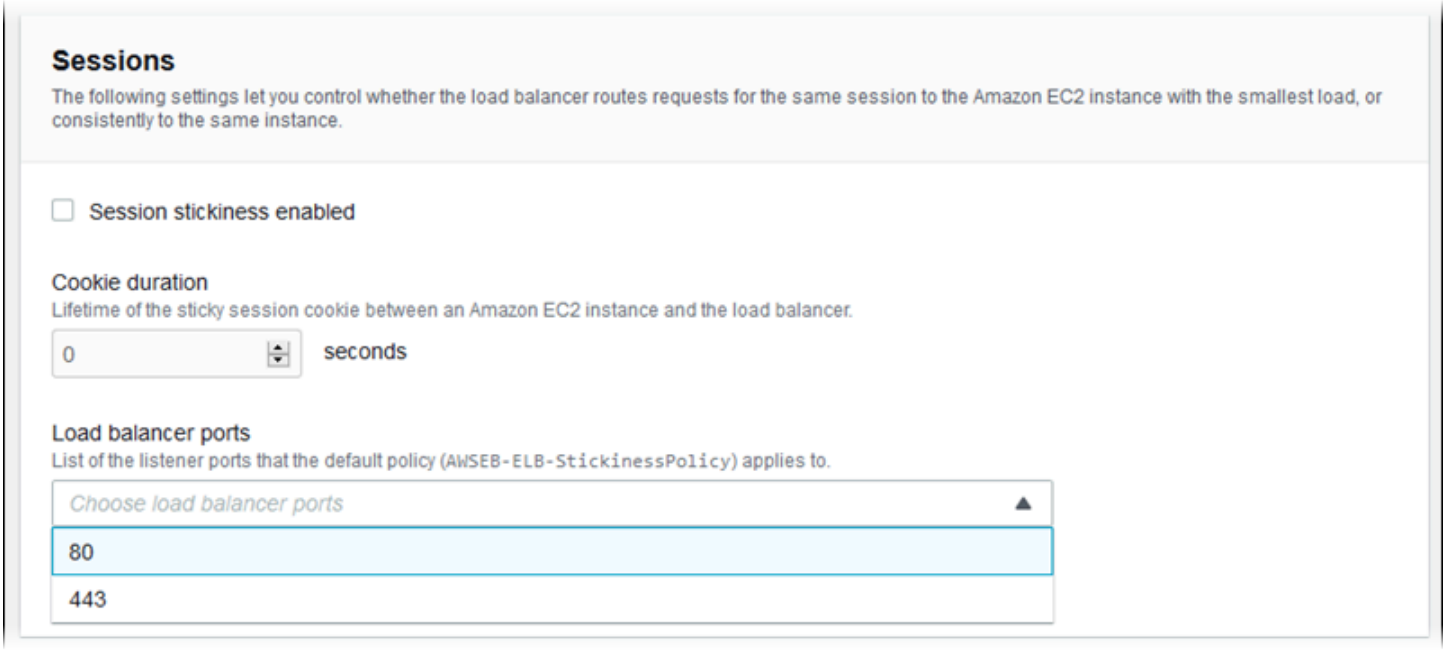
In SSL certificate (Certificato SSL), scegli l'ARN del certificato SSL. Ad esempio, `arn:aws:iam::123456789012:server-certificate/abc/certs/build` o `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`.



Per informazioni dettagliate sulla configurazione di HTTPS e sull'utilizzo dei certificati in Elastic Beanstalk, consulta Configurazione del protocollo HTTPS per l'ambiente Elastic Beanstalk.

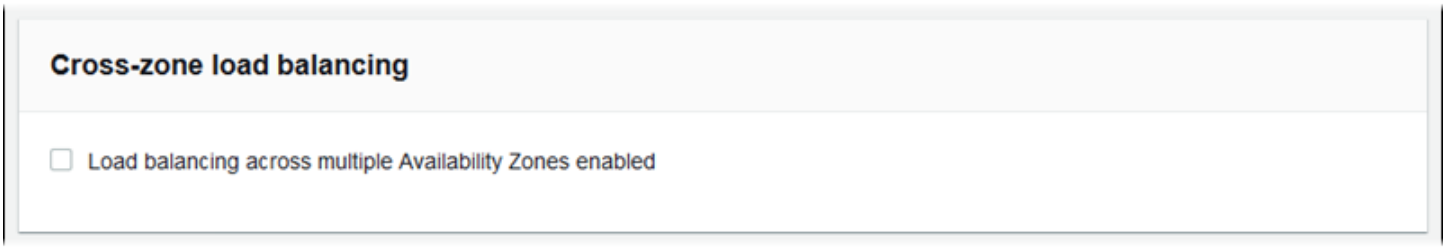
Sessioni

Seleziona o deseleziona la casella Session stickiness enabled (Adesione alla sessione abilitata) per attivare o disattivare le sticky session. Usa Cookie duration (Durata del cookie) per configurare la durata di una sticky session, fino a **`1000000`** secondi. Nell'elenco Load balancer ports (Porte del load balancer) selezionare le porte del listener a cui si applica il criterio predefinito (`AWSEB-ELB-StickinessPolicy`).



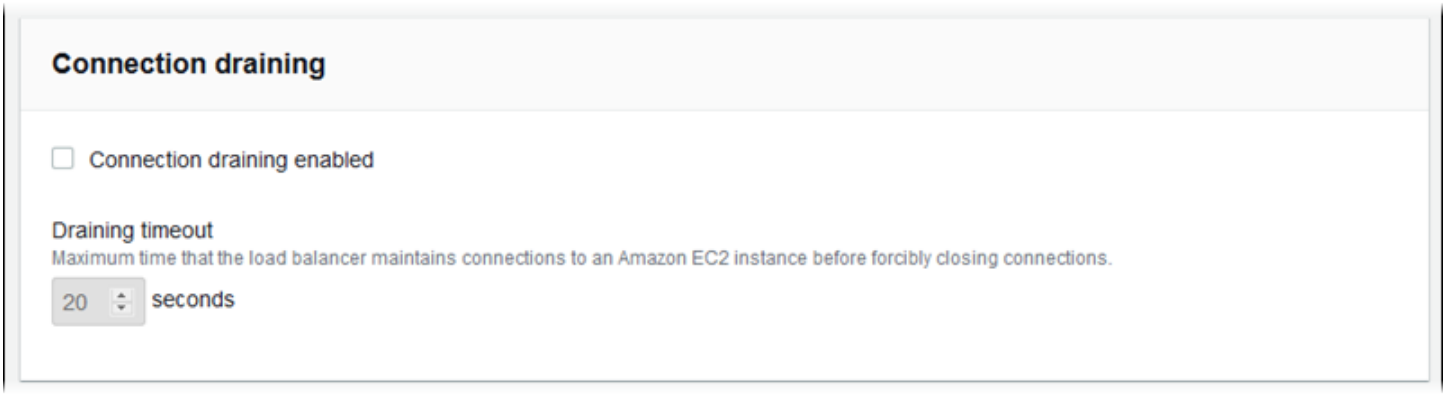
Bilanciamento del carico su più zone

Seleziona o deseleziona la casella Load balancing across multiple Availability Zones enabled (Attivazione del load balancer in più zone di disponibilità) per attivare o disattivare il bilanciamento del carico in più zone.



Connection Draining

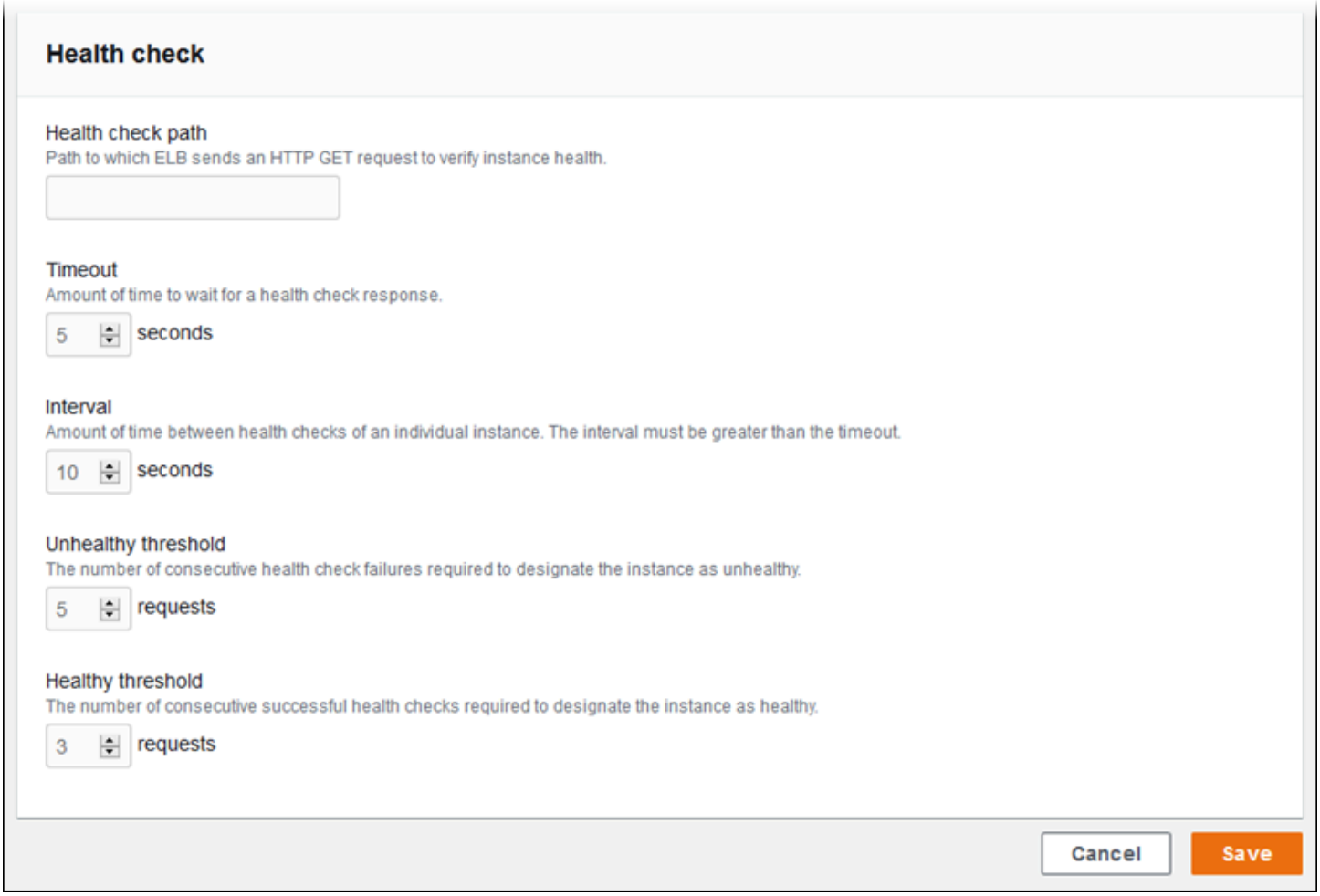
Seleziona o deseleziona la casella Session stickiness enabled (Adesione alla sessione abilitata) per attivare o disattivare Connection Draining. Imposta Draining timeout (Timeout Draining) fino a **`3600`** secondi.



Controllo dell’integrità

Utilizza le seguenti impostazioni per configurare controlli di stato del load balancer:

- Health check path (Percorso di controllo dello stato): percorso in cui il load balancer invia le richieste di controllo dello stato. Se il percorso non è impostato, il load balancer tenta di effettuare una connessione TCP sulla porta 80 per verificare lo stato.
- Timeout: il periodo di attesa espresso in secondi di una risposta durante un controllo dello stato.
- Interval (Intervallo): il periodo di tempo espresso in secondi che decorre tra i controlli dello stato di una singola istanza. L'intervallo deve essere maggiore del timeout.
- Unhealthy threshold (Soglia di mancata integrità), Healthy threshold (Soglia di integrità): il numero di controlli dello stato consecutivi che non devono o devono essere superati rispettivamente prima che Elastic Load Balancing modifichi lo stato di integrità di un'istanza.



> **Note**
>
> Il controllo dello stato di Elastic Load Balancing non influisce sul comportamento del controllo dello stato del gruppo Auto Scaling di un ambiente. Le istanze che non superano un controllo dello stato di Elastic Load Balancing non vengono sostituite automaticamente da Amazon Auto EC2 Scaling a meno che non configuri manualmente Amazon Auto EC2 Scaling a tale scopo. Per informazioni dettagliate, vedi Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk.

Per ulteriori informazioni sui controlli dello stato e su come influiscono sullo stato di integrità generale dell'ambiente, consulta Report dello stato di base.

Configurazione di un sistema Classic Load Balancer utilizzando l'interfaccia a riga di comando EB

Quando esegui eb create, l'interfaccia a riga di comando EB ti chiede di scegliere un tipo di load balancer.

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1):
```

Premi Enter (Invio) per selezionare `classic`.

Puoi anche specificare un tipo di load balancer con l'opzione `--elb-type`.

```
$ eb create test-env --elb-type classic
```

Spazi dei nomi di configurazione di Classic Load Balancer

Le impostazioni relative ai sistemi Classic Load Balancer sono disponibili nei seguenti spazi dei nomi:

- `aws:elb:healthcheck`: configura le soglie, l'intervallo di controllo e il timeout per i controlli dello stato del load balancer.
- `aws:elasticbeanstalk:application`: configura l'URL del controllo dello stato.
- `aws:elb:loadbalancer`: abilita il load balancer su più zone. Assegna i gruppi di sicurezza al load balancer predefinito e ignora il gruppo di sicurezza creato da Elastic Beanstalk. Questo spazio dei nomi include anche opzioni obsolete per la configurazione di listener protetti e standard che sono state sostituite da opzioni nello spazio dei nomi `aws:elb:listener`.
- `aws:elb:listener`: configura il listener predefinito sulla porta 80, un listener protetto sulla porta 443 o listener aggiuntivi per qualsiasi protocollo su qualsiasi porta. Se specifichi `aws:elb:listener` come namespace, le impostazioni si applicano al listener predefinito sulla

porta 80. Se specifichi una porta (ad esempio, `aws:elb:listener:443`), su questa porta viene configurato un listener.

- `aws:elb:policies`: configura impostazioni aggiuntive per il tuo load balancer. Utilizza le opzioni di questo namespace per configurare i listener su porte arbitrarie, modificare impostazioni di sessione permanenti aggiuntive e configurare il load balancer per connettersi alle istanze Amazon in modo sicuro. EC2

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Example.ebextensions/loadbalancer-terminatehttps.config

Il file di configurazione di esempio seguente crea un listener HTTPS sulla porta 443, assegna un certificato che il load balancer utilizza per terminare la connessione sicura e disabilita il listener predefinito sulla porta 80. Il load balancer inoltra le richieste decrittografate alle istanze del tuo ambiente su HTTP:80. EC2

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: HTTPS
    SSLCertificateId: arn:aws:acm:us-
east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678
    InstancePort: 80
    InstanceProtocol: HTTP
  aws:elb:listener:
    ListenerEnabled: false
```

Configurazione di un Application Load Balancer

Quando abiliti il bilanciamento del carico, il tuo AWS Elastic Beanstalk ambiente è dotato di un sistema di bilanciamento del carico Elastic Load Balancing per distribuire il traffico tra le istanze del tuo ambiente. Elastic Load Balancing supporta diversi tipi di load balancer. Per ulteriori informazioni, consulta la Guida per l'utente di Elastic Load Balancing. Elastic Beanstalk può creare un load balancer automaticamente o ti consente di specificare un load balancer condiviso creato da te.

In questo argomento viene descritta la configurazione di un sistema Application Load Balancer creato da Elastic Beanstalk e dedicato all'ambiente in uso. Consulta anche the section called

“Application Load Balancer condiviso”. Per informazioni sulla configurazione di tutti i tipi di load balancer supportati da Elastic Beanstalk, consulta the section called “Bilanciatore del carico”.

> **Note**
>
> È possibile scegliere il tipo di load balancer che l'ambiente utilizza solo durante la creazione dell'ambiente. Puoi modificare le impostazioni per gestire il comportamento del sistema di bilanciamento del carico in esecuzione nell'ambiente, ma non puoi modificarne il tipo. Inoltre, non è possibile passare da un sistema di bilanciamento del carico dedicato a uno condiviso o viceversa.

Introduzione

Un sistema Application Load Balancer ispeziona il traffico a livello di protocollo di rete dell'applicazione per identificare il percorso della richiesta in modo da poter indirizzare le richieste di percorsi diversi a destinazioni diverse.

Quando l'ambiente utilizza un sistema Application Load Balancer, per impostazione predefinita viene configurato da Elastic Beanstalk in modo da eseguire la stessa funzione di un sistema Classic Load Balancer. Il listener predefinito accetta le richieste HTTP sulla porta 80 e le distribuisce alle istanze nel tuo ambiente. Puoi aggiungere un listener protetto sulla porta 443 con un certificato per decrittare il traffico HTTPS, configurare il comportamento del controllo dello stato e inviare i log degli accessi dal sistema di bilanciamento del carico a un bucket Amazon Simple Storage Service (Amazon S3).

> **Note**
>
> A differenza di un Classic Load Balancer o di un Network Load Balancer, un Application Load Balancer non può avere TCP o listener di transport layer (layer 4). SSL/TLS Supporta solo listener HTTP e HTTPS. Inoltre, non è possibile utilizzare l'autenticazione di back-end per autenticare le connessioni HTTPS tra un sistema di bilanciamento del carico e le istanze di back-end.

In un ambiente Elastic Beanstalk puoi utilizzare un sistema Application Load Balancer per indirizzare il traffico per determinati percorsi a un processo diverso nelle istanze del server Web. Con un sistema Classic Load Balancer, tutto il traffico indirizzato a un listener viene instradato a un singolo processo sulle istanze di back-end. Con un sistema Application Load Balancer puoi configurare più regole sul

listener per instradare le richieste verso percorsi specifici a processi di back-end diversi. È possibile configurare ogni processo con la porta su cui il processo è in ascolto.

Ad esempio, è possibile eseguire un processo di login separato dall'applicazione principale. Anche se l'applicazione principale sulle istanze del tuo ambiente accetta la maggior parte delle richieste e resta in attesa sulla porta 80, il processo di login resta in attesa sulla porta 5000 e accetta le richieste nel percorso `/login`. Tutte le richieste in entrata dai clienti arrivano sulla porta 80. Con un sistema Application Load Balancer puoi configurare un solo listener per il traffico in entrata sulla porta 80, con due regole che instradano il traffico verso due processi separati, a seconda del percorso indicato nella richiesta. È possibile aggiungere una regola personalizzata che indirizza il traffico `/login` al processo di accesso in ascolto sulla porta 5000. La regola predefinita instrada tutte il traffico restante nel processo dell'applicazione principale in attesa sulla porta 80.

Una regola Application Load Balancer esegue il mapping di una richiesta a un gruppo di destinazione. In Elastic Beanstalk, un gruppo di destinazione è rappresentato da un processo. È possibile configurare un processo con un protocollo, una porta e le impostazioni di controllo dello stato. Il processo rappresenta il processo in esecuzione sulle istanze nel tuo ambiente. Il processo predefinito è un listener sulla porta 80 del server proxy inverso (nginx o Apache) su cui viene eseguita l'applicazione.

> **Note**
>
> Al di fuori di Elastic Beanstalk un gruppo di destinazione esegue il mapping a un gruppo di istanze. Un listener può utilizzare regole e gruppi di destinazione per instradare il traffico verso istanze diverse in base al percorso. All'interno di Elastic Beanstalk tutte le istanze dell'ambiente sono identiche, pertanto la distinzione viene fatta tra i processi in ascolto su porte diverse.

Un sistema Classic Load Balancer utilizza un unico percorso di controllo dello stato per l'intero ambiente. Con un sistema Application Load Balancer ogni processo ha un percorso di controllo dello stato separato monitorato dal sistema di bilanciamento del carico e dal monitoraggio dello stato avanzato di Elastic Beanstalk.

Per utilizzare un sistema Application Load Balancer, l'ambiente deve trovarsi in un VPC predefinito o personalizzato e deve essere dotato di un ruolo di servizio con il set standard di autorizzazioni. Se disponi di un ruolo del servizio precedente, potrebbe essere necessario aggiornare le autorizzazioni per includere `elasticloadbalancing:DescribeTargetHealth`

e `elasticloadbalancing:DescribeLoadBalancers`. Per ulteriori informazioni sui sistemi Application Load Balancer, consulta Cos'è un Application Load Balancer?.

> **Note**
>
> Il controllo dello stato di Application Load Balancer non utilizza il percorso del controllo dello stato di Elastic Beanstalk. Utilizza, invece, il percorso specifico configurato per ogni processo separatamente.

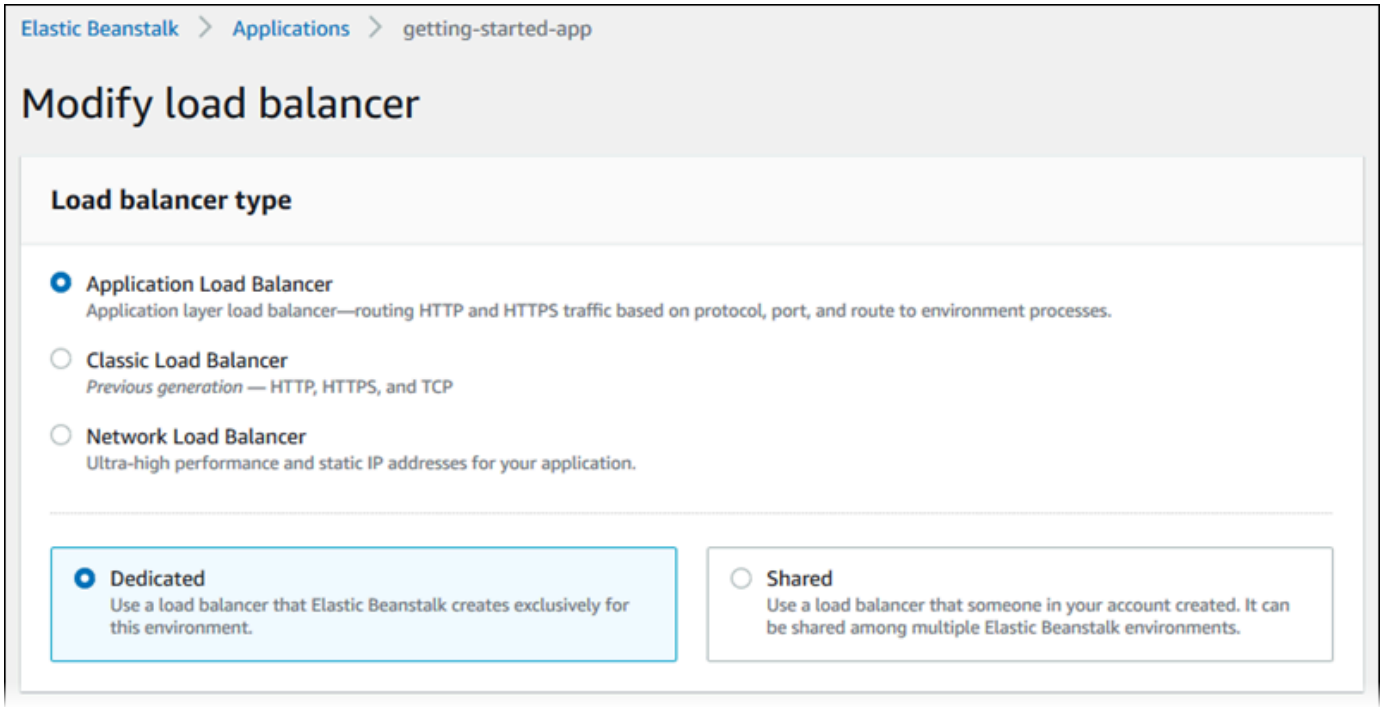
Configurazione di un sistema Application Load Balancer utilizzando la console Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk per configurare i listener, i processi e le regole di un sistema Application Load Balancer durante la creazione dell'ambiente o in un secondo momento quando l'ambiente è in esecuzione.

Per configurare un sistema Application Load Balancer nella console Elastic Beanstalk durante la creazione dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegliere Create a new environment (Crea un nuovo ambiente) per iniziare a creare l'ambiente.
4. Nella pagina principale della procedura guidata, prima di scegliere Create environment (Crea ambiente), scegli Configure more options (Configura più opzioni).
5. Scegli la configurazione preimpostata High availability (Alta disponibilità).

   In alternativa, nella categoria di configurazione Capacity (Capacità), configura un tipo di ambiente Load balanced (Con bilanciamento del carico). Per informazioni dettagliate, vedi Capacity.
6. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).
7. Seleziona le opzioni Application Load Balancer e Dedicated (Dedicato) se non sono già selezionate.



8. Apporta alla configurazione di Application Load Balancer le modifiche richieste dall'ambiente.
9. Scegli Save (Salva) e apporta eventuali altre modifiche di configurazione richieste dal tuo ambiente.
10. Seleziona Create environment (Crea ambiente).

Per configurare un sistema Application Load Balancer di un ambiente in esecuzione nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).

> **Note**
>
> Se la categoria di configurazione di Load balancer (Sistema di bilanciamento del carico) non presenta un pulsante Edit (Modifica), il tuo ambiente non dispone di un load

balancer. Per informazioni su come impostarne uno, consultare Modifica del tipo di ambiente.

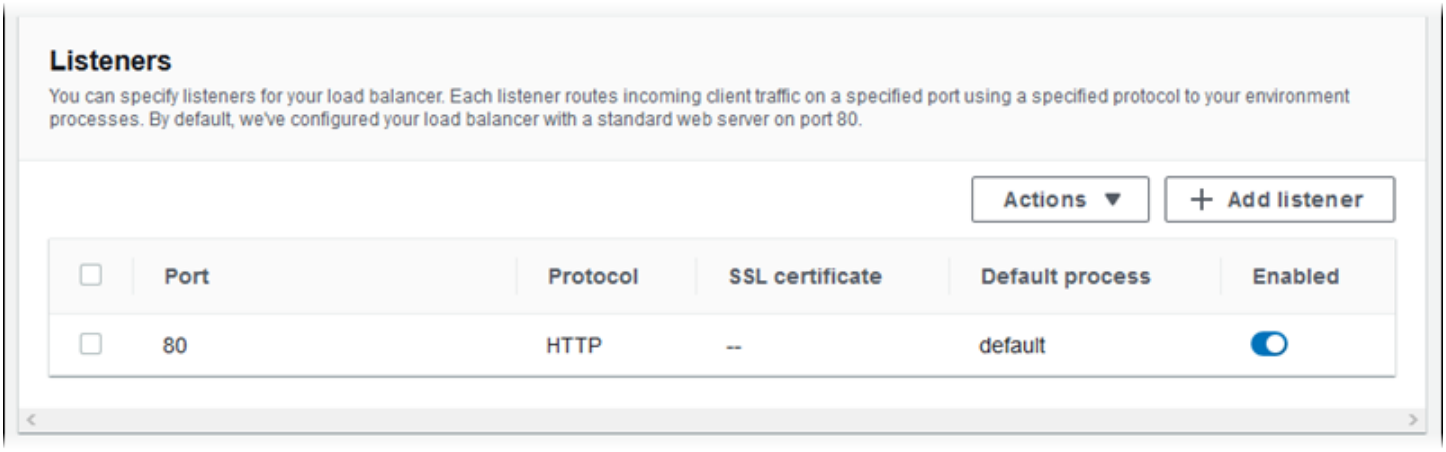
5. Apporta alla configurazione di Application Load Balancer le modifiche richieste dall'ambiente.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Impostazioni di Application Load Balancer

- Listener
- Processes
- Regole
- Acquisizione dei log di accesso

Listener

Utilizza questo elenco per specificare i listener per il tuo load balancer. Ogni listener instrada il traffico client in entrata su una determinata porta utilizzando un protocollo specificato per uno o più processi sulle tue istanze. Inizialmente, l'elenco mostra il listener predefinito che indirizza il traffico HTTP in entrata sulla porta 80 verso un processo denominato predefinito.



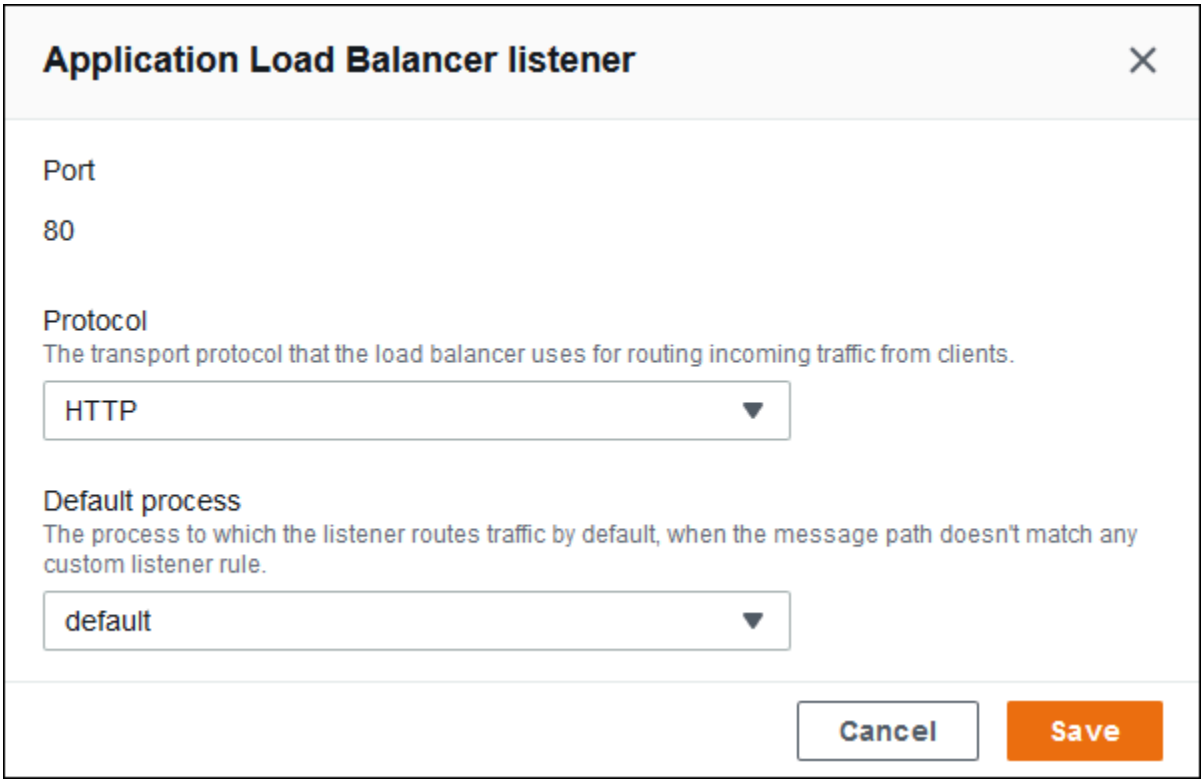
Configurazione di un listener esistente

1. Seleziona la casella di controllo accanto alla voce della tabella e successivamente Actions (Operazioni) e Edit (Modifica).
2. Utilizza la finestra di dialogo Application Load Balancer listener (Listener di Application Load Balancer) per modificare le impostazioni, quindi seleziona Save (Salva).

Aggiunta di un listener

1. Scegli Add listener (Aggiungi listener).
2. Nella finestra di dialogo Application Load Balancer listener (Listener del sistema di bilanciamento del carico applicazioni), configura le impostazioni e scegli Add (Aggiungi).

Utilizza le impostazioni della finestra di dialogo Listener di Application Load Balancer per scegliere la porta e il protocollo utilizzati dal listener per intercettare il traffico e il processo per instradare il traffico. Se scegli il protocollo HTTPS, configura le impostazioni SSL.



Prima di configurare un listener HTTPS, assicurati di disporre di un certificato SSL valido. Esegui una delle seguenti operazioni:

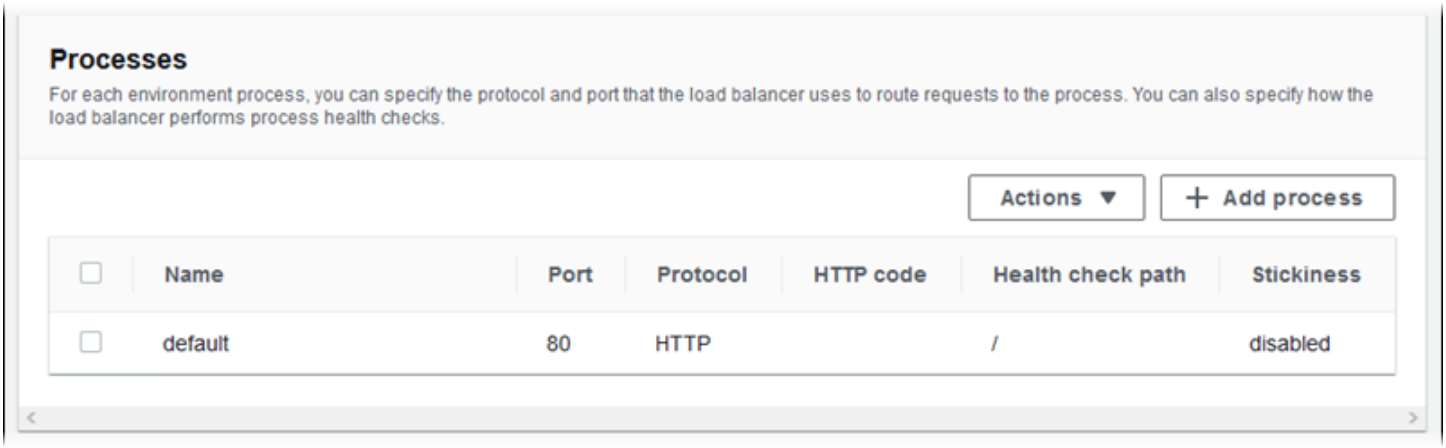
- Se AWS Certificate Manager (ACM) è disponibile nella tua AWS regione, crea o importa un certificato utilizzando ACM. Per ulteriori informazioni sulla richiesta di un certificato ACM, consulta Richiesta di un certificato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di certificati nella Guida per l'utente di AWS Certificate Manager .

- Se ACM non è disponibile nella tua AWS regione, carica il certificato e la chiave esistenti su IAM. Per ulteriori informazioni sulla creazione e sul caricamento dei certificati in IAM, consulta l'argomento relativo all'utilizzo dei certificati del server nella Guida per l'utente di IAM.

Per ulteriori informazioni sulla configurazione di HTTPS e sull'utilizzo dei certificati in Elastic Beanstalk, consulta Configurazione del protocollo HTTPS per l'ambiente Elastic Beanstalk.

Processes

Utilizza questo elenco per specificare i processi per il tuo sistema di bilanciamento del carico. Un processo è la destinazione di instradamento del traffico da parte dei listener. Ogni listener instrada il traffico client in entrata su una determinata porta utilizzando un protocollo specificato per uno o più processi sulle tue istanze. Inizialmente, l'elenco mostra il processo predefinito, che intercetta il traffico in entrata sulla porta HTTP 80.



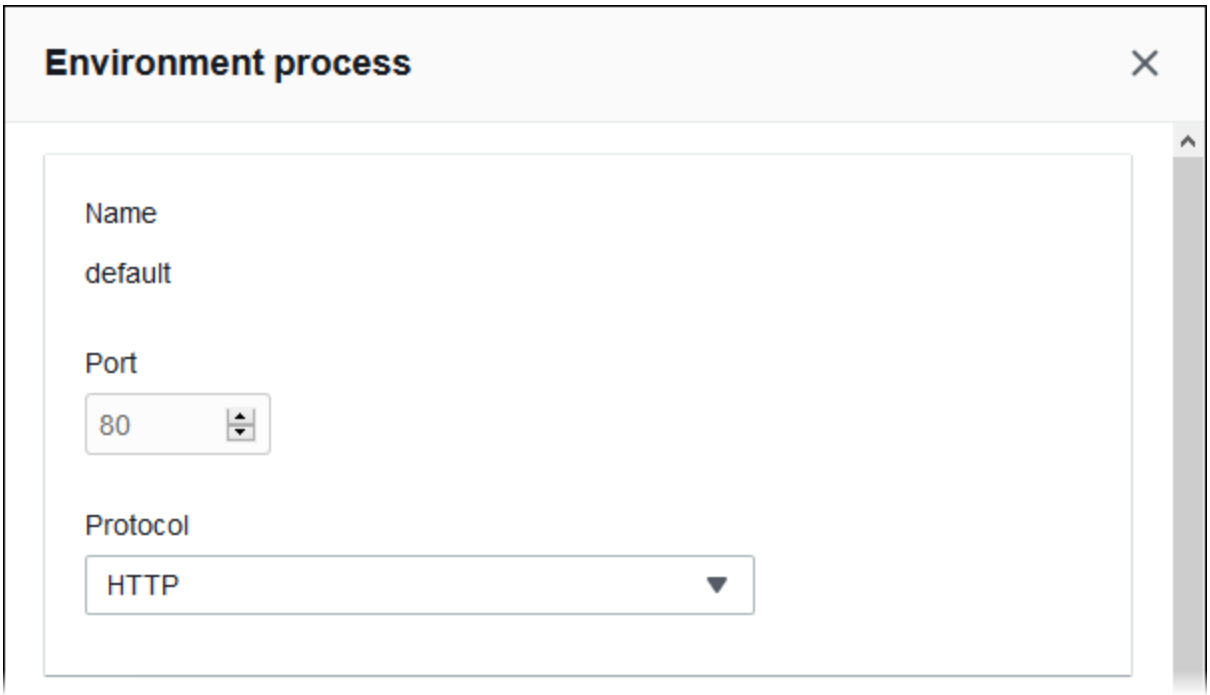
Puoi modificare le impostazioni di un processo esistente o aggiungerne uno nuovo. Per iniziare a modificare un processo dell'elenco o per aggiungere un processo, utilizza la stessa procedura indicata per l'elenco di listener. Viene visualizzata la finestra di dialogo Environment process (Processo ambiente).

Impostazioni della finestra di dialogo del processo di ambiente di Application Load Balancer

- Definizione
- Controllo dell’integrità
- Sessioni

Definizione

Utilizza queste impostazioni per definire il processo: Name (Nome), Port (Porta) e Protocol (Protocollo) attraverso cui riceverà le richieste.



Controllo dell’integrità

Utilizza le seguenti impostazioni per configurare controlli di stato del processo:

- HTTP code (Codice HTTP): codice di stato HTTP che indica un processo integro.
- Path (Percorso): percorso della richiesta di controllo dello stato del processo.
- Timeout: il periodo di attesa espresso in secondi di una risposta durante un controllo dello stato.
- Interval (Intervallo): il periodo di tempo espresso in secondi che decorre tra i controlli dello stato di una singola istanza. L'intervallo deve essere maggiore del timeout.
- Unhealthy threshold (Soglia di mancata integrità), Healthy threshold (Soglia di integrità): il numero di controlli dello stato consecutivi che non devono o devono essere superati rispettivamente prima che Elastic Load Balancing modifichi lo stato di integrità di un'istanza.
- Deregistration delay (Ritardo annullamento della registrazione): il periodo di tempo espresso in secondi necessario al completamento delle richieste prima che venga annullata la registrazione di un'istanza.

Health check

HTTP code
HTTP status code of a healthy instance in your environment.

Path
Path to which the load balancer sends HTTP health check requests.

/

Timeout
Amount of time to wait for a health check response.

5 seconds

Interval
Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

15 seconds

Unhealthy threshold
The number of consecutive health check failures required to designate the instance as unhealthy.

5 requests

Healthy threshold
The number of consecutive successful health checks required to designate the instance as healthy.

3 requests

Deregistration delay
Amount of time to wait for active requests to complete before deregistering.

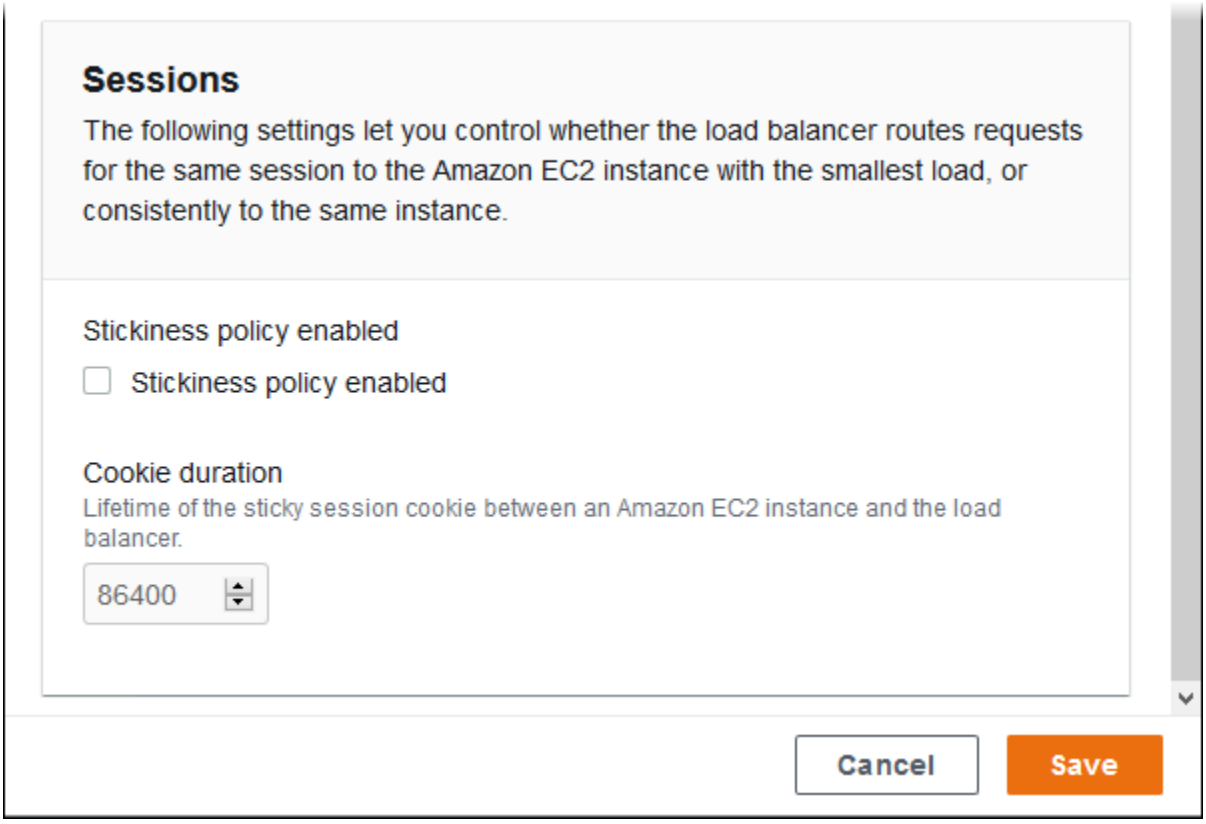
20 seconds

> **Note**
>
> Il controllo dello stato di Elastic Load Balancing non influisce sul comportamento del controllo dello stato del gruppo Auto Scaling di un ambiente. Le istanze che non superano un controllo dello stato di Elastic Load Balancing non vengono sostituite automaticamente da Amazon Auto EC2 Scaling a meno che non configuri manualmente Amazon Auto EC2 Scaling a tale scopo. Per informazioni dettagliate, vedi Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk.

Per ulteriori informazioni sui controlli dello stato e su come influiscono sullo stato di integrità generale dell'ambiente, consulta Report dello stato di base.

Sessioni

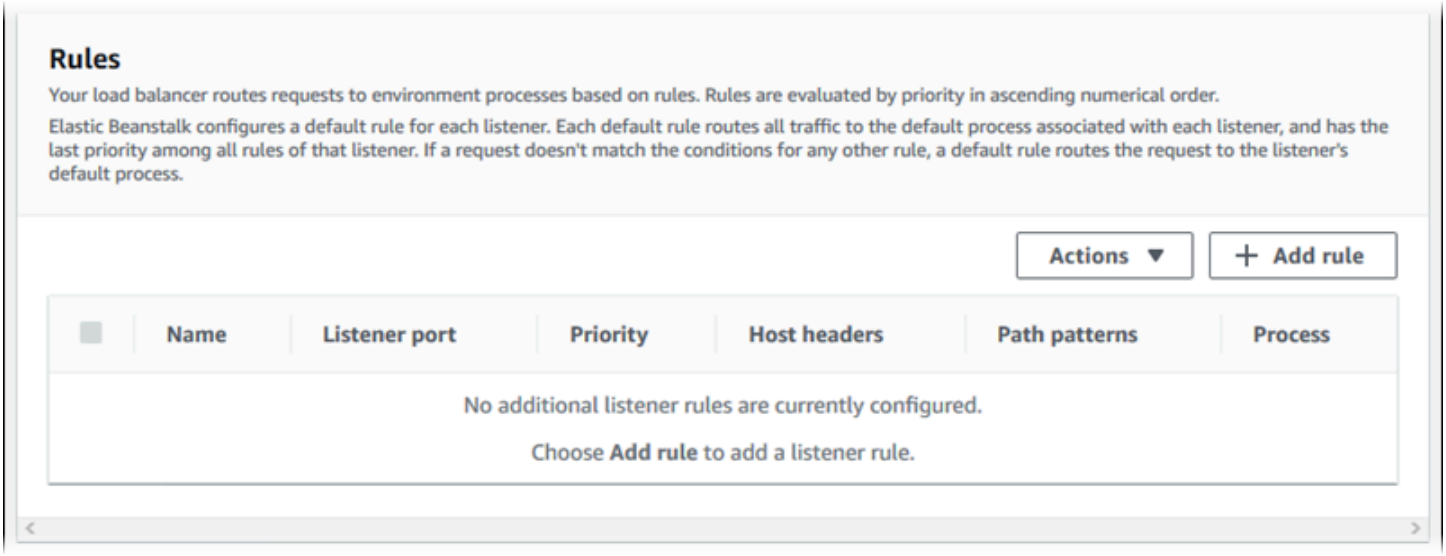
Seleziona o deseleziona la casella Stickiness policy enabled (Policy di adesione abilitata) per attivare o disattivare le sticky session. Usa Cookie duration (Durata del cookie) per configurare la durata di una sticky session, fino a **`604800`** secondi.



Regole

Utilizza questo elenco per specificare le regole dei listener personalizzate per il tuo sistema di bilanciamento del carico. Una regola associa le richieste ricevute dal listener relative a un determinato modello di percorso a un processo di destinazione. Ogni listener può avere più regole, richieste di instradamento su percorsi distinti per diversi processi delle istanze.

Le regole hanno priorità numeriche che stabiliscono la precedenza in cui vengono applicate alle richieste in entrata. Per ogni nuovo listener aggiunto, Elastic Beanstalk aggiunge una regola predefinita che instrada il traffico di tutti i listener verso il processo predefinito. La precedenza della regola predefinita è la più bassa: viene applicata nel caso in cui non ci siano altre regole per lo stesso listener che corrispondono alla richiesta in entrata. Inizialmente, se non sono state aggiunte regole personalizzate, l'elenco è vuoto. Le regole predefinite di tutti i listener non vengono visualizzate.



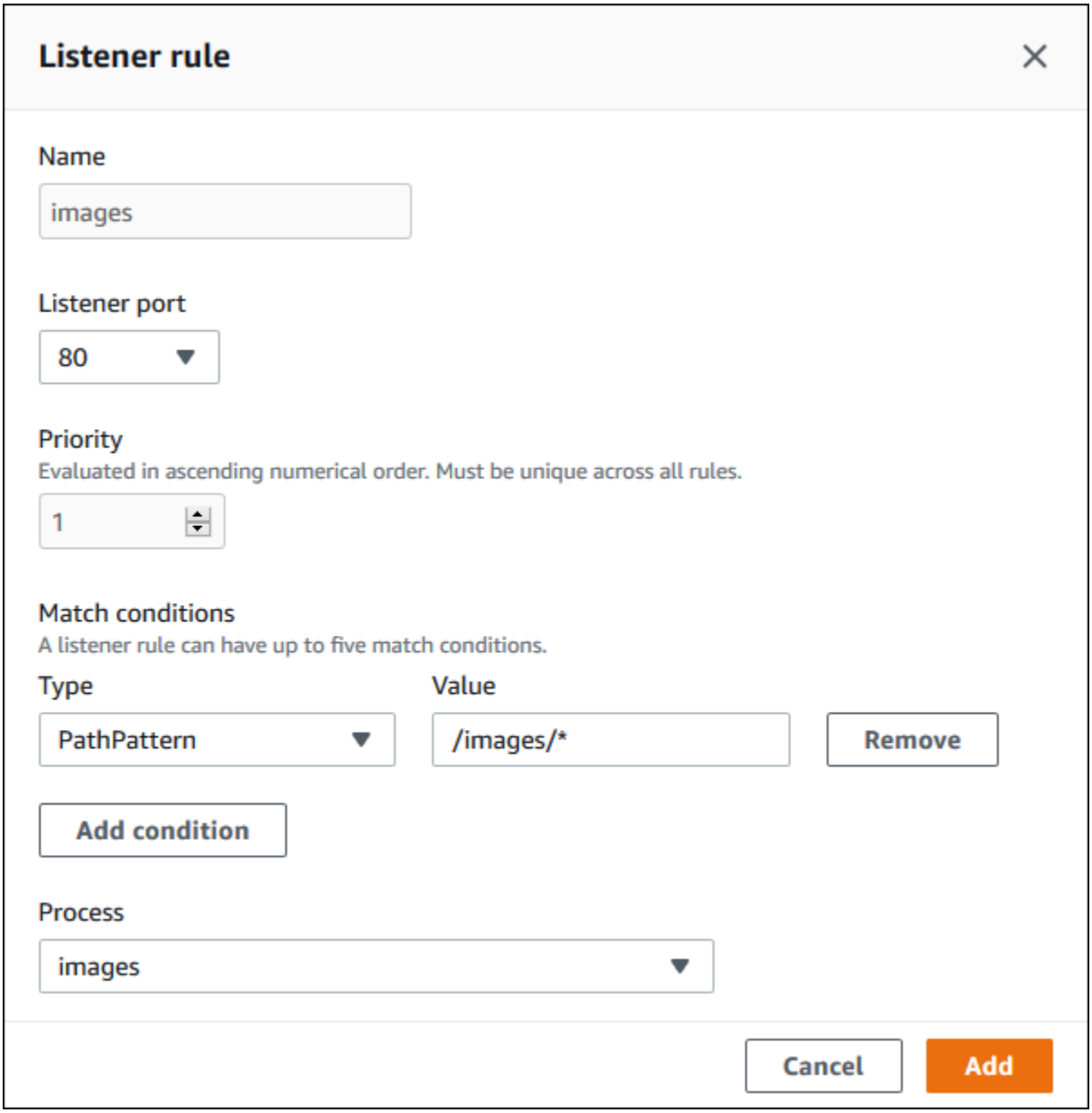
Puoi modificare le impostazioni di una regola esistente o aggiungere una nuova regola. Per iniziare a modificare una regola dell'elenco o per aggiungere una regola, utilizza la stessa procedura indicata per l'elenco di listener. Viene visualizzata la finestra di dialogo Listener rule (Regola listener), con le impostazioni seguenti:

- Name (Nome): nome della regola.
- Listener port (Porta listener): porta del listener a cui si applica la regola.
- Priority (Priorità): priorità della regola. Un numero di priorità inferiore ha la precedenza. Le priorità delle regole del listener devono essere uniche.
- Match conditions (Condizioni di corrispondenza): elenco delle condizioni dell'URL della richiesta a cui si applica la regola. Esistono due tipi di condizioni: HostHeader(la parte del dominio dell'URL) e

PathPattern(la parte del percorso dell'URL). È possibile aggiungere fino a cinque condizioni. Ogni valore di condizione è lungo fino a 128 caratteri e può includere caratteri jolly.

- Process (Processo): processo al quale il sistema di bilanciamento del carico instrada le richieste che soddisfano la regola.

Quando si modifica una regola esistente, non è possibile modificarne Name (Nome) e Listener port (Porta listener).

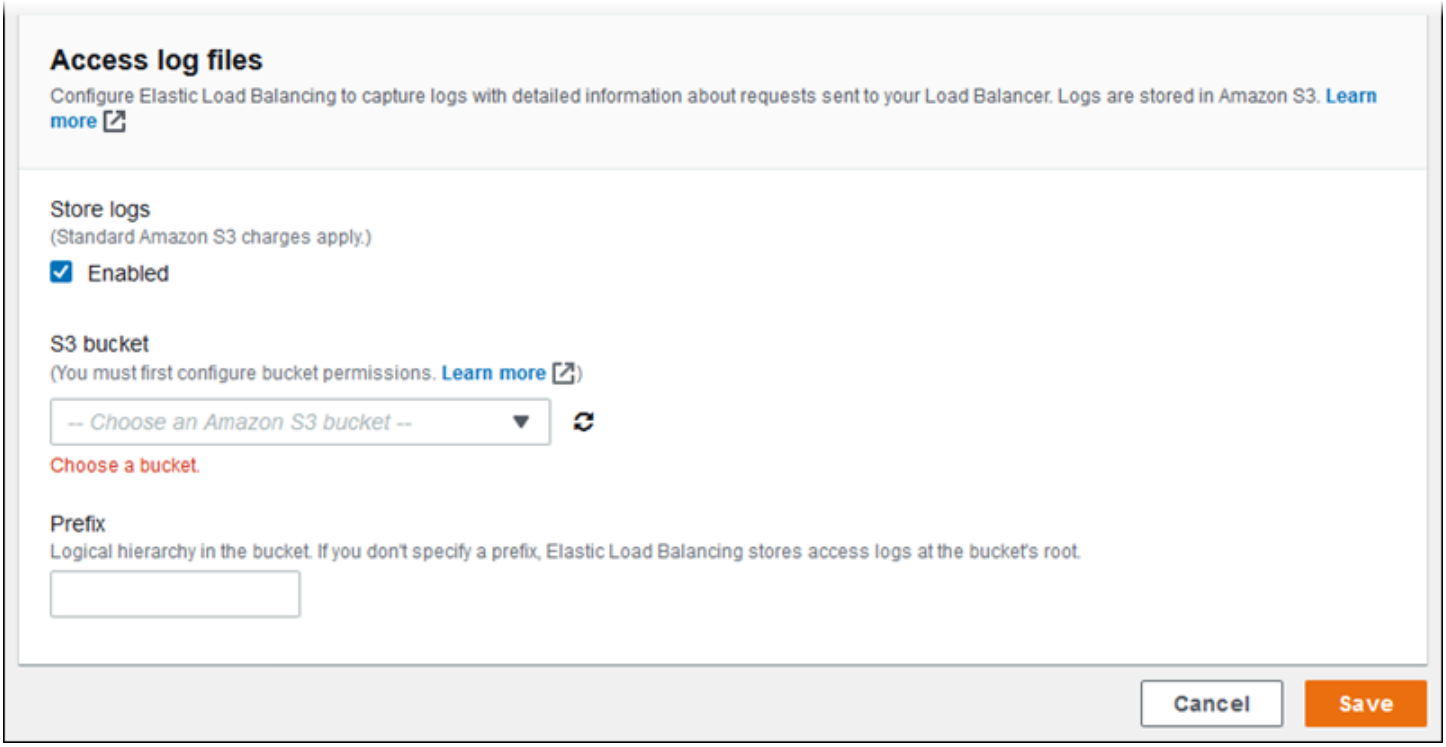


Acquisizione dei log di accesso

Utilizza queste impostazioni per configurare Elastic Load Balancing al fine di acquisire i log con le informazioni dettagliate sulle richieste inviate al tuo Application Load Balancer. Per impostazione predefinita, l'acquisizione dei log di accesso è disattivata. Quando Store logs (Archivia i log) è abilitato, Elastic Load Balancing archivia i log nel bucket S3 che configuri. L'impostazione del prefisso indica una cartella di livello superiore nel bucket per i log. Elastic Load Balancing inserisce i log in una cartella denominata `AWSLogs` con il prefisso impostato. Se non specifichi un prefisso, Elastic Load Balancing posiziona la cartella a livello root del bucket.

> **Note**
>
> Se il bucket Amazon S3 che configuri per l'acquisizione dei log di accesso non è il bucket creato da Elastic Beanstalk per il tuo account, assicurati di aggiungere una policy utente con le autorizzazioni appropriate per i tuoi utenti (IAM). AWS Identity and Access Management Le policy utente gestite fornite da Elastic Beanstalk coprono solo le autorizzazioni per le risorse gestite da Elastic Beanstalk.

Per informazioni dettagliate sui log di accesso, incluse le autorizzazioni e altri requisiti, consulta Log di accesso per Application Load Balancer.



Esempio: Application Load Balancer con un listener protetto e due processi

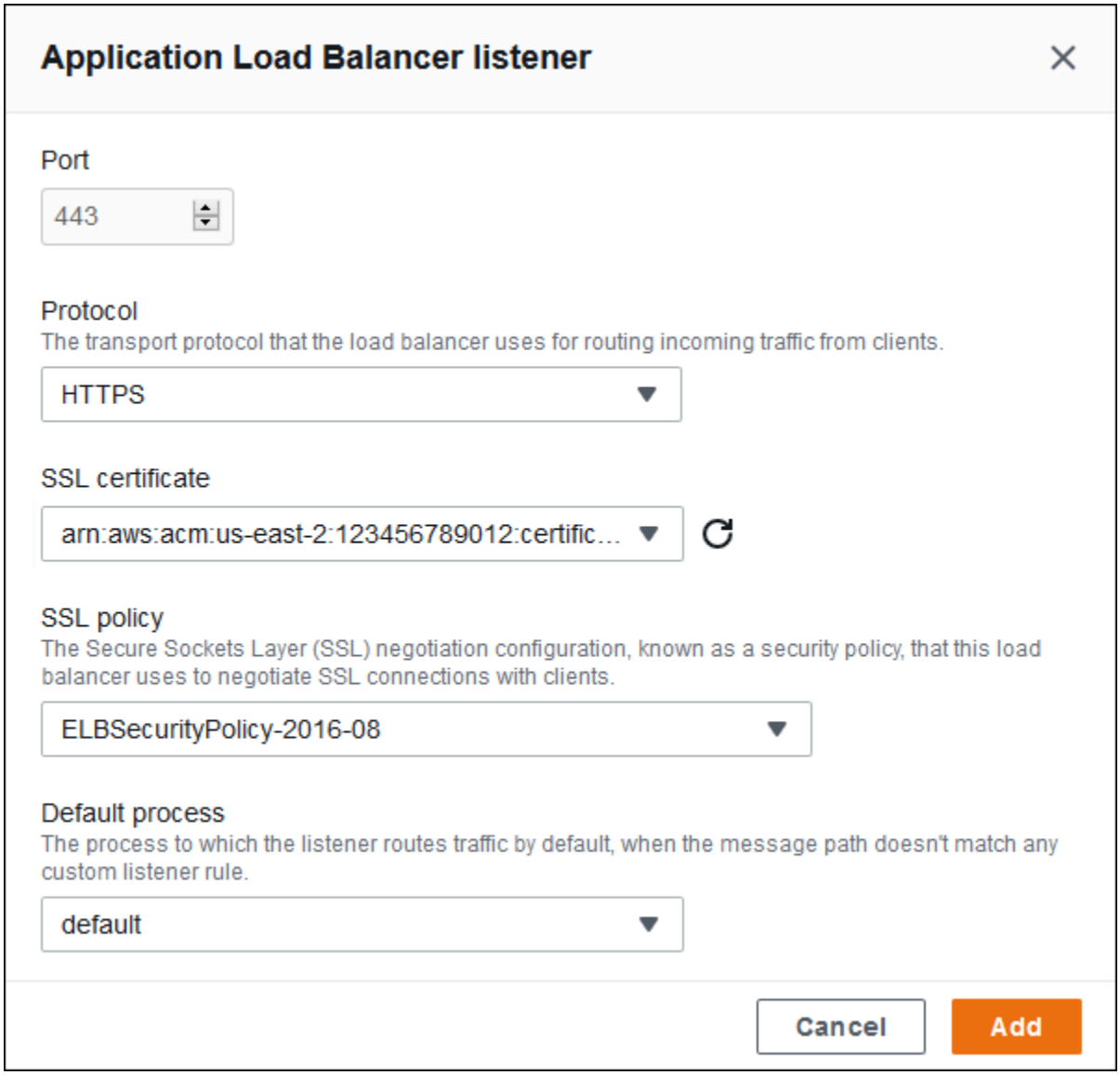
In questo esempio, l'applicazione richiede la crittografia del end-to-end traffico e un processo separato per la gestione delle richieste amministrative.

Per configurare il sistema Application Load Balancer dell'ambiente e soddisfare questi requisiti, puoi rimuovere il listener predefinito, aggiungere un listener HTTPS, indicare che il processo predefinito utilizza la porta 443 su HTTPS e aggiungere un processo e una regola del listener per il traffico di amministrazione su un percorso diverso.

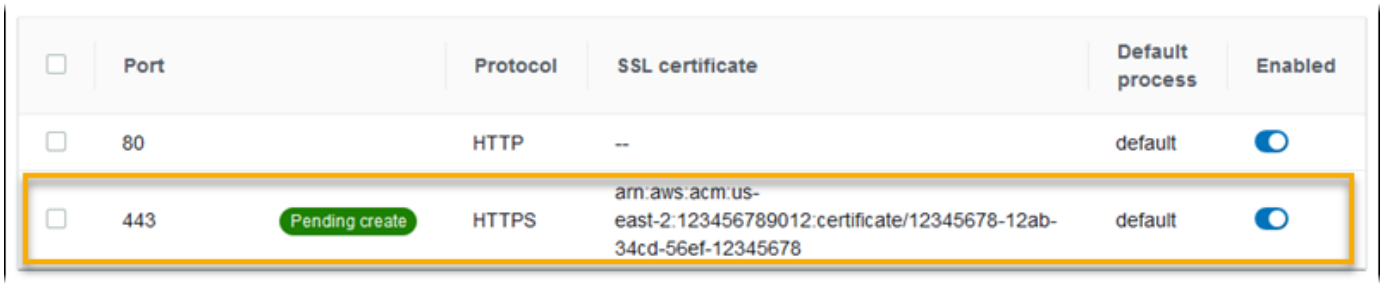
Configurazione del sistema di bilanciamento del carico per questo esempio

1. Aggiungi un listener protetto. In Porta digita **`443`**. In Protocollo, seleziona **`HTTPS`**. In Certificato SSL, scegli l'ARN del certificato SSL. Ad esempio, **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** o **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**.

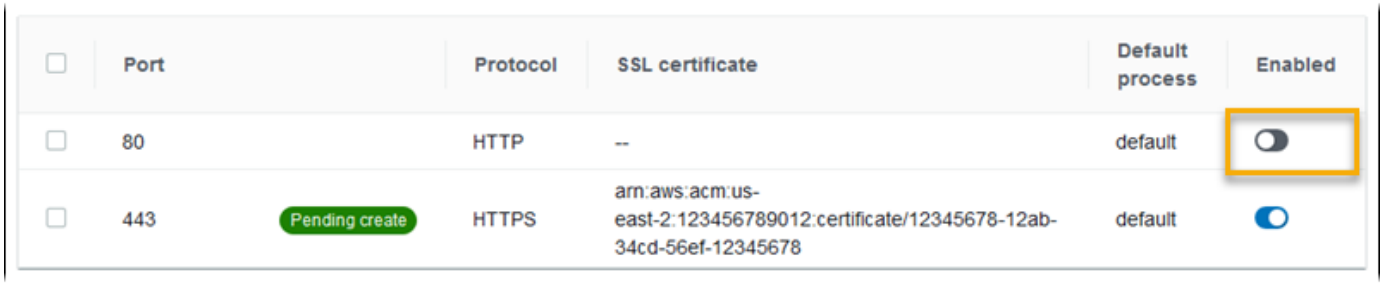
   Per Processo predefinito, mantieni **`default`** selezionato.



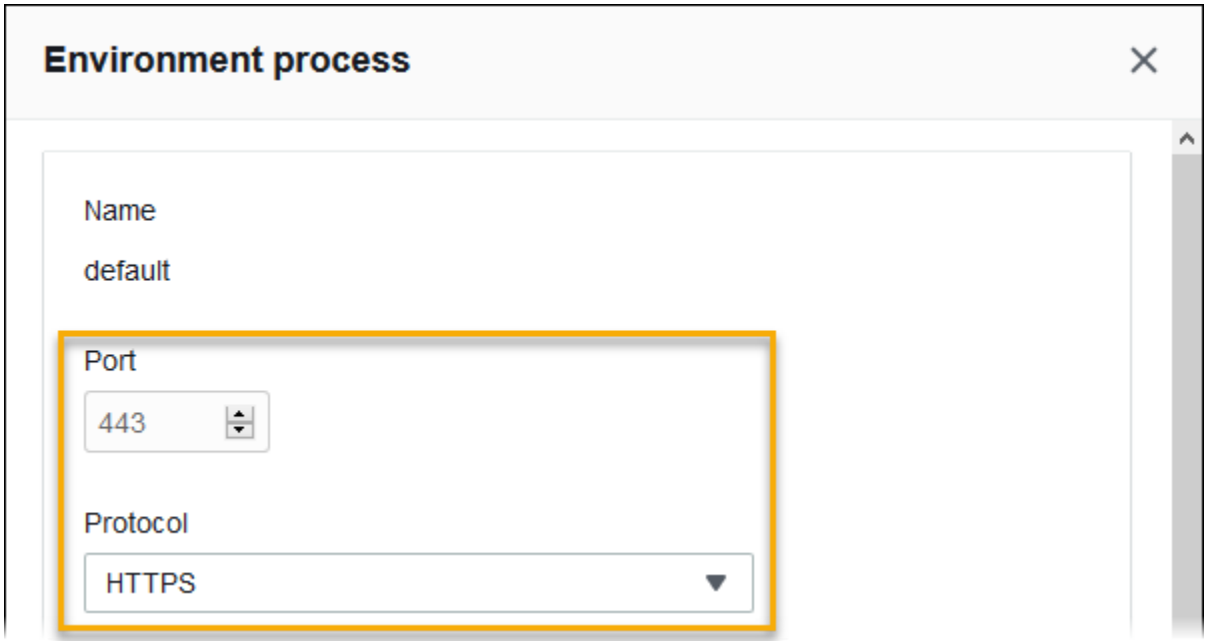
Ora puoi visualizzare il nuovo listener nell'elenco.



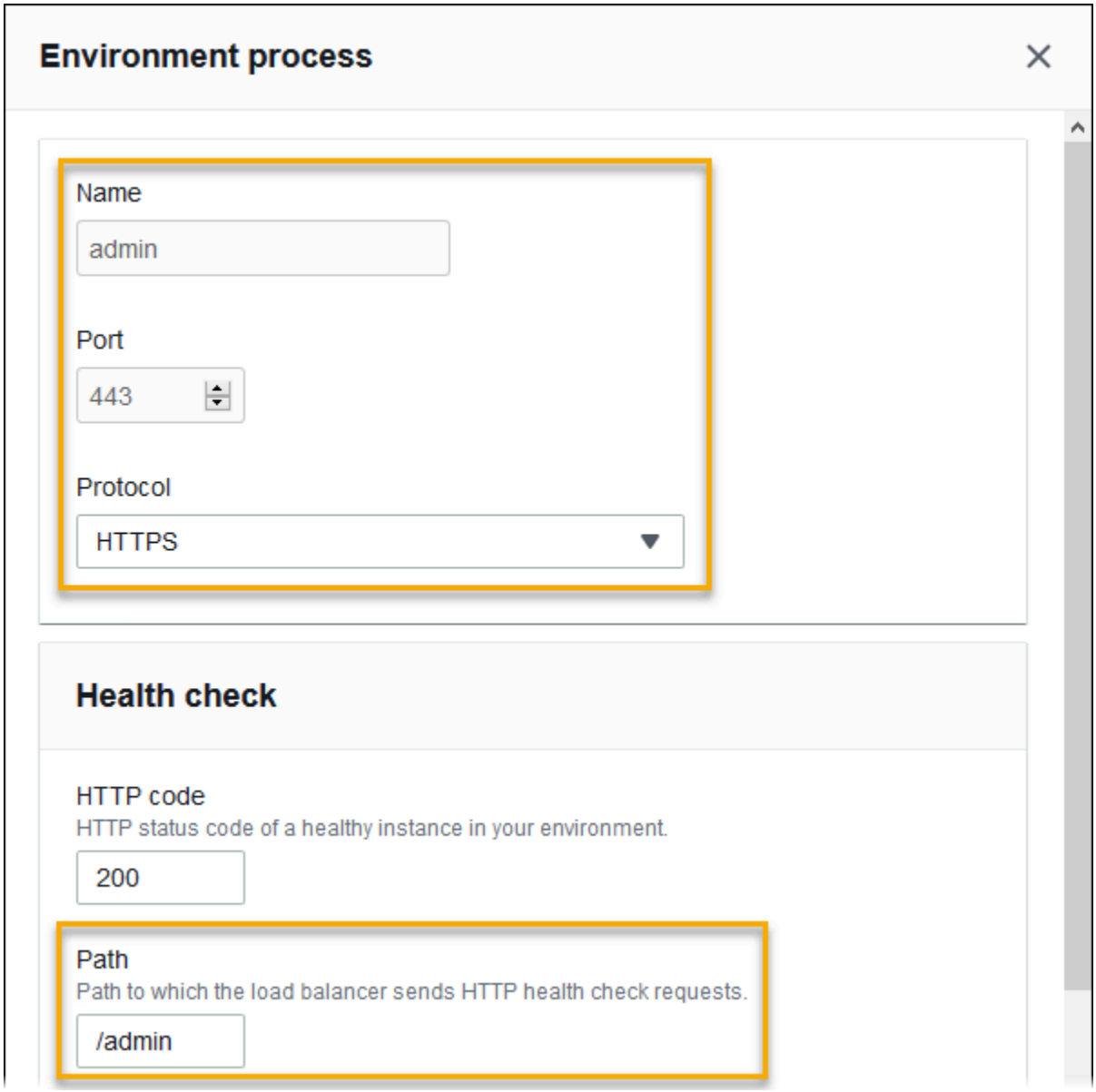
2. Disattiva il listener HTTP predefinito della porta 80. Per il listener predefinito, disattiva l'opzione Enabled (Abilitato).



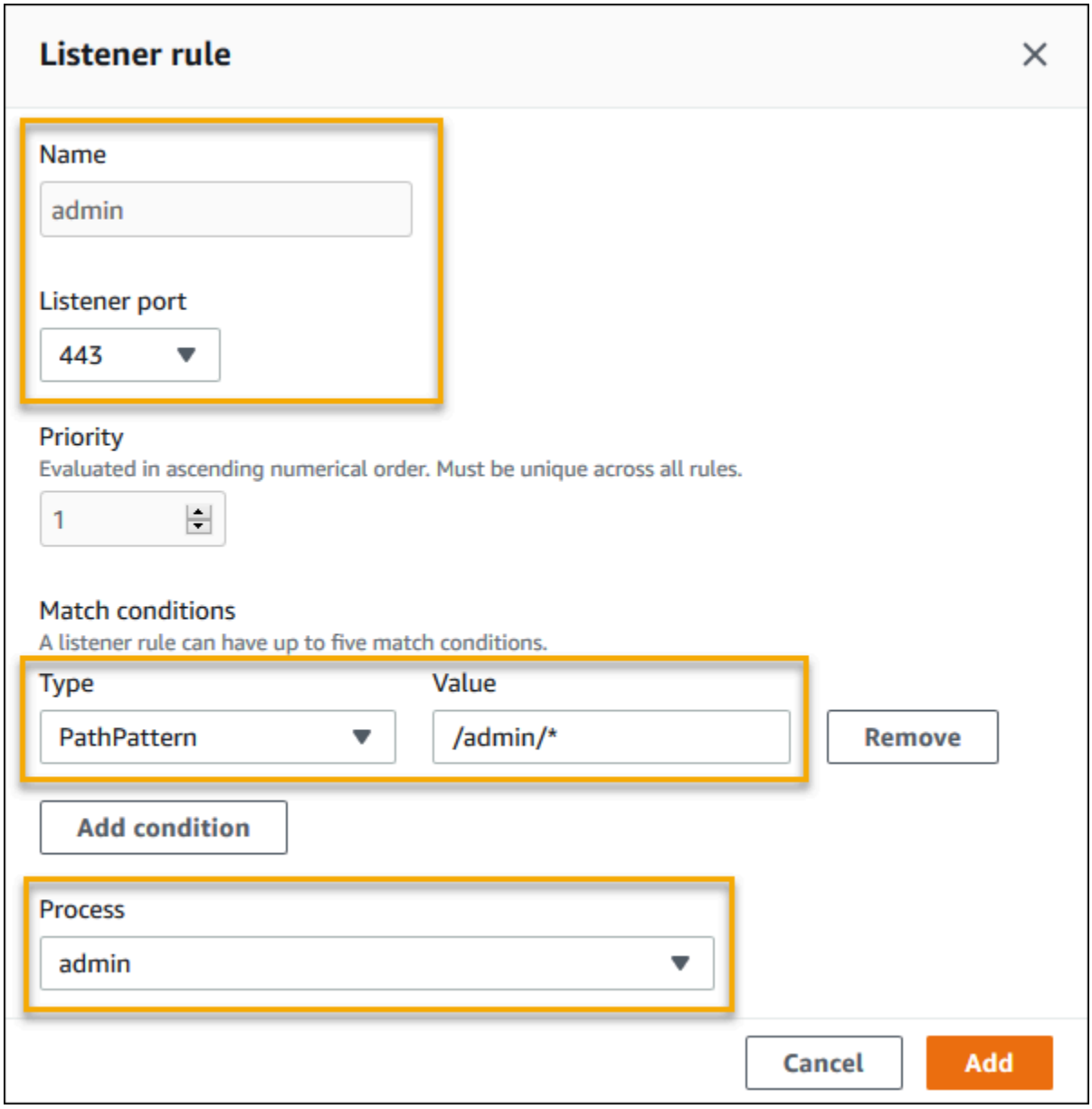
3. Configura il processo predefinito per HTTPS. Seleziona il processo predefinito, quindi in Actions (Operazioni), scegli Edit (Modifica). In Porta digita **443**. In Protocollo, seleziona **HTTPS**.



4. Aggiungi un processo admin. In Name (Nome) digitare **admin**. In Porta digita **443**. In Protocollo, seleziona **HTTPS**. In Controllo dello stato, per Percorso, digita **/admin**.



5. Aggiungi una regola per il traffico amministrativo. In Name (Nome) digitare **`admin`**. In Porta listener, digitare **`443`**. Per le condizioni Match, aggiungi un PathPatternvalore**`/admin/*`**. Per Processo, selezionare **`admin`**.



Configurazione di un sistema Application Load Balancer utilizzando l'interfaccia a riga di comando EB

Quando esegui eb create, l'interfaccia a riga di comando EB ti chiede di scegliere un tipo di load balancer.

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
```

```
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 2):
```

Puoi specificare un tipo di sistema di bilanciamento del carico anche con l'opzione `--elb-type`.

```
$ eb create test-env --elb-type application
```

Spazi dei nomi di Application Load Balancer

Le impostazioni relative ai sistemi Application Load Balancer sono disponibili nei seguenti spazi dei nomi:

- `aws:elasticbeanstalk:environment`: scegli il tipo di sistema di bilanciamento del carico per l'ambiente. Il valore di un sistema Application Load Balancer è `application`.

  Non è possibile impostare questa opzione nei file di configurazione (.Ebextensions).
- `aws:elbv2:loadbalancer`: configura i log di accesso e altre impostazioni che si applicano al sistema Application Load Balancer nel suo complesso.
- `aws:elbv2:listener`: configura i listener nel sistema Application Load Balancer. Queste impostazioni vengono mappate alle impostazioni in `aws:elb:listener` per i sistemi Classic Load Balancer.
- `aws:elbv2:listenerrule`: configura le regole che instradano il traffico verso processi diversi, a seconda del percorso della richiesta. Le regole sono univoche per i sistemi Application Load Balancer.
- `aws:elasticbeanstalk:environment:process`: configura i controlli dello stato e specifica porta e protocollo per i processi eseguiti sulle istanze del tuo ambiente. Le impostazioni relative a porta e protocollo sono mappate alle impostazioni relative a porta e protocollo dell'istanza in `aws:elb:listener` per un listener su un sistema Classic Load Balancer. Le impostazioni relative ai controlli dello stato sono mappate alle impostazioni nei namespace `aws:elb:healthcheck` e `aws:elasticbeanstalk:application`.

Example.ebextensions/ alb-access-logs .config

Il seguente file di configurazione consente di accedere ai caricamenti log per un ambiente con un sistema Application Load Balancer.

```
option_settings:
  aws:elbv2:loadbalancer:
    AccessLogsS3Bucket: amzn-s3-demo-bucket
    AccessLogsS3Enabled: 'true'
    AccessLogsS3Prefix: beanstalk-alb
```

Example.ebextensions/ alb-default-process .config

Il seguente file di configurazione modifica le impostazioni di adesione e controllo dello stato per il processo predefinito.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '15'
    HealthCheckPath: /
    HealthCheckTimeout: '5'
    HealthyThresholdCount: '3'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: HTTP
    StickinessEnabled: 'true'
    StickinessLBCookieDuration: '43200'
```

Example.ebextensions/ alb-secure-listener .config

Il seguente file di configurazione aggiunge un listener protetto e un processo corrispondente sulla porta 443.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-
east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
  aws:elasticbeanstalk:environment:process:https:
```

```
    Port: '443'
    Protocol: HTTPS
```

Example.ebextensions/ alb-admin-rule .config

Il seguente file di configurazione aggiunge un listener protetto con una regola che instrada il traffico con il percorso di richiesta `/admin` a un processo denominato `admin` in ascolto sulla porta 4443.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
    Rules: admin
    SSLCertificateArns: arn:aws:acm:us-
east-2:123456789012:certificate/21324896-0fa4-412b-bf6f-f362d6eb6dd7
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:admin:
    HealthCheckPath: /admin
    Port: '4443'
    Protocol: HTTPS
  aws:elbv2:listenerrule:admin:
    PathPatterns: /admin/*
    Priority: 1
    Process: admin
```

Configurazione di un sistema Application Load Balancer condiviso

Quando abiliti il bilanciamento del carico, il tuo AWS Elastic Beanstalk ambiente è dotato di un sistema di bilanciamento del carico Elastic Load Balancing per distribuire il traffico tra le istanze del tuo ambiente. Elastic Load Balancing supporta diversi tipi di load balancer. Per ulteriori informazioni, consulta la Guida per l'utente di Elastic Load Balancing. Elastic Beanstalk può creare automaticamente un sistema di bilanciamento del carico oppure puoi specificare un sistema di bilanciamento del carico condiviso che hai creato.

In questo argomento viene descritta la configurazione di un sistema Application Load Balancer condiviso che crei e associ all'ambiente. Consulta anche the section called “Application Load Balancer”. Per informazioni sulla configurazione di tutti i tipi di load balancer supportati da Elastic Beanstalk, consulta Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk.

> **Note**
>
> È possibile scegliere il tipo di load balancer che l'ambiente utilizza solo durante la creazione dell'ambiente. Puoi modificare le impostazioni per gestire il comportamento del sistema di bilanciamento del carico in esecuzione nell'ambiente, ma non puoi modificarne il tipo. Inoltre, non è possibile passare da un sistema di bilanciamento del carico dedicato a uno condiviso o viceversa.

Introduzione

Un sistema di bilanciamento del carico condiviso è un sistema di bilanciamento del carico che puoi creare e gestire autonomamente utilizzando il servizio Amazon Elastic Compute Cloud (Amazon EC2) e quindi utilizzato in più ambienti Elastic Beanstalk.

Quando crei un ambiente di dimensionamento con bilanciamento del carico e scegli di utilizzare un sistema Application Load Balancer, per impostazione predefinita Elastic Beanstalk crea un sistema di bilanciamento del carico dedicato all'ambiente. Per informazioni su cosa è un sistema Application Load Balancer e come funziona in un ambiente Elastic Beanstalk, consulta l'introduzione alla configurazione di un sistema Application Load Balancer per Elastic Beanstalk.

In alcune situazioni si potrebbe voler risparmiare il costo di avere più sistemi di bilanciamento del carico dedicati. Ciò può essere utile quando si dispone di più ambienti, ad esempio se l'applicazione è una suite di microservizi anziché un servizio monolitico. In questi casi è possibile scegliere di utilizzare un sistema di bilanciamento del carico condiviso.

Per utilizzare un sistema di bilanciamento del carico condiviso, devi prima crearlo in Amazon EC2 e aggiungere uno o più listener. Durante la creazione di un ambiente Elastic Beanstalk è quindi necessario fornire il sistema di bilanciamento del carico e scegliere una porta del listener. Elastic Beanstalk associa il listener al processo predefinito nell'ambiente. È possibile aggiungere regole di listener personalizzate per instradare il traffico da intestazioni e percorsi host specifici ad altri processi di ambiente.

Elastic Beanstalk aggiunge un tag al sistema di bilanciamento del carico condiviso. Il nome del tag è `elasticbeanstalk:shared-elb-environment-count`, e il suo valore è il numero di ambienti che condividono questo sistema di bilanciamento del carico.

L'utilizzo di un sistema di bilanciamento del carico condiviso è diverso dall'utilizzo di uno dedicato in vari modi.

Regarding (Motivo)	Application Load Balancer dedicato	Application Load Balancer condiviso
Gestione	Elastic Beanstalk crea e gestisce il sistema di bilanciamento del carico, i listener, le regole del listener e i processi (gruppi di destinazione). Elastic Beanstalk inoltre rimuove questi elementi quando termini l'ambiente. Elastic Beanstalk può impostare l'acquisizione del log di accesso del sistema di bilanciamento del carico, se scegli questa opzione.	Puoi creare e gestire il sistema di bilanciamento del carico e i listener al di fuori di Elastic Beanstalk. Elastic Beanstalk crea e gestisce una regola predefinita e un processo predefinito e tu aggiungi le regole e i processi. Elastic Beanstalk rimuove le regole del listener e i processi aggiunti durante la creazione dell'ambiente.
Regole dei listener	Elastic Beanstalk crea una regola predefinita per ogni listener per instradare tutto il traffico al processo predefinito del listener.	Elastic Beanstalk associa una regola predefinita solo a un listener di porta 80, se esistente. Se scegli una porta del listener predefinita diversa, devi associare la regola predefinita alla porta (la console Elastic Beanstalk e l'EB CLI consentono di eseguire questa operazione). Per risolvere i conflitti di condizione delle regole del listener in ambienti che condividono il sistema di bilanciamento del carico, Elastic Beanstalk aggiunge il CNAME dell'ambiente alla regola del listener come condizione di intestazione host. Elastic Beanstalk considera le impostazioni di priorità delle regole relative tra gli ambienti che condividono il sistema di bilanciamento del carico e le associa alle priorità assolute durante la creazione.
Gruppi di sicurezza	Elastic Beanstalk crea un gruppo di sicurezza predefinito e lo collega	È possibile configurare uno o più gruppi di sicurezza da utilizzare per il sistema di

Regarding (Motivo)	Application Load Balancer dedicato	Application Load Balancer condiviso
	al sistema di bilanciamento del carico.	bilanciamento del carico. In caso contrario , Elastic Beanstalk verifica se un gruppo di sicurezza esistente e gestito è già collegato al sistema di bilanciamento del carico. In caso contrario, Elastic Beanstalk crea un gruppo di sicurezza e lo collega al sistema di bilanciam ento del carico. Elastic Beanstalk elimina questo gruppo di sicurezza quando termina l'ultimo ambiente che condivide il sistema di bilanciamento del carico.
Aggiornam enti	Puoi aggiornare il sistema Application Load Balancer dopo la creazione dell'ambiente. È possibile modificare i listener, le regole del listener e i processi. È possibile configurare l'acquisizione del log di accesso del sistema di bilanciamento del carico.	Non è possibile utilizzare Elastic Beanstalk per configurare l'acquisizione del log di accesso in Application Load Balancer e non è possibile aggiornare i listener e le regole del listener dopo la creazione dell'ambiente. È possibile aggiornare solo i processi (gruppi di destinazi one). Per configurare l'acquisizione dei log di accesso e aggiornare i listener e le regole degli ascoltatori, usa Amazon. EC2

Configurazione di un sistema Application Load Balancer condiviso utilizzando la console Elastic Beanstalk

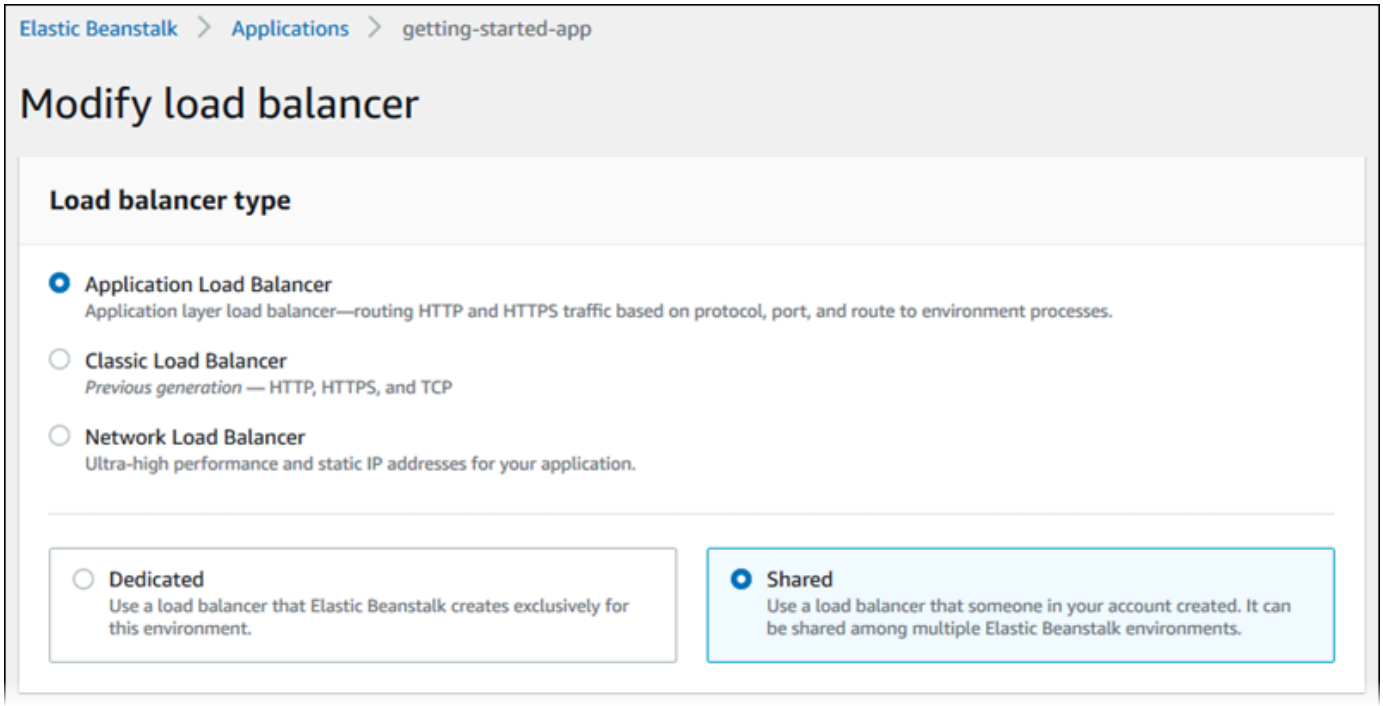
Puoi utilizzare la console Elastic Beanstalk per configurare un sistema Application Load Balancer condiviso durante la creazione dell'ambiente. È possibile selezionare uno dei sistemi di bilanciamento del carico condivisibili dell'account da utilizzare nell'ambiente, selezionare la porta del listener predefinita e configurare processi aggiuntivi e regole del listener.

Non è possibile modificare la configurazione del sistema Application Load Balancer condivisa nella console Application Load Balancer dopo la creazione dell'ambiente. Per configurare gli ascoltatori, le regole degli ascoltatori, i processi (gruppi target) e l'acquisizione dei log di accesso, usa Amazon. EC2

Per configurare un sistema Application Load Balancer nella console Elastic Beanstalk durante la creazione dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegliere Create a new environment (Crea un nuovo ambiente) per iniziare a creare l'ambiente.
4. Nella pagina principale della procedura guidata, prima di scegliere Create environment (Crea ambiente), scegli Configure more options (Configura più opzioni).
5. Scegli la configurazione preimpostata High availability (Alta disponibilità).

   In alternativa, nella categoria di configurazione Capacity (Capacità), configura un tipo di ambiente Load balanced (Con bilanciamento del carico). Per informazioni dettagliate, vedi Capacity.
6. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).
7. Seleziona l'opzione Application Load Balancer (Sistema di bilanciamento del carico applicazioni) se non è già selezionata, quindi seleziona l'opzione Shared (Condiviso).



8. Apporta alla configurazione del sistema Application Load Balancer condiviso le modifiche richieste dall'ambiente.

9. Scegli Save (Salva) e apporta eventuali altre modifiche di configurazione richieste dal tuo ambiente.
10. Seleziona Create environment (Crea ambiente).

Impostazioni di Application Load Balancer condiviso

- Application Load Balancer condiviso
- Processes
- Regole

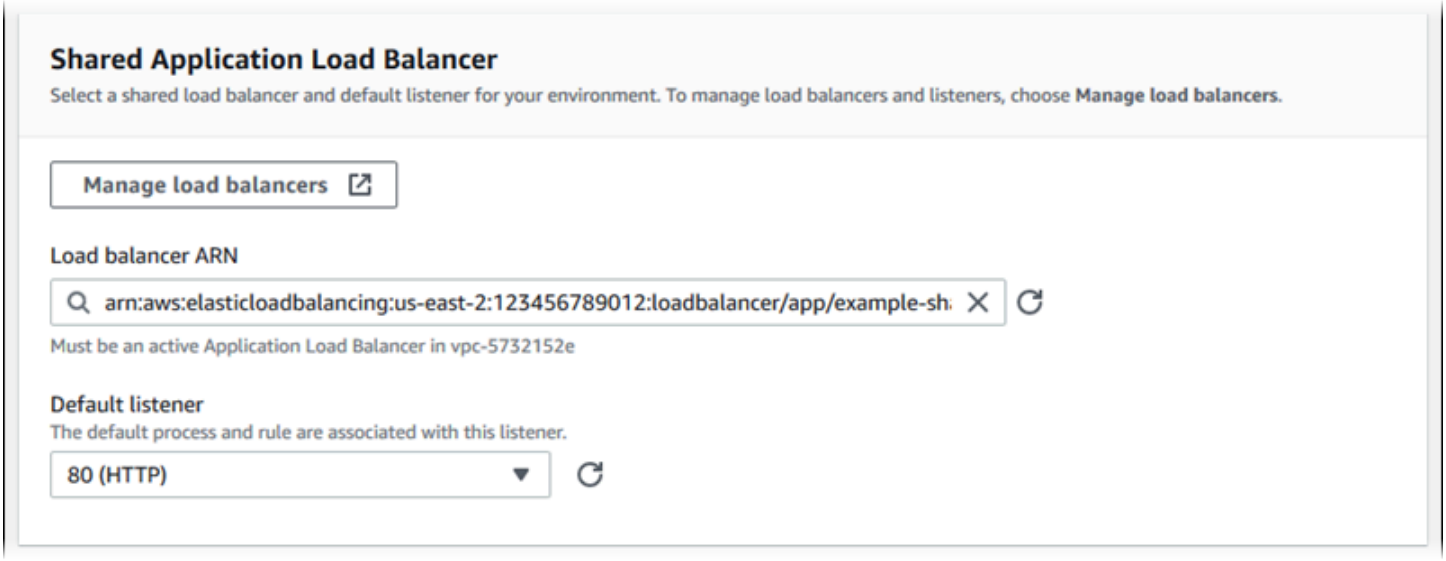
Application Load Balancer condiviso

Questa sezione ti consente di scegliere un sistema Application Load Balancer condiviso per l'ambiente e configurare il routing del traffico predefinito.

Prima di poter configurare qui un Application Load Balancer condiviso, usa Amazon EC2 per definire almeno un Application Load Balancer da condividere, con almeno un listener, nel tuo account. Se non l'hai già fatto, puoi scegliere Manage load balancers (Gestisci sistemi di bilanciamento del carico). Elastic Beanstalk apre EC2 la console Amazon in una nuova scheda del browser.

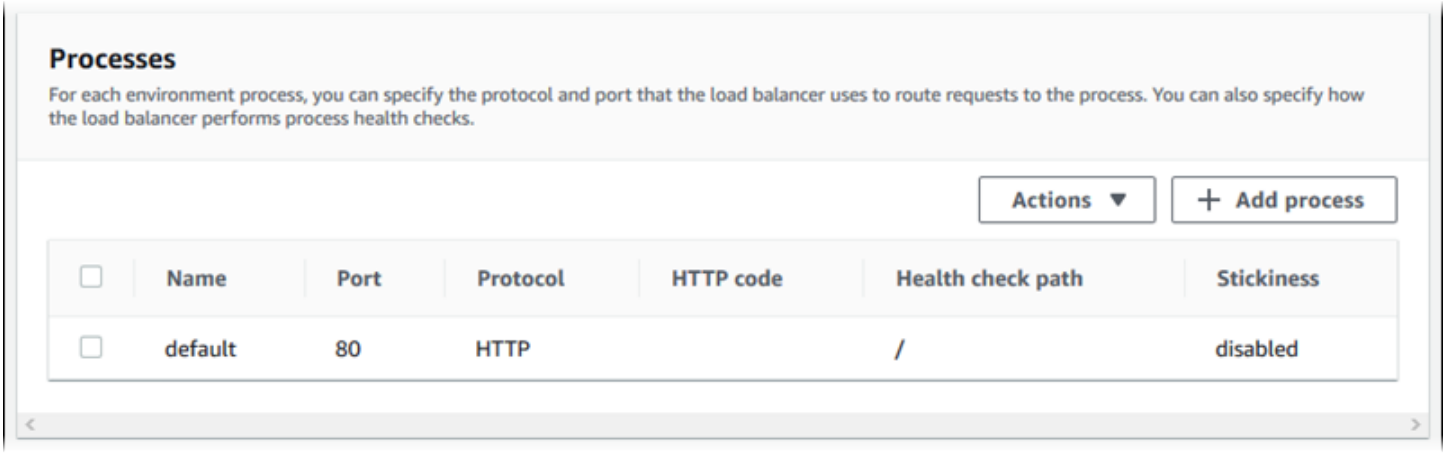
Al termine della configurazione dei sistemi di bilanciamento del carico condivisi all'esterno di Elastic Beanstalk, configura le seguenti impostazioni in questa sezione della console:

- Load balancer ARN (ARN del sistema di bilanciamento del carico): il sistema di bilanciamento del carico condiviso da utilizzare in questo ambiente. Seleziona da un elenco di sistemi di bilanciamento del carico o inserisci un Amazon Resource Name (ARN) del sistema di bilanciamento del carico.
- Default listener port (Porta predefinita del listener: la porta del listener su cui è in ascolto il sistema di bilanciamento del carico condiviso. Selezionare da un elenco di porte listener esistenti. Il traffico proveniente da questo listener con CNAME dell'ambiente nell'intestazione host viene instradato a un processo predefinito in questo ambiente.



Processes

Utilizza questo elenco per specificare i processi per il tuo sistema di bilanciamento del carico. Un processo è la destinazione di instradamento del traffico da parte dei listener. Inizialmente, l'elenco mostra il processo predefinito, che riceve il traffico dal listener predefinito.



Per configurare un processo esistente

1. Seleziona la casella di controllo accanto alla voce della tabella e successivamente Actions (Operazioni) e Edit (Modifica).
2. Utilizzare la finestra dialogo Environment process (Processo ambiente) per modificare le impostazioni, quindi scegliere Save (Salva).

Per aggiungere un processo

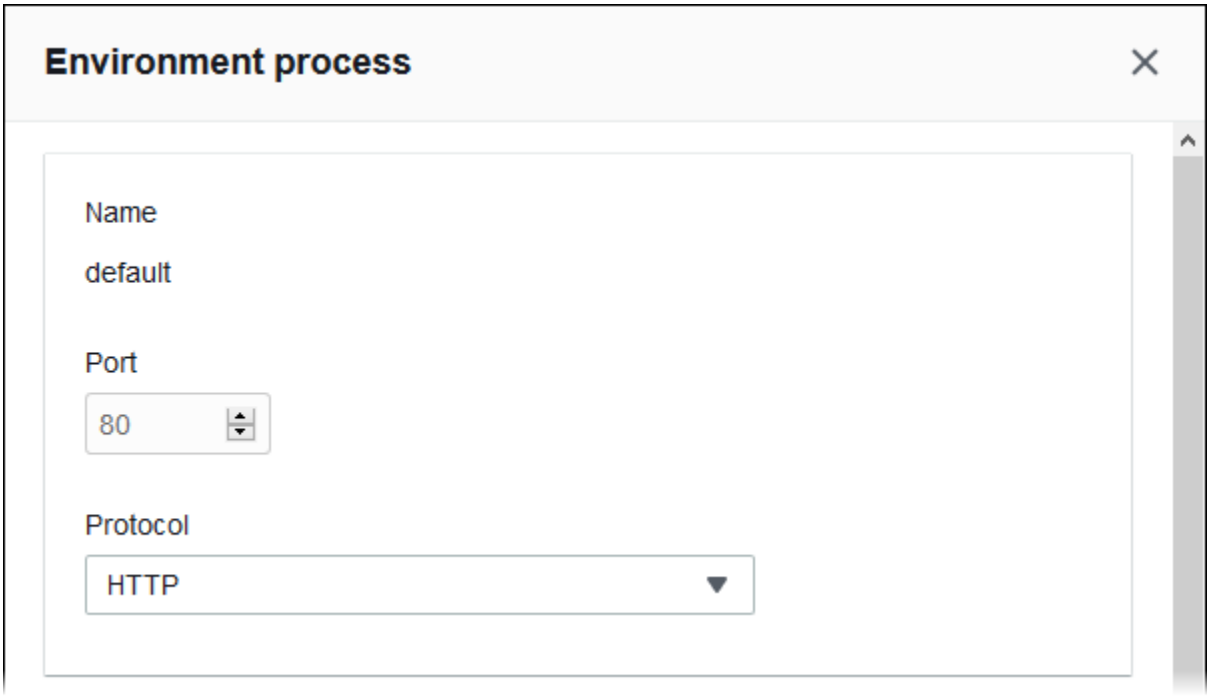
1. Scegliere Add process (Aggiungi processo).
2. Nella finestra di dialogo Environment process (Processo ambiente) configurare le impostazioni desiderate e quindi scegliere Add (Aggiungi).

Impostazioni della finestra di dialogo del processo di ambiente di Application Load Balancer

- Definizione
- Controllo dell'integrità
- Sessioni

Definizione

Utilizza queste impostazioni per definire il processo: Name (Nome), Port (Porta) e Protocol (Protocollo) attraverso cui riceverà le richieste.



Controllo dell'integrità

Utilizza le seguenti impostazioni per configurare controlli di stato del processo:

- HTTP code (Codice HTTP): codice di stato HTTP che indica un processo integro.
- Path (Percorso): percorso della richiesta di controllo dello stato del processo.

- Timeout: il periodo di attesa espresso in secondi di una risposta durante un controllo dello stato.
- Interval (Intervallo): il periodo di tempo espresso in secondi che decorre tra i controlli dello stato di una singola istanza. L'intervallo deve essere maggiore del timeout.
- Unhealthy threshold (Soglia di mancata integrità), Healthy threshold (Soglia di integrità): il numero di controlli dello stato consecutivi che non devono o devono essere superati rispettivamente prima che Elastic Load Balancing modifichi lo stato di integrità di un'istanza.
- Deregistration delay (Ritardo annullamento della registrazione): il periodo di tempo espresso in secondi necessario al completamento delle richieste prima che venga annullata la registrazione di un'istanza.

Health check

HTTP code
HTTP status code of a healthy instance in your environment.

Path
Path to which the load balancer sends HTTP health check requests.

/

Timeout
Amount of time to wait for a health check response.

5 seconds

Interval
Amount of time between health checks of an individual instance. The interval must be greater than the timeout.

15 seconds

Unhealthy threshold
The number of consecutive health check failures required to designate the instance as unhealthy.

5 requests

Healthy threshold
The number of consecutive successful health checks required to designate the instance as healthy.

3 requests

Deregistration delay
Amount of time to wait for active requests to complete before deregistering.

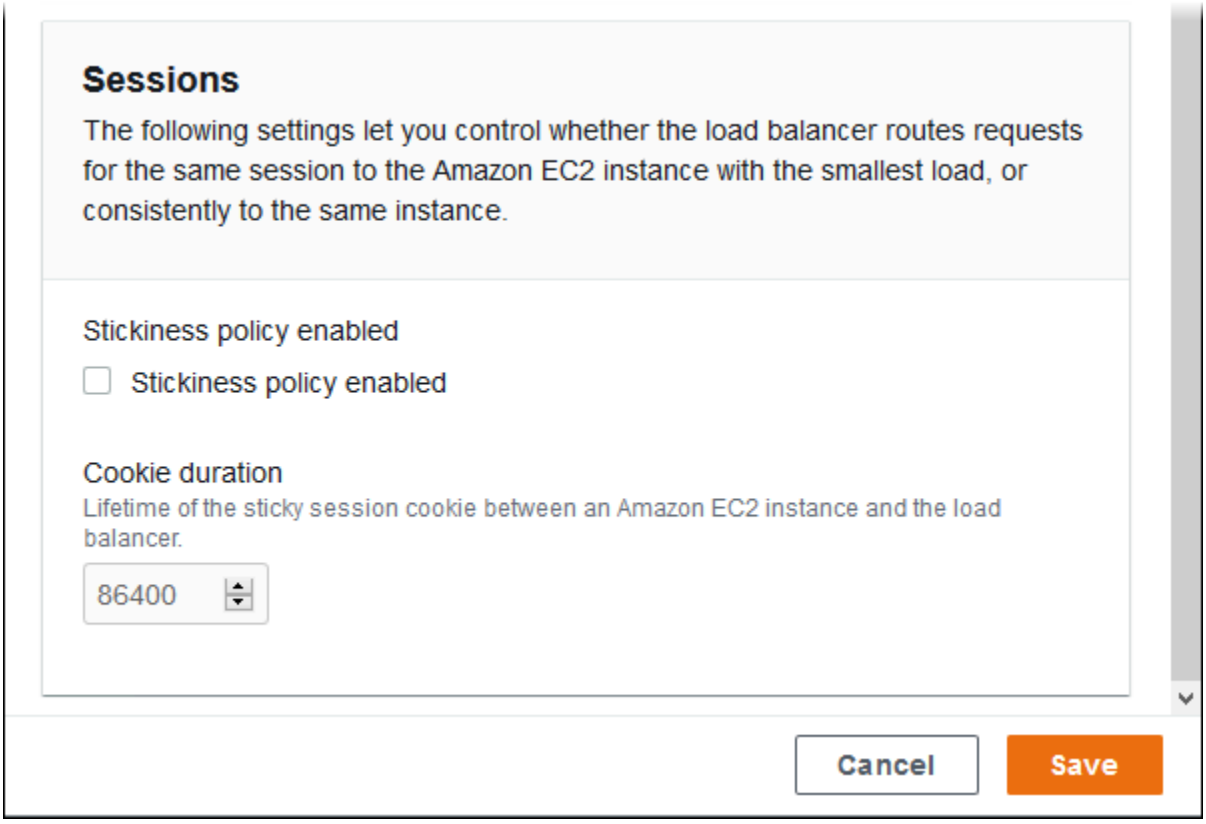
20 seconds

> **Note**
>
> Il controllo dello stato di Elastic Load Balancing non influisce sul comportamento del controllo dello stato del gruppo Auto Scaling di un ambiente. Le istanze che non superano un controllo dello stato di Elastic Load Balancing non vengono sostituite automaticamente da Amazon Auto EC2 Scaling a meno che non configuri manualmente Amazon Auto EC2 Scaling a tale scopo. Per informazioni dettagliate, vedi Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk.

Per ulteriori informazioni sui controlli dello stato e su come influiscono sullo stato di integrità generale dell'ambiente, consulta Report dello stato di base.

Sessioni

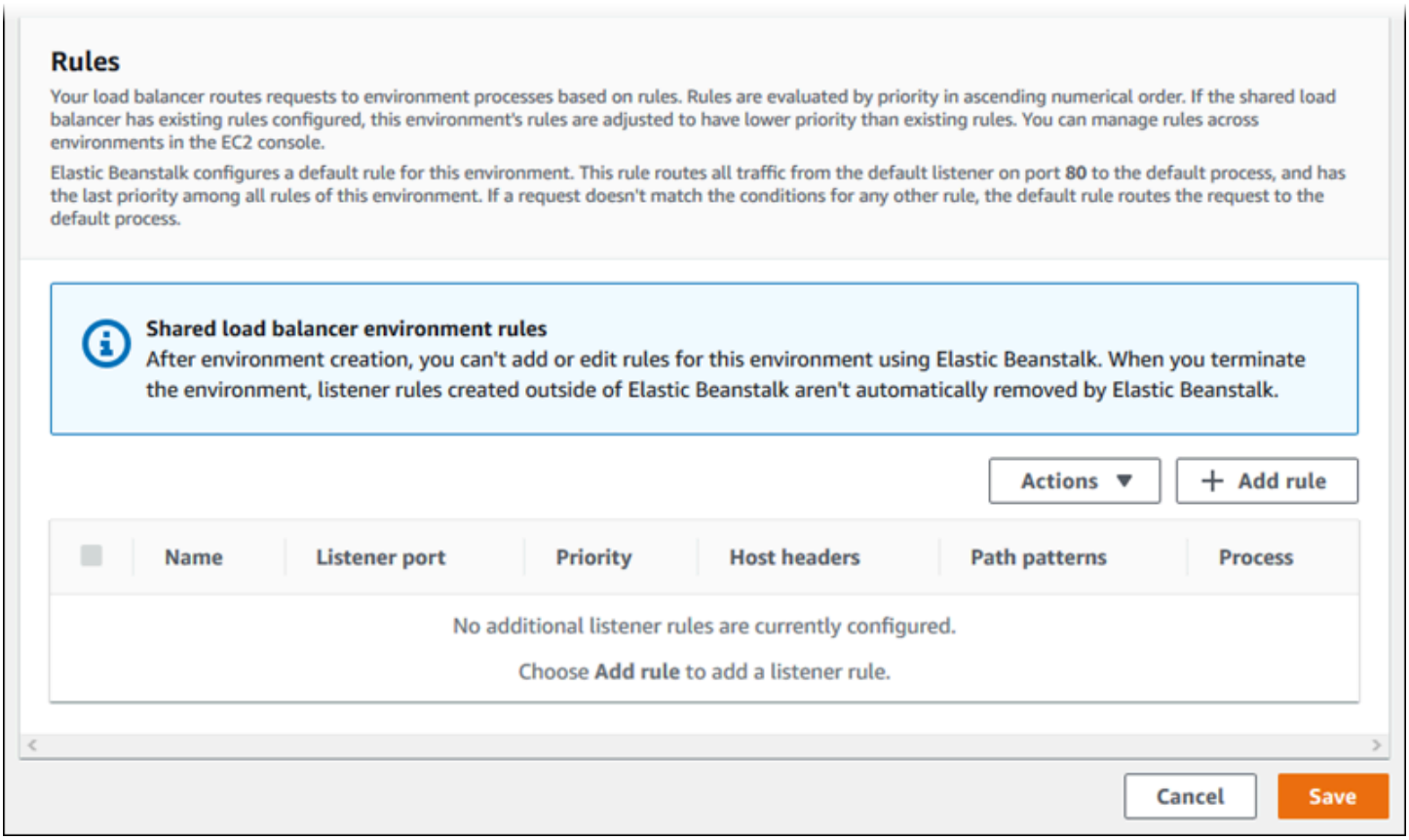
Seleziona o deseleziona la casella Stickiness policy enabled (Policy di adesione abilitata) per attivare o disattivare le sticky session. Usa Cookie duration (Durata del cookie) per configurare la durata di una sticky session, fino a **`604800`** secondi.



Regole

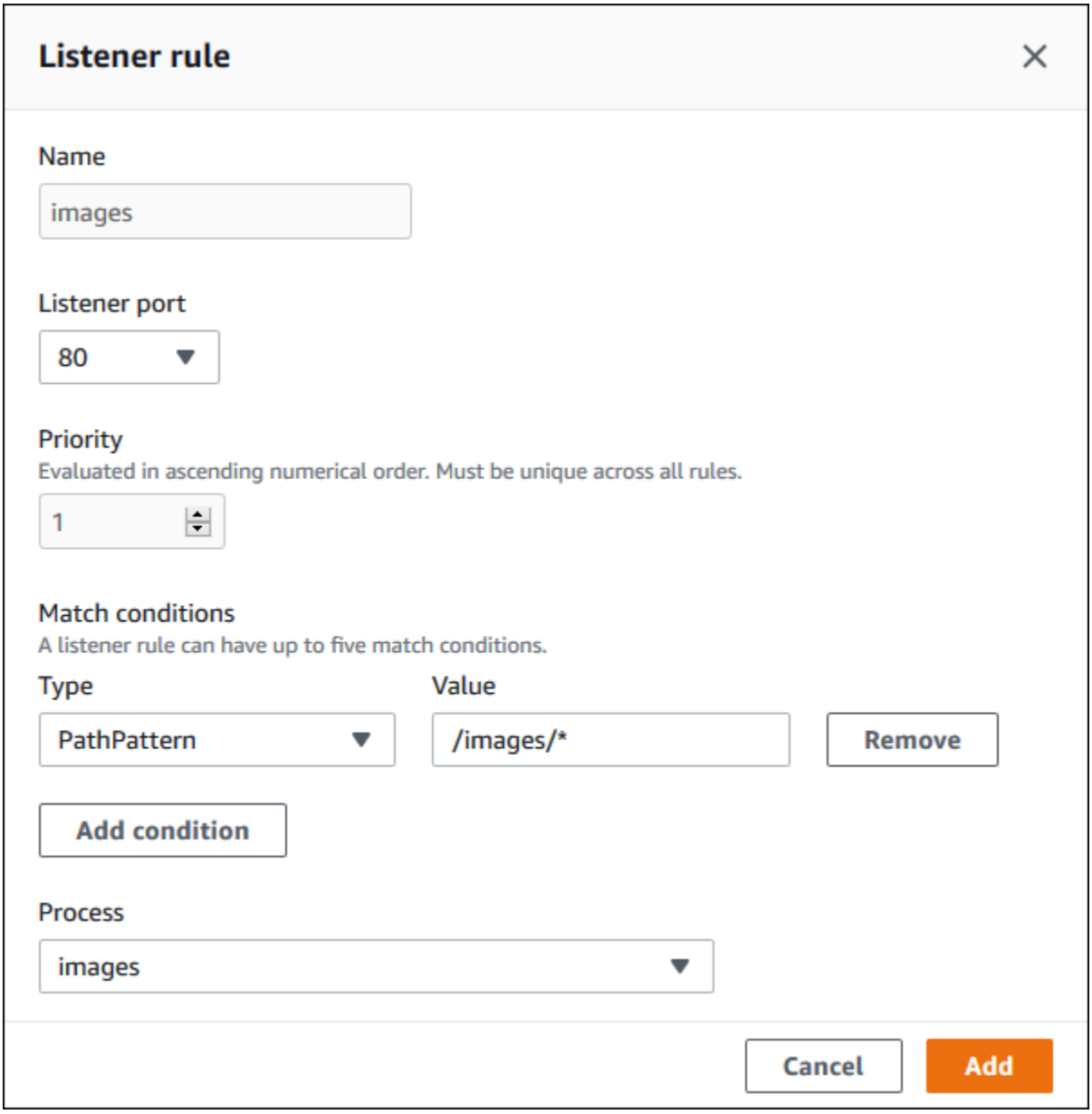
Utilizza questo elenco per specificare le regole dei listener personalizzate per il sistema di bilanciamento del carico. Una regola associa le richieste ricevute dal listener relative a un determinato modello di percorso a un processo di destinazione. Ogni listener può avere più regole, instradando le richieste su percorsi diversi a processi diversi su istanze dei diversi ambienti che condividono il listener.

Le regole hanno priorità numeriche che stabiliscono la precedenza in cui vengono applicate alle richieste in entrata. Elastic Beanstalk aggiunge una regola predefinita che instrada tutto il traffico del listener predefinito al processo predefinito del nuovo ambiente. La precedenza della regola predefinita è la più bassa: viene applicata nel caso in cui non ci siano altre regole per lo stesso listener che corrispondono alla richiesta in entrata. Inizialmente, se non sono state aggiunte regole personalizzate, l'elenco è vuoto. La regola predefinita non viene visualizzata.



Puoi modificare le impostazioni di una regola esistente o aggiungere una nuova regola. Per iniziare a modificare una regola dell'elenco o per aggiungere una regola, utilizza la stessa procedura indicata per l'elenco dei processi. Viene visualizzata la finestra di dialogo Listener rule (Regola listener), con le impostazioni seguenti:

- Name (Nome): nome della regola.
- Listener port (Porta listener): porta del listener a cui si applica la regola.
- Priority (Priorità): priorità della regola. Un numero di priorità inferiore ha la precedenza. Le priorità delle regole del listener devono essere uniche. Elastic Beanstalk tratta le priorità delle regole come relative tra gli ambienti di condivisione e le mappa alle priorità assolute durante la creazione.
- Match conditions (Condizioni di corrispondenza): elenco delle condizioni dell'URL della richiesta a cui si applica la regola. Esistono due tipi di condizioni: HostHeader(la parte del dominio dell'URL) e PathPattern(la parte del percorso dell'URL). Una condizione è riservata al sottodominio di ambiente ed è possibile aggiungere fino a quattro condizioni. Ogni valore di condizione ha una lunghezza massima di 128 caratteri e può includere caratteri jolly.
- Process (Processo): processo al quale il sistema di bilanciamento del carico instrada le richieste che soddisfano la regola.



Esempio: utilizzare un Application Load Balancer condiviso per un'applicazione sicura micro-service-based

In questo esempio l'applicazione è costituita da diversi microservizi, ognuno implementato come un ambiente Elastic Beanstalk. Inoltre, è necessaria la crittografia end-to-end del traffico. Dimostreremo uno degli ambienti di micro-servizi, che ha un processo principale per le richieste degli utenti e un processo separato per la gestione delle richieste amministrative.

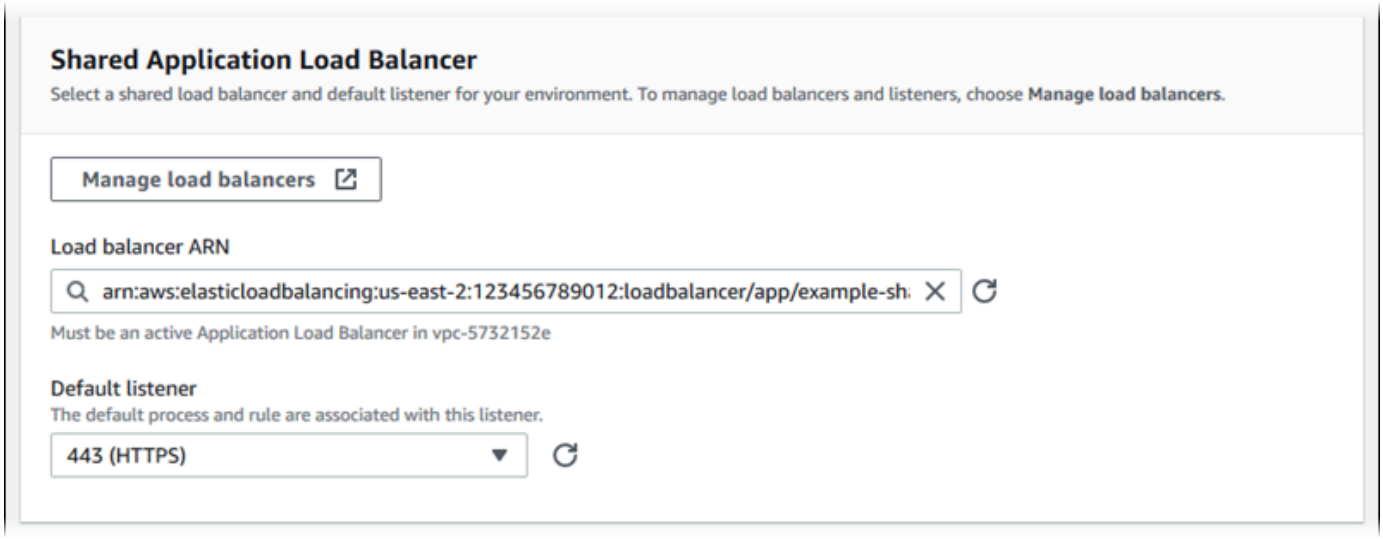
Per soddisfare questi requisiti, usa Amazon EC2 per creare un Application Load Balancer da condividere tra i tuoi micro servizi. Aggiungere un listener sicuro sulla porta 443 e il protocollo HTTPS. Quindi aggiungi più certificati SSL al listener: uno per dominio dei microservizi. Per

informazioni dettagliate sulla creazione di Application Load Balancer e del listener protetto, consulta Creazione di un sistema Application Load Balancer e Creazione di un listener HTTPS per il sistema Application Load Balancer nella Guida per l'utente di Application Load Balancer.

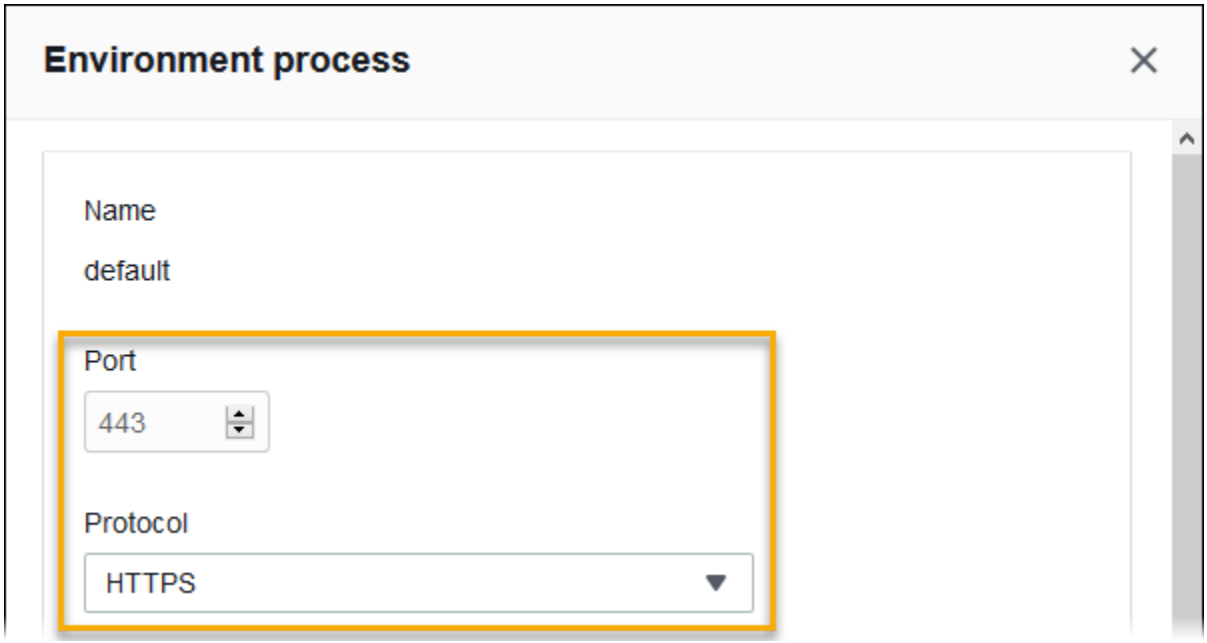
In Elastic Beanstalk configura ogni ambiente di microservizi per utilizzare il sistema Application Load Balancer condiviso e impostare la porta del listener predefinita su 443. Nel caso del particolare ambiente che stiamo dimostrando qui, indicare che il processo predefinito ascolta la porta 443 su HTTPS e aggiungere un processo e una regola di listener per il traffico di amministrazione su un percorso diverso.

Configurazione del sistema di bilanciamento del carico condiviso per questo esempio

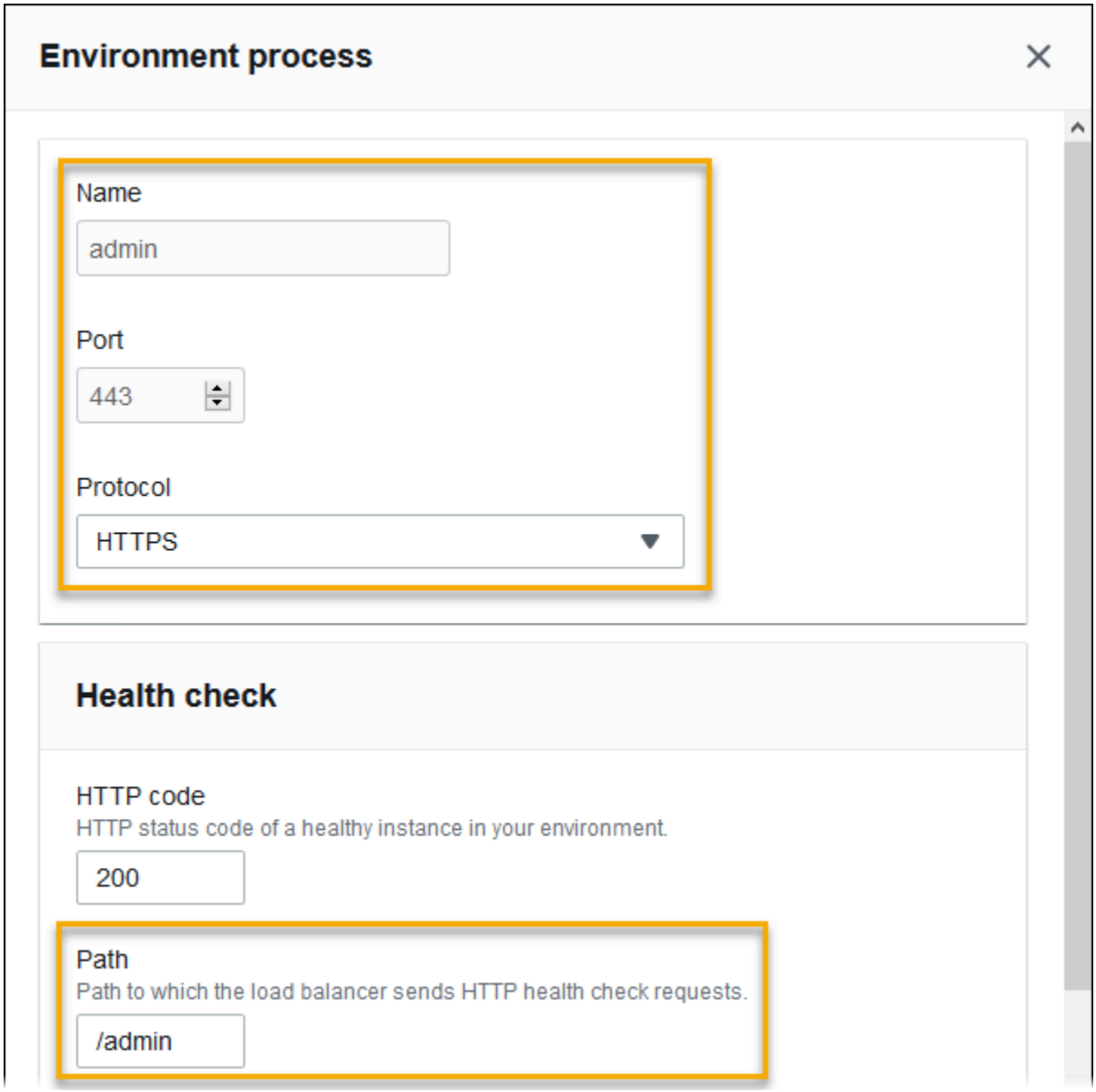
1. Nella sezione Shared Application Load Balancer (Application Load Balancer condiviso) seleziona il sistema di bilanciamento del carico e quindi seleziona **443** per Default listener port (Porta del listener predefinita). La porta del listener dovrebbe essere già selezionata se è l'unico listener che dispone del sistema di bilanciamento del carico.



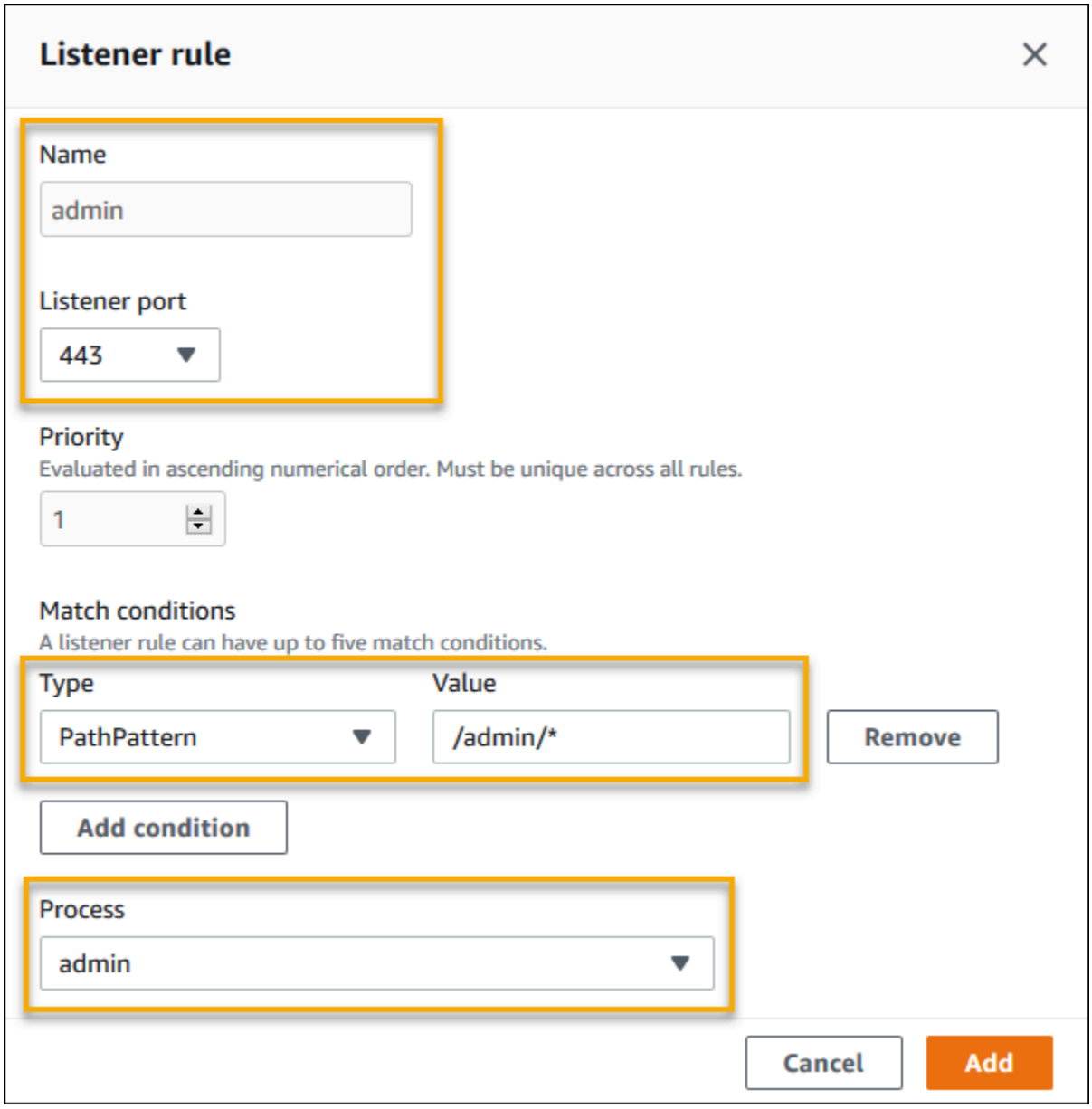
2. Configura il processo predefinito per HTTPS. Seleziona il processo predefinito, quindi in Actions (Operazioni), scegli Edit (Modifica). Per Port (Porta), immettere **443**. In Protocollo, seleziona **HTTPS**.



3. Aggiungi un processo admin. In Nome, inserisci **`admin`**. Per Port (Porta), immettere **`443`**. In Protocollo, seleziona **`HTTPS`**. In Health check (Controllo dello stato), per Path (Percorso), digita **`/admin`**.



4. Aggiungi una regola per il traffico amministrativo. In Nome, inserisci **`admin`**. In Listener port (Porta listener), immettere **`443`**. Per le condizioni Match, aggiungi PathPatternun valore**`/admin/*`**. Per Processo, selezionare **`admin`**.



Configurazione di un sistema Application Load Balancer condiviso utilizzando l'interfaccia a riga di comando EB

Quando esegui eb create, l'interfaccia a riga di comando EB ti chiede di scegliere un tipo di load balancer. Se scegli `application` (impostazione predefinita) e se l'account dispone di almeno un sistema Application Load Balancer condivisibile, l'interfaccia a riga di comando EB chiede anche se vuoi utilizzare un sistema Application Load Balancer condiviso. Se si risponde **y**, viene richiesto anche di selezionare il sistema di bilanciamento del carico e la porta predefinita.

```
$ eb create
```

```
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 2):

Your account has one or more sharable load balancers. Would you like your new
 environment to use a shared load balancer?(y/N) y

Select a shared load balancer
1)MySharedALB1 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB1/6d69caa75b15d46e
2)MySharedALB2 - arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/
MySharedALB2/e574ea4c37ad2ec8
(default is 1): 2

Select a listener port for your shared load balancer
1) 80
2) 100
3) 443
(default is 1): 3
```

È inoltre possibile specificare un sistema di bilanciamento del carico condiviso utilizzando le opzioni dei comandi.

```
$ eb create test-env --elb-type application --shared-lb MySharedALB2 --shared-lb-
port 443
```

Spazi dei nomi di Application Load Balancer condiviso

Le impostazioni relative ai sistemi Application Load Balancer condivisi sono disponibili nei seguenti spazi dei nomi:

- `aws:elasticbeanstalk:environment`: scegli il tipo di sistema di bilanciamento del carico per l'ambiente e indica a Elastic Beanstalk che verrà utilizzato un sistema di bilanciamento del carico condiviso.

Non è possibile impostare queste due opzioni nei file di configurazione (.Ebextensions).

- `aws:elbv2:loadbalancer`: configura l'ARN del sistema Application Load Balancer condiviso e i gruppi di sicurezza.
- `aws:elbv2:listener`: associa i listener del sistema Application Load Balancer condiviso ai processi di ambiente elencando le regole del listener.
- `aws:elbv2:listenerrule`: configura le regole del listener che instradano il traffico verso processi diversi, a seconda del percorso della richiesta. Le regole sono univoche per i sistemi Application Load Balancer, dedicati e condivisi.
- `aws:elasticbeanstalk:environment:process`: configura i controlli dello stato e specifica porta e protocollo per i processi eseguiti sulle istanze del tuo ambiente.

Example.ebextensions/ application-load-balancer-shared .config

Per iniziare a utilizzare un sistema Application Load Balancer condiviso, usa la console Elastic Beanstalk, l'interfaccia a riga di comando EB o l'API per impostare il tipo di sistema di bilanciamento del carico su `application` e scegli di utilizzare un sistema di bilanciamento del carico condiviso. Utilizzare un file di configurazione per configurare il sistema di bilanciamento del carico condiviso.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

> **Note**
>
> È possibile configurare questa opzione solo durante la creazione dell'ambiente.

Example.ebextensions/ alb-shared-secure-listener .config

Il seguente file di configurazione seleziona un listener sicuro predefinito sulla porta 443 per il sistema di bilanciamento del carico condiviso e imposta il processo predefinito per ascoltare sulla porta 443.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

```
  aws:elbv2:listener:443:
    rules: default
  aws:elasticbeanstalk:environment:process:default:
    Port: '443'
    Protocol: HTTPS
```

Example.ebextensions/ alb-shared-admin-rule .config

Il seguente file di configurazione si basa sull'esempio precedente e aggiunge una regola che instrada il traffico con un percorso di richiesta di `/admin` a un processo denominato `admin` in ascolto sulla porta 4443.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
  aws:elbv2:listener:443:
    rules: default,admin
  aws:elasticbeanstalk:environment:process:default:
    Port: '443'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:admin:
    HealthCheckPath: /admin
    Port: '4443'
    Protocol: HTTPS
  aws:elbv2:listenerrule:admin:
    PathPatterns: /admin/*
    Priority: 1
    Process: admin
```

Configurazione di un sistema Network Load Balancer

Quando abiliti il bilanciamento del carico, il tuo AWS Elastic Beanstalk ambiente è dotato di un sistema di bilanciamento del carico Elastic Load Balancing per distribuire il traffico tra le istanze del tuo ambiente. Elastic Load Balancing supporta diversi tipi di load balancer. Per ulteriori informazioni, consulta la Guida per l'utente di Elastic Load Balancing. Elastic Beanstalk può creare un load balancer automaticamente o ti consente di specificare un load balancer condiviso creato da te.

In questo argomento viene descritta la configurazione di un sistema Network Load Balancer creato da Elastic Beanstalk e dedicato all'ambiente in uso. Per informazioni sulla configurazione di tutti i tipi

di load balancer supportati da Elastic Beanstalk, consulta Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk.

> **Note**
>
> È possibile scegliere il tipo di load balancer che l'ambiente utilizza solo durante la creazione dell'ambiente. Puoi modificare le impostazioni per gestire il comportamento del sistema di bilanciamento del carico in esecuzione nell'ambiente, ma non puoi modificarne il tipo.

Introduzione

Con un sistema Network Load Balancer il listener predefinito accetta le richieste TCP sulla porta 80 e le distribuisce alle istanze nel tuo ambiente. È possibile configurare il comportamento dei controlli dello stato e configurare la porta listener o aggiungere un listener su un'altra porta.

> **Note**
>
> A differenza di un sistema Classic Load Balancer o Application Load Balancer, un sistema Network Load Balancer non può avere listener HTTP o HTTPS a livello di applicazione (livello 7). Supporta i listener TCP e TLS del livello di trasporto (livello 4). Per stabilire connessioni HTTPS sicure tra i client Web e l'ambiente, è possibile configurare un listener TLS con un certificato SSL e una politica di sicurezza. Puoi anche utilizzare un listener TCP e installare un certificato autofirmato sulle istanze dell'ambiente e configurare le istanze per l'ascolto sulla porta appropriata (in genere 443) e gestire direttamente le connessioni HTTPS. La configurazione varia in base alla piattaforma. Per istruzioni, consulta Configurazione della terminazione HTTPS sull'istanza. Quindi configura il tuo sistema Network Load Balancer per aggiungere un listener che mappi un processo in ascolto sulla porta corretta.

Un sistema Network Load Balancer supporta i controlli dello stato attivi. Questi controlli sono basati su messaggi verso il percorso root (/). Un sistema Network Load Balancer supporta anche i controlli dello stato passivi. Rileva automaticamente le istanze di back-end difettose e consente di instradare il traffico solo verso istanze corrette.

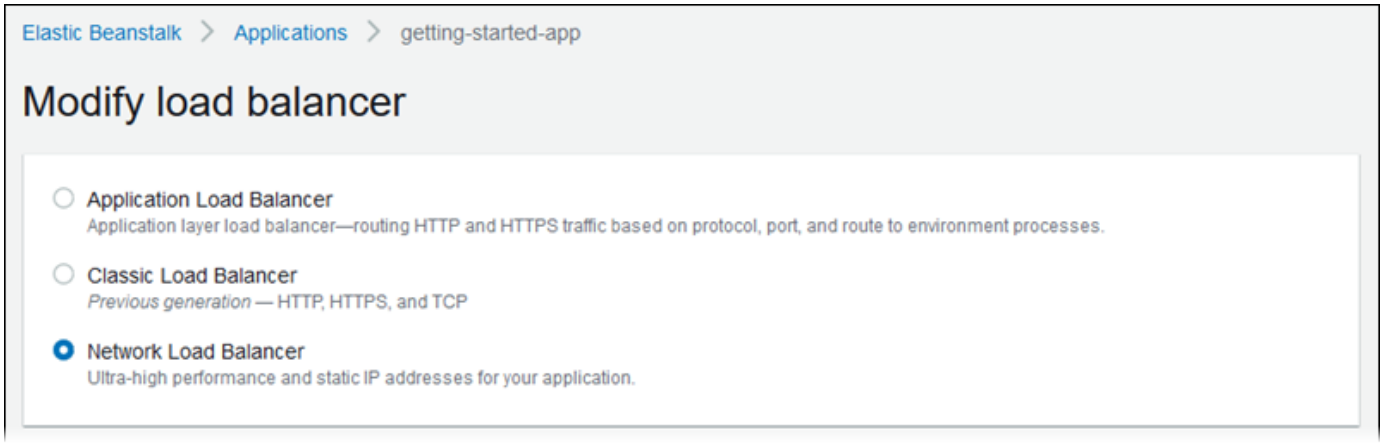
Configurazione di un sistema Network Load Balancer utilizzando la console Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk per configurare i listener e i processi di Network Load Balancer durante la creazione dell'ambiente o in un secondo momento quando l'ambiente è in esecuzione.

Per configurare un sistema Network Load Balancer nella console Elastic Beanstalk durante la creazione dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegliere Create a new environment (Crea un nuovo ambiente) per iniziare a creare l'ambiente.
4. Nella pagina principale della procedura guidata, prima di scegliere Create environment (Crea ambiente), scegli Configure more options (Configura più opzioni).
5. Scegli la configurazione preimpostata High availability (Alta disponibilità).

   In alternativa, nella categoria di configurazione Capacity (Capacità), configura un tipo di ambiente Load balanced (Con bilanciamento del carico). Per informazioni dettagliate, vedi Capacity.
6. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).
7. Seleziona l'opzione Network Load Balancer (Sistema di bilanciamento del carico di rete) se non è già selezionata.



8. Apporta alla configurazione di Network Load Balancer le modifiche richieste dall'ambiente.

9. Scegli Save (Salva) e apporta eventuali altre modifiche di configurazione richieste dal tuo ambiente.
10. Seleziona Create environment (Crea ambiente).

Per configurare un sistema Network Load Balancer di un ambiente in esecuzione nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).

> **Note**
>
> Se la categoria di configurazione di Load balancer (Sistema di bilanciamento del carico) non presenta un pulsante Edit (Modifica), il tuo ambiente non dispone di un load balancer. Per informazioni su come impostarne uno, consultare Modifica del tipo di ambiente.

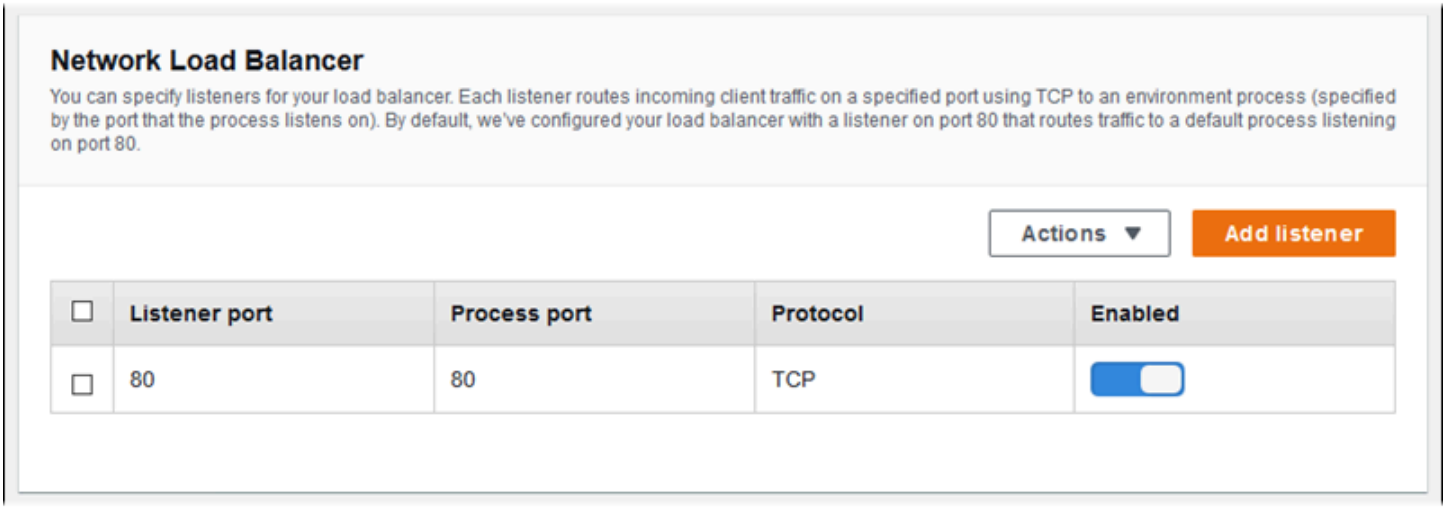
5. Apporta alla configurazione di Network Load Balancer le modifiche richieste dall'ambiente.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Impostazioni di Network Load Balancer

- Listener
- Processes

Listener

Utilizza questo elenco per specificare i listener per il tuo load balancer. Ogni listener instrada il traffico client in entrata su una determinata porta verso un processo sulle istanze. Inizialmente, l'elenco mostra il listener predefinito che indirizza il traffico in entrata sulla porta 80 verso un processo denominato default (predefinito), che ascolta la porta 80.



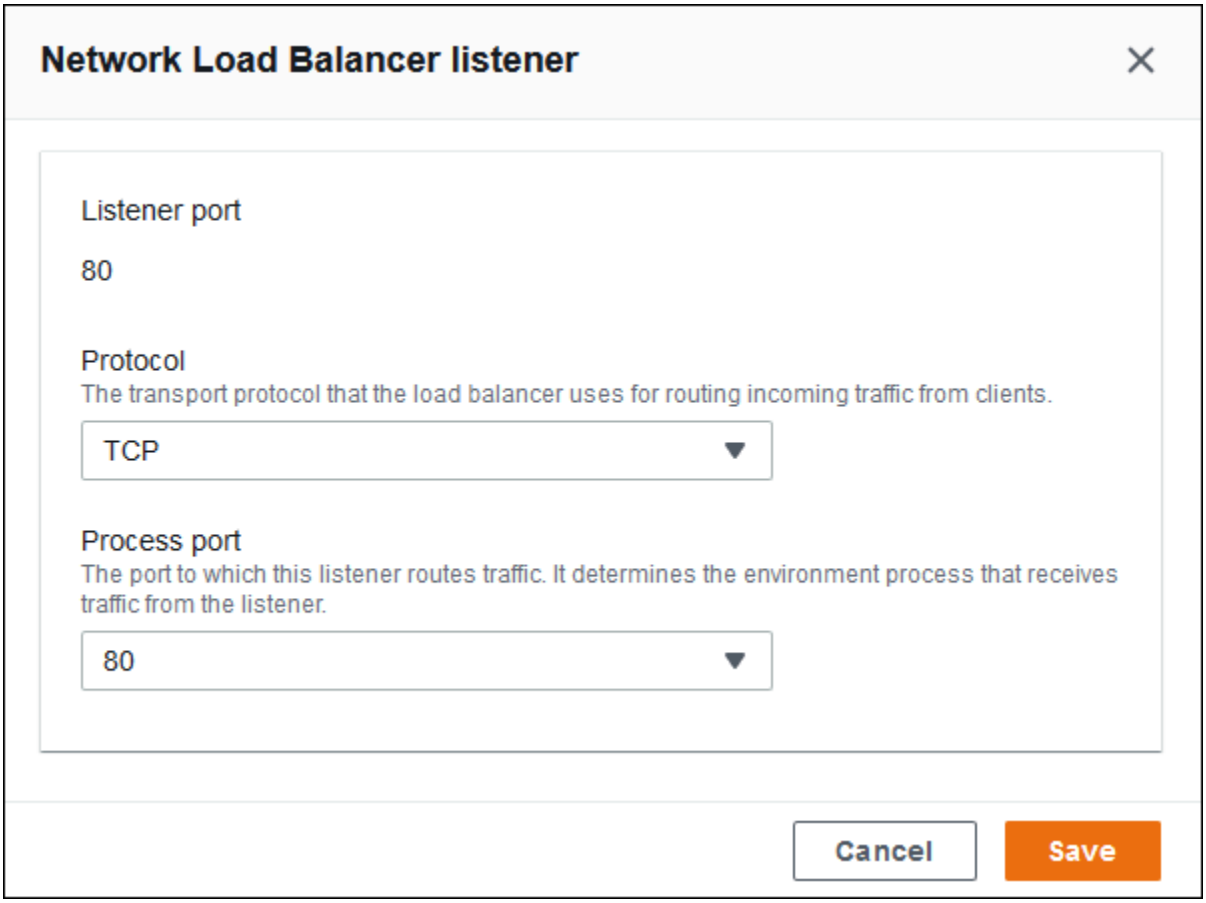
Configurazione di un listener esistente

1. Seleziona la casella di controllo accanto alla voce della tabella e successivamente Actions (Operazioni) e Edit (Modifica).
2. Utilizza la finestra di dialogo Network Load Balancer listener (Listener del Network Load Balancer) per modificare le impostazioni, quindi seleziona Save (Salva).

Aggiunta di un listener

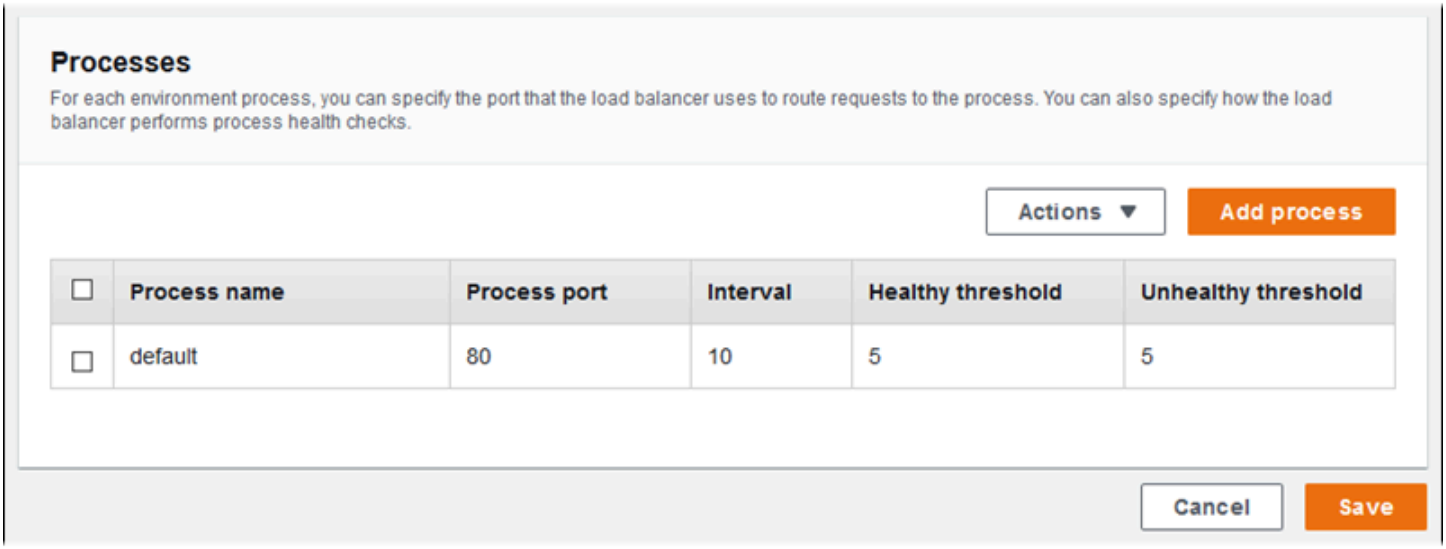
1. Scegli Add listener (Aggiungi listener).
2. Nella finestra di dialogo Network Load Balancer listener (Listener del Network Load Balancer), configura le impostazioni e scegli Add (Aggiungi).

Utilizza la finestra di dialogo Network Load Balancer listener (Listener del Network Load Balancer) per configurare la porta su cui il listener ascolta il traffico e il processo in cui si desidera instradare il traffico (specificato dalla porta su cui il processo è in ascolto).



Processes

Utilizza questo elenco per specificare i processi per il tuo sistema di bilanciamento del carico. Un processo è la destinazione di instradamento del traffico da parte dei listener. Ogni listener instrada il traffico client in entrata su una determinata porta verso un processo sulle istanze. Inizialmente, l'elenco mostra il processo predefinito, che intercetta il traffico in entrata sulla porta 80.



Puoi modificare le impostazioni di un processo esistente o aggiungerne uno nuovo. Per iniziare a modificare un processo dell'elenco o per aggiungere un processo, utilizza la stessa procedura indicata per l'elenco di listener. Viene visualizzata la finestra di dialogo Environment process (Processo ambiente).

Impostazioni della finestra di dialogo del processo di ambiente di Network Load Balancer

- Definizione
- Controllo dell'integrità

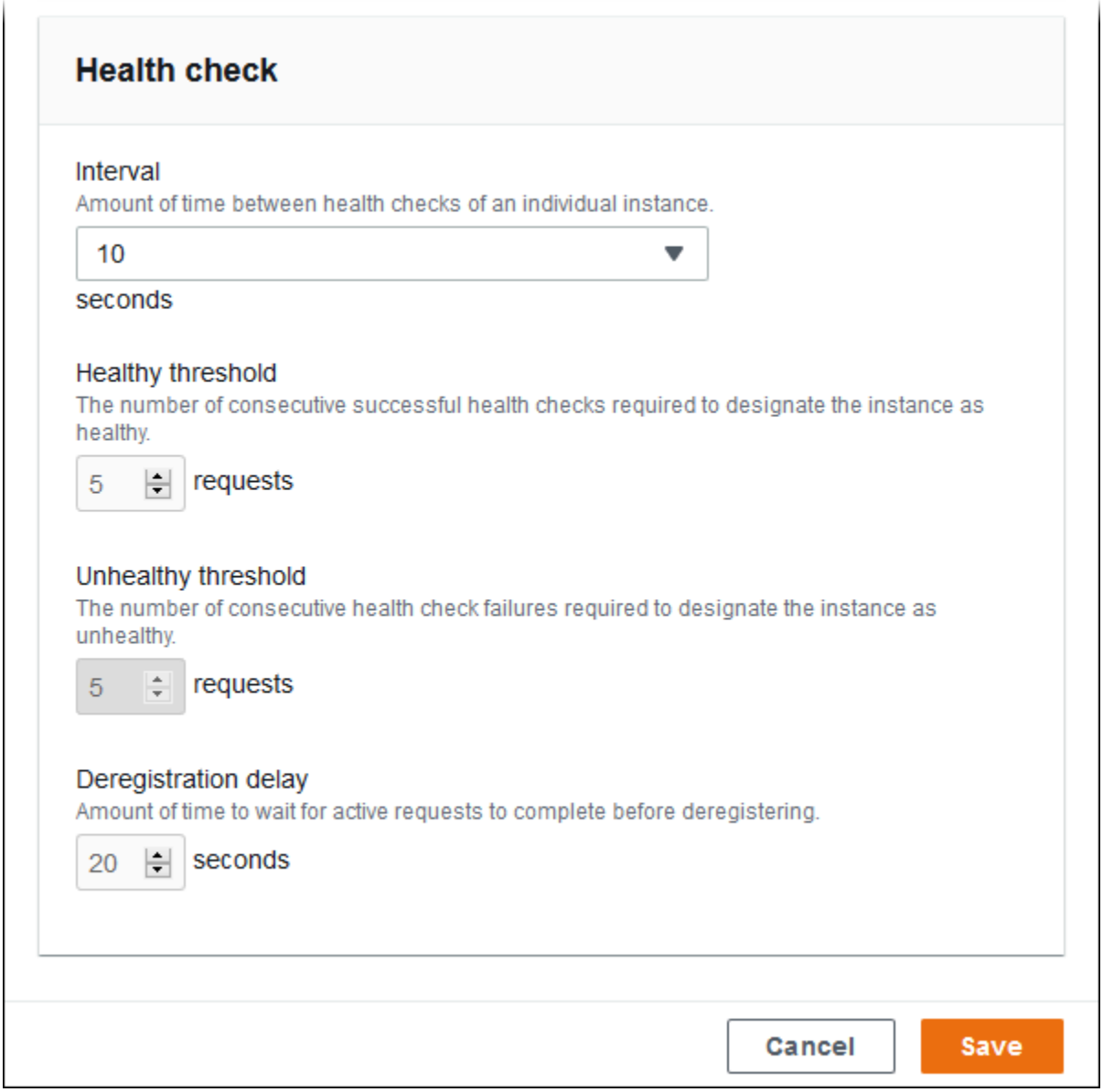
Definizione

Utilizza queste impostazioni per definire il processo: Name (Nome), Process port (Porta processo) attraverso cui riceverà le richieste.

Controllo dell’integrità

Utilizza le seguenti impostazioni per configurare controlli di stato del processo:

- Interval (Intervallo): il periodo di tempo espresso in secondi che decorre tra i controlli dello stato di una singola istanza.
- Healthy threshold (Soglia di integrità): numero di controlli dello stato che devono essere superati prima che Elastic Load Balancing modifichi lo stato di integrità di un'istanza. Per Network Load Balancer, Unhealthy threshold (Soglia di mancata integrità) è un'impostazione di sola lettura che è sempre uguale al valore di soglia di integrità.
- Deregistration delay (Ritardo annullamento della registrazione): il periodo di tempo espresso in secondi necessario al completamento delle richieste prima che venga annullata la registrazione di un'istanza.



> **Note**
>
> Il controllo dello stato di Elastic Load Balancing non influisce sul comportamento del controllo dello stato del gruppo Auto Scaling di un ambiente. Le istanze che non superano il controllo dello stato di Elastic Load Balancing non vengono sostituite automaticamente da Amazon EC2 Auto Scaling a meno che non configuri manualmente Amazon EC2 Auto Scaling per eseguire questa operazione. Per informazioni dettagliate, vedi Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk.

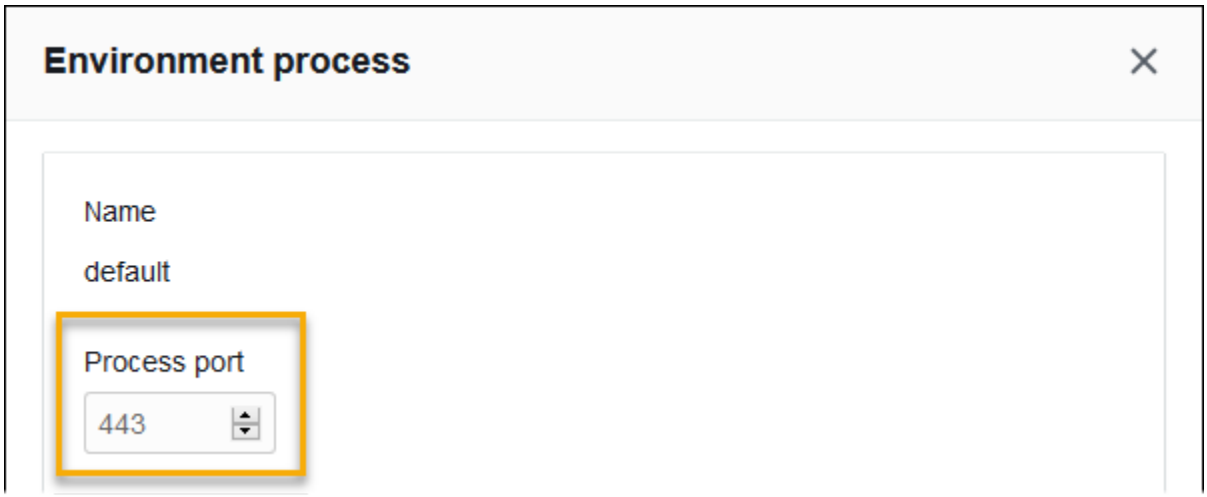
Per ulteriori informazioni sui controlli dello stato e su come influiscono sullo stato di integrità generale dell'ambiente, consulta Report dello stato di base.

Esempio: Network Load Balancer per un ambiente con crittografia end-to-end

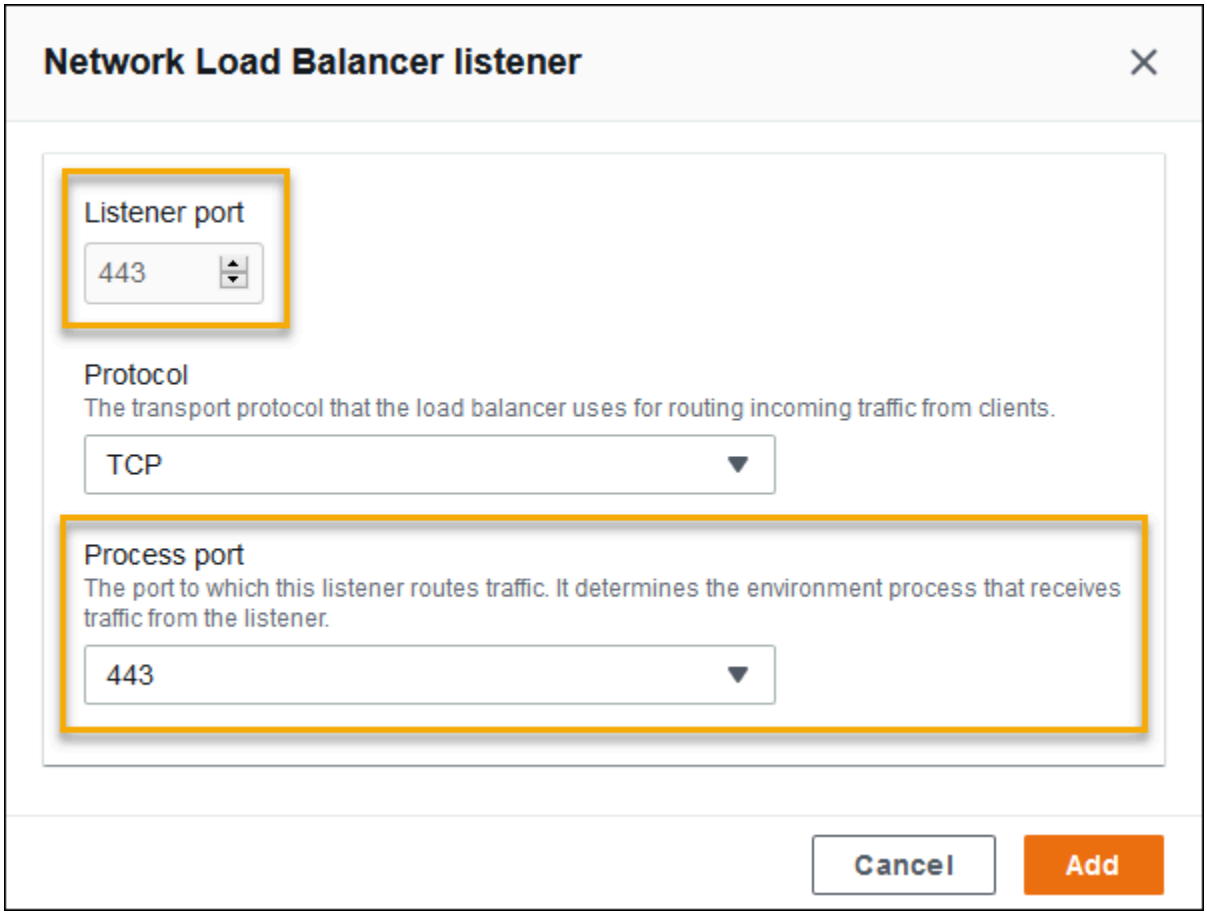
In questo esempio, l'applicazione richiede la crittografia del traffico end-to-end. Per configurare il sistema Network Load Balancer dell'ambiente affinché soddisfi questi requisiti puoi impostare il processo predefinito per ascoltare la porta 443, aggiungere un listener alla porta 443 che instradi il traffico verso il processo predefinito e disabilitare il listener predefinito.

Configurazione del sistema di bilanciamento del carico per questo esempio

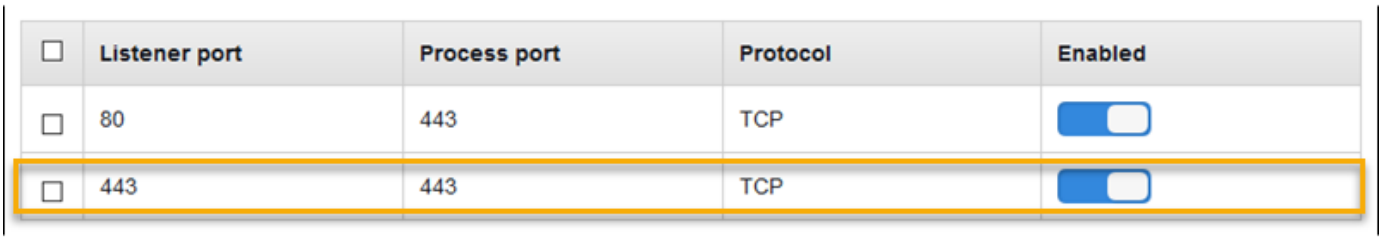
1. Configura il processo predefinito. Seleziona il processo predefinito, quindi in Actions (Operazioni), scegli Edit (Modifica). In Process port (Porta processo), digita `443`.



2. Aggiungi un listener alla porta 443. Aggiungi un nuovo listener. In Porta listener, digitare `443`. In Process port (Porta processo), assicurati che `443` sia selezionato.

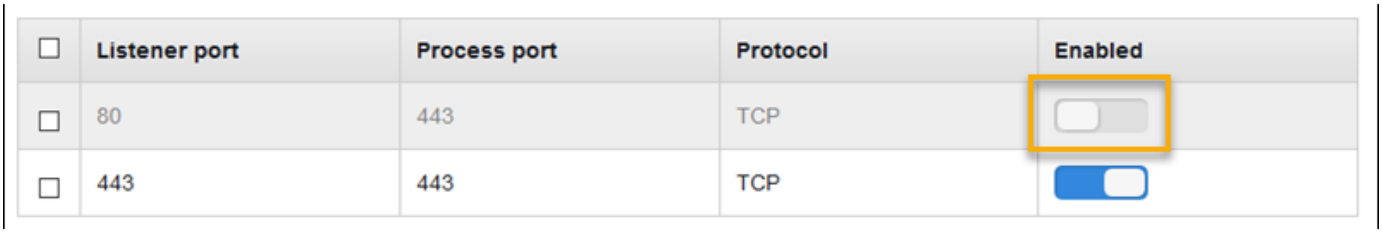


Ora puoi visualizzare il nuovo listener nell'elenco.



Listener port	Process port	Protocol	Enabled
80	443	TCP	
443	443	TCP	

3. Disattiva il listener predefinito della porta 80. Per il listener predefinito, disattiva l'opzione Enabled (Abilitato).



Listener port	Process port	Protocol	Enabled
80	443	TCP	
443	443	TCP	

Configurazione di un sistema Network Load Balancer utilizzando l'interfaccia a riga di comando EB

Quando esegui eb create, l'interfaccia a riga di comando EB ti chiede di scegliere un tipo di load balancer.

```
$ eb create
Enter Environment Name
(default is my-app): test-env
Enter DNS CNAME prefix
(default is my-app): test-env-DLW24ED23SF

Select a load balancer type
1) classic
2) application
3) network
(default is 1): 3
```

Puoi specificare un tipo di sistema di bilanciamento del carico anche con l'opzione `--elb-type`.

```
$ eb create test-env --elb-type network
```

Spazi dei nomi di Network Load Balancer

Le impostazioni relative ai sistemi Network Load Balancer sono disponibili nei seguenti spazi dei nomi:

- `aws:elasticbeanstalk:environment`: scegli il tipo di sistema di bilanciamento del carico per l'ambiente. Il valore di un sistema Network Load Balancer è `network`.
- `aws:elbv2:listener`: configura i listener nel sistema Network Load Balancer. Queste impostazioni vengono mappate alle impostazioni in `aws:elb:listener` per i sistemi Classic Load Balancer.
- `aws:elasticbeanstalk:environment:process`: configura i controlli dello stato e specifica porta e protocollo per i processi eseguiti sulle istanze del tuo ambiente. Le impostazioni relative a porta e protocollo sono mappate alle impostazioni relative a porta e protocollo dell'istanza in `aws:elb:listener` per un listener su un sistema Classic Load Balancer. Le impostazioni relative ai controlli dello stato sono mappate alle impostazioni nei namespace `aws:elb:healthcheck` e `aws:elasticbeanstalk:application`.

Example. ebextensions/network-load-balancer.config

Per iniziare a lavorare con un sistema Network Load Balancer, utilizza un file di configurazione per impostare il tipo di sistema bilanciamento del carico su `network`.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

> **Note**
>
> Il tipo di sistema di bilanciamento del carico può essere impostato solo durante la creazione dell'ambiente.

Example. ebextensions/nlb-processo-predefinito. config

Il seguente file di configurazione modifica le impostazioni di controllo dello stato del processo predefinito.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '10'
    HealthyThresholdCount: '5'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: TCP
```

Example. ebextensions/nlb-tls-listener.config

Il seguente file di configurazione aggiunge un listener TLS sulla porta 443 con un certificato SSL. Il load balancer gestisce la connessione sicura e inoltra il traffico al processo predefinito sulla porta 80.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
    Protocol: TLS
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
####################################
    SSLPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
```

Example. ebextensions/nlb-secure-listener.config

Il seguente file di configurazione utilizza un listener TCP per inoltrare il traffico crittografato alle istanze. Le istanze gestiscono direttamente la connessione sicura.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

L'opzione `DefaultProcess` è così denominata a causa degli Application Load Balancer, che possono avere listener non predefiniti sulla stessa porta per il traffico verso percorsi specifici (vedere Application Load Balancer per i dettagli). Per un sistema Network Load Balancer, l'opzione specifica l'unico processo di destinazione per questo listener.

In questo esempio, abbiamo denominato il processo `https` perché ascolta per proteggere il traffico (HTTPS). Il listener invia il traffico al processo sulla porta designata utilizzando il protocollo TCP. Questo comportamento è corretto, perché il traffico di rete HTTP e HTTPS viene implementato su TCP.

Configurazione dei sistemi di bilanciamento del carico Elastic Beanstalk dual-stack

Puoi consentire agli ambienti Elastic Beanstalk di IPv4 servire IPv6 entrambi i protocolli con bilanciatori di carico configurati dual-stack. Quando crei un ambiente con bilanciamento del carico, l'infrastruttura utilizza come impostazione predefinita. IPv4 È possibile scegliere di creare nuovi ambienti con configurazione dual-stack o aggiornare gli ambienti solo esistenti IPv4 in dual-stack.

Per consentire ai sistemi di bilanciamento del carico del tuo ambiente di servire sia il traffico di rete che il traffico di IPv4 rete, imposta l'opzione nel IPv6 namespace su dualstack. *`IpAddressType`* aws:elbv2:loadbalancer

> **Note**
>
> Elastic Beanstalk supporta solo protocolli dual-stack per Application Load Balancer e Network Load Balancer. Non supporta il dual-stack per ambienti che utilizzano Classic Load Balancer o ambienti a istanza singola.

Prerequisiti di Amazon VPC

Prima di configurare il sistema di bilanciamento del carico nell'ambiente Elastic Beanstalk, devi prima completare alcuni passaggi di configurazione con Amazon VPC:

1. Associa un blocco IPv6 CIDR al VPC del tuo ambiente.
2. Associa i blocchi IPv6 CIDR a tutte le sottoreti del VPC.
3. (Facoltativo) Se il tuo ambiente scambia traffico di rete con componenti esterni al VPC, dispone di tabelle di routing che specificano con quali reti il VPC può comunicare. In questo caso è necessario aggiornare le tabelle di routing VPC per abilitare il traffico. IPv6

Puoi completare queste configurazioni prerequisite con la console o i comandi Amazon VPC. AWS CLI Gli argomenti seguenti ti guideranno e ti indirizzeranno alla Amazon VPC User Guide e al riferimento dei AWS CLI comandi per maggiori dettagli.

> **Note**
>
> Dopo aver completato la configurazione del VPC, attendi alcuni minuti che le modifiche si propaghino prima di configurare il dual-stack per il tuo sistema di bilanciamento del carico. Se riscontri errori di configurazione del VPC o della sottorete durante la configurazione dual-stack, attendi qualche minuto che la configurazione del VPC si propaghi e riprova la configurazione dual-stack.

Completa i prerequisiti VPC utilizzando la console

La Amazon VPC User Guide fornisce passaggi dettagliati per completare queste attività prerequisite.

1. Consulta la Fase 1: Associa un blocco IPv6 CIDR al tuo VPC e alle sottoreti nella Amazon VPC User Guide.

   Questo passaggio prevede due procedure da completare:

   - Associa un blocco IPv6 CIDR al tuo VPC.
   - Associa un blocco IPv6 CIDR alle tue sottoreti VPC.
2. (Facoltativo) Se il tuo ambiente scambia traffico di rete con componenti esterni al VPC, dispone di tabelle di routing che specificano con quali reti il VPC può comunicare. In questo caso è necessario aggiornare le tabelle di routing VPC per abilitare il traffico. IPv6 Per completare questa

configurazione, consulta la Fase 2: Aggiornamento delle tabelle di routing nella Amazon VPC User Guide.

Completa i prerequisiti VPC utilizzando AWS CLI

È possibile utilizzare il AWS CLI per completare e verificare le configurazioni dei prerequisiti.

1. Associa un blocco IPv6 CIDR al VPC del tuo ambiente.
   a. Usa il associate-vpc-cidr-blockcomando per associare un blocco CIDR al tuo VPC.
   b. Usa il comando describe-vpcs per verificare le configurazioni del tuo VPC.

   Example comandi

```
# Associate an Amazon-provided IPv6 CIDR block with your VPC
aws ec2 associate-vpc-cidr-block \
    --vpc-id vpc-12345678 \
    --region us-east-1  \
    --amazon-provided-ipv6-cidr-block

# Verify the IPv6 CIDR block association
aws ec2 describe-vpcs \
    --vpc-ids vpc-12345678 \
    --region us-east-1  \
    --query 'Vpcs[0].Ipv6CidrBlockAssociationSet'
```

2. Associa i blocchi IPv6 CIDR a tutte le sottoreti del VPC.
   a. Usa il associate-subnet-cidr-blockcomando per associare un blocco CIDR alla tua sottorete.
   b. Utilizzate il comando describe-subnets per verificare le configurazioni delle sottoreti.

   Example comandi

```
# List all subnets in your VPC
aws ec2 describe-subnets \
    --region us-east-1  \
    --filters "Name=vpc-id,Values=vpc-12345678" \
    --query 'Subnets[].{SubnetId:SubnetId,AvailabilityZone:AvailabilityZone}'
```

```
# Associate IPv6 CIDR block with each subnet
aws ec2 associate-subnet-cidr-block \
    --subnet-id subnet-12345678 \
    --region us-east-1  \
    --ipv6-cidr-block 2001:db8::/64

# Verify IPv6 CIDR block association for all subnets
aws ec2 describe-subnets \
    --region us-east-1  \
    --filters "Name=vpc-id,Values=vpc-12345678" \
    --query 'Subnets[].
{SubnetId:SubnetId,Ipv6CidrBlock:Ipv6CidrBlockAssociationSet[0].Ipv6CidrBlock}'
```

3. (Facoltativo) Se il tuo ambiente scambia traffico di rete con componenti esterni al VPC, dispone di tabelle di routing che specificano con quali reti il VPC può comunicare. In questo caso è necessario aggiornare le tabelle di routing VPC per abilitare il traffico. IPv6

   a. Usa il comando create-route per aggiungere una rotta in una tabella di rotte all'interno del VPC.

   b. Usa il describe-route-tablescomando per verificare le tabelle delle rotte.

   Example comandi

```
# Add IPv6 route to Internet Gateway for public subnets
aws ec2 create-route \
    --route-table-id rtb-12345678 \
    --destination-ipv6-cidr-block ::/0 \
    --gateway-id igw-12345678  \
    --region us-east-1

# Add IPv6 route to NAT Gateway for private subnets (if applicable)
aws ec2 create-route \
    --route-table-id rtb-87654321 \
    --destination-ipv6-cidr-block ::/0 \
    --nat-gateway-id nat-12345678 \
    --region us-east-1

# Verify routes
aws ec2 describe-route-tables \
    --route-table-ids rtb-12345678 \
    --region us-east-1 \
    --query 'RouteTables[0].Routes'
```

Configurazione del dual-stack per il sistema di bilanciamento del carico Elastic Beanstalk

Dopo aver configurato la configurazione dei prerequisiti del VPC per il tuo ambiente, puoi configurare il load balancer con l'opzione dual-stack, in modo che possa servire entrambi i protocolli. IPv4 IPv6 Puoi utilizzare la console Elastic Beanstalk AWS CLI, `.ebextensions` i file di configurazione AWS e l'SDK per configurare il load balancer per servire il traffico dual-stack.

Utilizzo della console

Puoi utilizzare la console Elastic Beanstalk per configurare il dual-stack per il bilanciamento del carico del tuo ambiente.

> **Note**
>
> Questa configurazione dipende dai tempi di propagazione dei dati in diversi punti. Considerate i seguenti requisiti di tempistica quando configurate il sistema di bilanciamento del carico e testate le modifiche.
> Tempo di configurazione del VPC. Dopo aver completato la configurazione del VPC descritta inPrerequisiti di Amazon VPC, attendi alcuni minuti che le modifiche si propaghino prima di seguire queste procedure. Se riscontri errori di configurazione del VPC o della sottorete durante la configurazione dual-stack, attendi qualche minuto e riprova i passaggi di configurazione.
> Tempo di propagazione DNS. Dopo aver impostato l'`IpAddressType`opzione su dualstack, attendi alcuni minuti che le modifiche si propaghino prima di eseguire il test. Il completamento della propagazione DNS di Route 53 può richiedere 1-2 minuti. Durante questo periodo, potresti riscontrare problemi durante il test della comunicazione da un client all'applicazione se stai avviando richieste utilizzando il protocollo. IPv6

Creazione di un nuovo ambiente: per configurare il sistema di bilanciamento del carico per il supporto dual-stack

1. Avvia la console Elastic Beanstalk e inizia la procedura per creare un nuovo ambiente. Dopo aver impostato i campi obbligatori Service role e EC2 instance profile nella pagina Configura l'accesso al servizio, continua con i passaggi di questa procedura per impostare il load balancer sulla configurazione dual-stack. Per ulteriori informazioni su come iniziare, consulta. Creazione di un ambiente Elastic Beanstalk

2. In Configura l'accesso al servizio seleziona Avanti.
3. Viene visualizzata la pagina Configura rete, database e tag.

   Se hai completato quanto Prerequisiti di Amazon VPC descritto nella sezione precedente, hai già configurato il VPC e le sottoreti richiesti. In questo caso, salta questo passaggio insieme ai relativi passaggi secondari per passare alla selezione del VPC.

   a. Per configurare il VPC e le sottoreti, puoi selezionare Crea VPC per accedere alla console VPC. Segui la procedura riportata in Completa i prerequisiti VPC utilizzando la console.
   b. Attendi alcuni minuti per la propagazione degli aggiornamenti del VPC, quindi torna alla console Elastic Beanstalk e seleziona Aggiorna per continuare con il passaggio successivo.

      Se riscontri errori di configurazione del VPC o della sottorete nei passaggi rimanenti, attendi qualche minuto per consentire alla configurazione del VPC di propagarsi e riprova a eseguire i passaggi.
4. Nella pagina Configura rete, database e tag, seleziona un valore dal menu a discesa VPC a cui è associato IPv6 un blocco CIDR.

   Dopo aver selezionato un VPC, le sottoreti dell'istanza verranno popolate con le sottoreti VPC.
5. Seleziona una o più sottoreti di istanze, quindi seleziona Avanti.
6. Viene visualizzata la pagina Configura il traffico e il ridimensionamento delle istanze.

   Nelle impostazioni di rete del Load Balancer, seleziona Enable for Dualstack (&). IPv4 IPv6
7. Seleziona il tipo di Load balancer. Sia l'Application Load Balancer che il Network Load Balancer supportano il dualstack.
8. È possibile continuare a configurare altre opzioni di bilanciamento del carico nella pagina corrente della console. Per ulteriori informazioni sulle opzioni e sulla configurazione del load balancer, consulta gli altri argomenti di questo capitolo.
9. Continua con i passaggi per completare la configurazione e la creazione del tuo ambiente. Per ulteriori informazioni, consulta Creazione di un ambiente Elastic Beanstalk.

Ambiente esistente: per configurare il sistema di bilanciamento del carico per il supporto dual-stack

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria Configurazione di rete e database, scegli Modifica.
5. Se hai completato quanto Prerequisiti di Amazon VPC descritto nella sezione precedente, hai già configurato il VPC e le sottoreti richiesti. In questo caso, salta questo passaggio insieme ai relativi passaggi secondari per passare alla selezione del VPC.

   a. Per configurare il VPC e le sottoreti, puoi selezionare Crea VPC per accedere alla console VPC. Segui la procedura riportata in Completa i prerequisiti VPC utilizzando la console.

   b. Attendi alcuni minuti per la propagazione degli aggiornamenti del VPC, quindi torna alla console Elastic Beanstalk e seleziona Aggiorna per continuare con il passaggio successivo.

      Se riscontri errori di configurazione del VPC o della sottorete nei passaggi rimanenti, attendi qualche minuto per consentire alla configurazione del VPC di propagarsi e riprova a eseguire i passaggi.
6. Nella pagina Rete e database, seleziona un valore dal menu a discesa VPC a cui è associato IPv6 un blocco CIDR.

   Dopo aver selezionato un VPC, le sottoreti dell'istanza verranno popolate con le sottoreti VPC.
7. Seleziona una o più sottoreti di istanze.
8. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.
9. Nella categoria di configurazione Instance traffic and scaling (Traffico e dimensionamento delle istanze), scegli Edit (Modifica).
10. Nelle impostazioni di rete del Load Balancer, seleziona Abilita per Dualstack (&). IPv4 IPv6
11. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Usando il AWS CLI

È possibile utilizzare l'interfaccia a riga di AWS comando (AWS CLI) per configurare i sistemi di bilanciamento del carico dell'ambiente in modo che servano sia IPv6 il traffico di rete che quello di IPv4 rete. Questa sezione fornisce esempi dei comandi create-environment e update-environment con lo spazio dei nomi. aws:elbv2:loadbalancer

> **Note**
>
> Questa configurazione dipende dalla tempistica di propagazione dei dati in diversi punti. Considerate i seguenti requisiti di tempistica quando configurate il sistema di bilanciamento del carico e testate le modifiche.
>
> Tempo di configurazione del VPC. Dopo aver completato la configurazione del VPC descritta inPrerequisiti di Amazon VPC, attendi alcuni minuti che le modifiche si propaghino prima di eseguire questi comandi per configurare il sistema di bilanciamento del carico per il dual-stack. Se riscontri errori di configurazione del VPC o della sottorete durante la configurazione dual-stack, attendi qualche minuto e riprova a eseguire i comandi.
>
> Tempo di propagazione DNS. Dopo aver impostato l'`IpAddressType`opzione su dualstack, attendi alcuni minuti che le modifiche si propaghino prima di eseguire il test. Il completamento della propagazione DNS di Route 53 può richiedere 1-2 minuti. Durante questo periodo, potresti riscontrare problemi durante il test della comunicazione da un client all'applicazione se stai avviando richieste utilizzando il protocollo. IPv6

Example di create-environment con configurazione dualstack (opzioni dello spazio dei nomi in linea)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.0 running Python 3.8" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-elasticbeanstalk-ec2-role \
Namespace=aws:elbv2:loadbalancer,OptionName=IpAddressType,Value=dualstack
```

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga. Il comando di esempio seguente illustra il comando update-environment.

Example di update-environment con configurazione dualstack (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
```

```
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.0 running Python 3.8" \
--option-settings \ file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:elbv2:loadbalancer",
    "OptionName": "IpAddressType",
    "Value": "dualstack"
  }
]
```

L'esempio seguente aggiorna un ambiente esistente su cui impostare l'opzione. `IpAddressType` IPv4

> **Note**
>
> Questo `update-environment` comando di esempio è utile se è necessario ripristinare la configurazione dell'ambiente da dual-stack a. IPv4

Example di update-environment su cui impostare IpAddressType IPv4

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2 v3.4.0 running Python 3.8" \
--option-settings \
Namespace=aws:elbv2:loadbalancer,OptionName=IpAddressType,Value=ipv4
```

Utilizzo dei file di configurazione.ebextensions

Puoi utilizzare i file di configurazione di Elastic Beanstalk per consentire ai sistemi di bilanciamento del carico dell'ambiente di servire sia il traffico di rete che quello di rete. IPv6 IPv4 Imposta l'*`IpAddressType`*opzione nel namespace su dualstack. aws:elbv2:loadbalancer

> **Note**
>
> Questa configurazione dipende dalla tempistica di propagazione dei dati in diversi punti. Considerate i seguenti requisiti di tempistica quando configurate il sistema di bilanciamento del carico e testate le modifiche.
>
> Tempo di configurazione del VPC. Dopo aver completato la configurazione VPC descritta inPrerequisiti di Amazon VPC, attendi alcuni minuti affinché le modifiche al VPC si propaghino prima di applicare le configurazioni. `.ebextensions` Se riscontri errori di configurazione del VPC o della sottorete durante questa configurazione, attendi qualche minuto e riprova.
>
> Tempo di propagazione DNS. Dopo aver impostato l'`IpAddressType`opzione su dualstack, attendi alcuni minuti che le modifiche si propaghino prima di eseguire il test. Il completamento della propagazione DNS di Route 53 può richiedere 1-2 minuti. Durante questo periodo, potresti riscontrare problemi durante il test della comunicazione da un client all'applicazione se stai avviando richieste utilizzando il protocollo. IPv6

Example.ebextensions/options.config per la configurazione dualstack del load balancer (sintassi abbreviata)

```
option_settings:
  aws:elbv2:loadbalancer:
    IpAddressType: dualstack
```

Example.ebextensions/options.config per la configurazione dualstack del load balancer (sintassi standard)

```
option_settings:
  - namespace: aws:elbv2:loadbalancer
    option_name: IpAddressType
    value: dualstack
```

Utilizzo dell' AWS SDK

È possibile configurare il dual-stack utilizzando. AWS SDKs Analogamente ai `create-environment` AWS CLI comandi `update-environment` and menzionati nella sezione precedente, è possibile utilizzare le azioni CreateEnvironmente UpdateEnvironmentAPI. Utilizzate il parametro `OptionSettings` request per specificare le opzioni del aws:elbv2:loadbalancer namespace.

> **Note**
>
> Questa configurazione dipende dalla tempistica di propagazione dei dati in diversi punti. Considerate i seguenti requisiti di tempistica quando configurate il sistema di bilanciamento del carico e testate le modifiche.
>
> Tempo di configurazione del VPC. Dopo aver completato la configurazione del VPC descritta inPrerequisiti di Amazon VPC, attendi alcuni minuti affinché le modifiche si propaghino prima di eseguire i programmi per configurare il load balancer per il dual-stack. Se riscontri errori di configurazione del VPC o della sottorete durante la configurazione dual-stack, attendi qualche minuto e prova a eseguire nuovamente i programmi per la configurazione dual-stack.
>
> Tempo di propagazione DNS. Dopo aver impostato l'`IpAddressType`opzione su dualstack, attendi alcuni minuti che le modifiche si propaghino prima di eseguire il test. Il completamento della propagazione DNS di Route 53 può richiedere 1-2 minuti. Durante questo periodo, potresti riscontrare problemi durante il test della comunicazione da un client all'applicazione se stai avviando richieste utilizzando il protocollo. IPv6

Risoluzione dei problemi

> **Prova Amazon Q Developer CLI per la risoluzione dei problemi assistita dall'intelligenza artificiale**
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Risoluzione dei problemi degli ambienti Elastic Beanstalk con Amazon Q Developer CLI nei blog. AWS

Questa sezione fornisce indicazioni per la risoluzione dei problemi relativi alla configurazione dual-stack del sistema di bilanciamento del carico.

Evento: in VPC *`vpc_id`* non sono configurati blocchi IPv6 CIDR. IPv6 I blocchi CIDR sono necessari per il bilanciamento del carico dualstack. Associa un blocco IPv6 CIDR al tuo VPC prima di utilizzare la modalità dualstack.

Al tuo VPC e a tutte le sottoreti devono essere associati dei blocchi IPv6 CIDR. Questo è uno dei prerequisiti del VPC da completare prima di configurare il sistema di bilanciamento del carico per il

supporto dualstack. Per ulteriori informazioni sul completamento di questa attività, vedere la parte precedente di questo argomento. Prerequisiti di Amazon VPC

Evento: una o più sottoreti per VPC *vpc_id* non hanno IPv6 blocchi CIDR configurati. IPv6 I blocchi CIDR sono necessari per le sottoreti utilizzate con il sistema di bilanciamento del carico dualstack. Associa i blocchi IPv6 CIDR a tutte le sottoreti richieste prima di utilizzare la modalità dualstack.

Tutte le sottoreti del tuo VPC devono avere blocchi IPv6 CIDR associati. Questo è uno dei prerequisiti del VPC da completare prima di configurare il sistema di bilanciamento del carico per il supporto dualstack. Per ulteriori informazioni sul completamento di questa attività, vedere la parte precedente di questo argomento. Prerequisiti di Amazon VPC

Errore: l'**`IpAddressType`**opzione può essere applicata solo agli ambienti Elastic Beanstalk configurati con Application Load Balancer o Network Load Balancer.

Questo messaggio indica che l'ambiente Elastic Beanstalk potrebbe essere un ambiente a istanza singola o che potrebbe utilizzare un Classic Load Balancer. È possibile configurare solo gli ambienti configurati con Application Load Balancer o Network Load Balancer. `IpAddressType`

Configurazione dei log di accesso

Puoi utilizzare i file di configurazione per configurare il load balancer dell'ambiente per caricare i log degli accessi in un bucket Amazon S3. Per istruzioni, consulta i seguenti file di configurazione di GitHub esempio su:

- `loadbalancer-accesslogs-existingbucket.config`: configura il load balancer per caricare i log degli accessi in un bucket Amazon S3 esistente.
- `loadbalancer-accesslogs-newbucket.config`: configura il load balancer per caricare i log degli accessi in un nuovo bucket.

Aggiunta di un database all'ambiente Elastic Beanstalk

Elastic Beanstalk fornisce l'integrazione accoppiata del database con Amazon Relational Database Service (Amazon RDS). È possibile utilizzare Elastic Beanstalk per aggiungere un database MySQL, PostgreSQL, Oracle o SQL Server ad un ambiente esistente o a uno nuovo dopo averlo creato. Quando aggiungi un'istanza di database accoppiata a un ambiente, Elastic Beanstalk fornisce le informazioni di connessione all'applicazione. In questo modo è possibile impostare le proprietà dell'ambiente per il nome host del database, la porta, il nome utente, la password e il nome database.

Vantaggi dell'utilizzo di un database accoppiato

Se è la prima volta che utilizzi un'istanza database con la tua applicazione, consigliamo di aggiungere un database a un ambiente di test con la console Elastic Beanstalk tramite il processo descritto in questo argomento. In questo modo è possibile verificare se la propria applicazione può leggere le proprietà dell'ambiente, costruire una stringa di connessione e connettersi a un'istanza database, senza la configurazione aggiuntiva richiesta per un database esterno a Elastic Beanstalk.

Considerazioni relative alla fase di produzione

Dopo aver verificato il corretto funzionamento dell'applicazione con il database, è possibile passare a un ambiente di produzione. A questo punto è possibile disaccoppiare il database dall'ambiente Elastic Beanstalk per passare a una configurazione che offre maggiore flessibilità. Il database disaccoppiato può rimanere operativo come istanza di database Amazon RDS esterna. L'integrità dell'ambiente non è influenzata dal disaccoppiamento del database. Se necessario, è possibile terminare l'ambiente e scegliere l'opzione di mantenere il database disponibile e operativo al di fuori di Elastic Beanstalk.

Vantaggi del passaggio a un database disaccoppiato

L'utilizzo di un database esterno presenta diversi vantaggi. È possibile connettersi al database esterno da più ambienti, utilizzare tipi di database non supportati dai database integrati ed eseguire blue/green distribuzioni. In alternativa all'utilizzo di un database disaccoppiato creato da Elastic Beanstalk, è anche possibile creare un'istanza database al di fuori dell'ambiente Elastic Beanstalk. Entrambe le opzioni comportano un'istanza database esterna al tuo ambiente Elastic Beanstalk e richiede l'aggiunta di un gruppo di sicurezza e la configurazione della stringa di connessione. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon RDS.

Sections

- Ciclo di vita del database
- Aggiunta di un'istanza database Amazon RDS all'ambiente tramite la console
- Connessione al database
- Configurazione di un'istanza database RDS integrata tramite la console
- Configurazione di un'istanza database RDS integrata tramite i file di configurazione
- Disaccoppiamento di un'istanza database RDS tramite la console
- Disaccoppiamento di un'istanza database RDS tramite i file di configurazione

Ciclo di vita del database

È possibile scegliere cosa accade al database dopo averlo disaccoppiato dall'ambiente Elastic Beanstalk. Le opzioni tra cui è possibile scegliere vengono definite collettivamente come policy di eliminazione. Le seguenti policy di eliminazione si applicano a un database dopo che l'utente lo disaccoppia da un ambiente Elastic Beanstalk o termina l'ambiente Elastic Beanstalk.

- Snapshot (Snapshot): prima che Elastic Beanstalk termini il database, ne salva un'istantanea. È possibile ripristinare un database da uno snapshot quando si aggiunge un'istanza database a un ambiente Elastic Beanstalk o quando viene creato un database autonomo. Per ulteriori informazioni sulla creazione di una nuova istanza database autonoma da uno snapshot, consultare Ripristino da uno snapshot DB nella Guida per l'utente di Amazon RDS. Potrebbero essere applicati costi per l'archiviazione degli snapshot di database. Per ulteriori informazioni, consulta la sezione relativa allo storage di backup in Prezzi di Amazon RDS.
- Delete (Elimina): Elastic Beanstalk termina il database. Una volta terminata, l'istanza del database non è più disponibile per nessuna operazione.
- Retain (Mantieni): l'istanza del database non viene terminata. Rimane disponibile e operativo, anche se disaccoppiato da Elastic Beanstalk. È quindi possibile configurare uno o più ambienti per connettersi al database come istanza di database Amazon RDS esterna. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon RDS.

Aggiunta di un'istanza database Amazon RDS all'ambiente tramite la console

Puoi aggiungere un'istanza database all'ambiente tramite la console Elastic Beanstalk.

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Puoi configurare le seguenti opzioni:

- Snapshot: scegli uno snapshot di database esistente. Elastic Beanstalk ripristina lo snapshot e lo aggiunge al tuo ambiente. Il valore predefinito è None (Nessuna). Quando il valore è None (Nessuno), è possibile configurare un nuovo database tramite le altre impostazioni della pagina.
- Engine (Motore): scegli un motore di database.
- Engine version (Versione motore): scegli una versione specifica del motore di database.
- Instance class (Classe istanza): scegli una classe di istanza database. Per informazioni sulle classi di istanza database, consulta https://aws.amazon.com/rds/.
- Storage: scegli la quantità di storage da rendere disponibile per il database. In seguito puoi aumentare ma non diminuire lo storage allocato. Per ulteriori informazioni sull'allocazione di storage, consulta Caratteristiche.
- Username (Nome utente): inserire un nome utente a propria scelta utilizzando una combinazione di numeri e lettere.
- Password: immetti una password di tua scelta contenente da 8 a 16 caratteri ASCII stampabili (esclusi `/`, `\` e `@`).
- Availability (Disponibilità): scegli High (Multi-AZ) (Alta (AZ multiple)) per eseguire un backup a caldo in una seconda zona di disponibilità per una disponibilità elevata.
- Database deletion policy (Policy di eliminazione del database): la policy di eliminazione determina cosa accade al database dopo che è stato disaccoppiato dal proprio ambiente. Può essere impostato su uno dei seguenti valori: `Create Snapshot`, `Retain`, or `Delete`. Questi valori sono descritti in Ciclo di vita del database in questo stesso argomento.

> **Note**
>
> Elastic Beanstalk crea un utente master per il database utilizzando il nome utente e la password forniti. Per ulteriori informazioni sull'utente master e i relativi privilegi, consulta Privilegi dell'account utente master.

L'aggiunta di un'istanza database richiede circa 10 minuti. Al termine dell'aggiornamento, il nuovo database è accoppiato al proprio ambiente. Il nome host e altre informazioni di connessione dell'istanza database sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente.

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Connessione al database

Utilizza le informazioni sulla connettività per la connessione al database dall'interno dell'applicazione tramite variabili di ambiente. Per ulteriori informazioni sull'utilizzo di Amazon RDS con le tue applicazioni, vedi i seguenti argomenti.

- Java SE: Connessione a un database (piattaforme Java SE)
- Java con Tomcat: Connessione a un database (piattaforme Tomcat)
- Node.js: Connessione a un database
- .NET: Connessione a un database

- PHP: Connessione a un database con PDO o My SQLi
- Python: Connessione a un database
- Ruby: Connessione a un database

Configurazione di un'istanza database RDS integrata tramite la console

È possibile visualizzare e modificare le impostazioni di configurazione per l'istanza database nella sezione Database della pagina Configuration (Configurazione) dell'ambiente nella console Elastic Beanstalk.

Per configurare l'istanza database dell'ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).

È possibile modificare le impostazioni Instance class (Classe istanza), Storage (Archiviazione), Password, Availability (Disponibilità) e Database deletion policy (Policy di eliminazione del database) dopo la creazione del database. In caso di modifica della classe dell'istanza, Elastic Beanstalk ripete il provisioning dell'istanza database.

Se Elastic Beanstalk non è più necessario per associare il database all'ambiente, è possibile scegliere di disaccoppiarlo selezionando Decouple database (Disaccoppiamento del database). È importante comprendere le opzioni e le considerazioni coinvolte in questa operazione. Per ulteriori informazioni, consulta the section called “Disaccoppiamento di un'istanza database RDS tramite la console”.

> **Attenzione**
>
> Non modificare le impostazioni nell'istanza database accoppiata al di fuori della funzionalità fornita da Elastic Beanstalk (ad esempio, nella console Amazon RDS). In caso contrario, la configurazione del database Amazon RDS potrebbe non essere sincronizzata con la definizione dell'ambiente. Quando aggiorni o riavvii l'ambiente, le impostazioni specificate nell'ambiente sovrascrivono le impostazioni definite al di fuori di Elastic Beanstalk.

Se devi modificare impostazioni non supportate direttamente da Elastic Beanstalk, usa i file di configurazione di Elastic Beanstalk.

Configurazione di un'istanza database RDS integrata tramite i file di configurazione

È possibile configurare l'istanza database dell'ambiente usando i file di configurazione. Usa le opzioni nel namespace `aws:rds:dbinstance`. L'esempio seguente modifica le dimensioni dello storage di database allocate impostandole su 100 GB.

Example db-instance-options.ebextensions/ .config

```
option_settings:
  aws:rds:dbinstance:
    DBAllocatedStorage: 100
```

Se vuoi configurare proprietà dell'istanza database non supportate da Elastic Beanstalk, puoi comunque usare un file di configurazione e specificare le impostazioni usando la chiave `resources`. L'esempio seguente imposta i valori sulle proprietà Amazon RDS `Iops` e `StorageType`.

Example.ebextensions/ db-instance-properties .config

```
Resources:
  AWSEBRDSDatabase:
    Type: AWS::RDS::DBInstance
    Properties:
      StorageType:io1
      Iops: 1000
```

Disaccoppiamento di un'istanza database RDS tramite la console

È possibile disaccoppiare il database da un ambiente Elastic Beanstalk senza compromettere l'integrità dell'ambiente. Prima di disaccoppiare il database, prendere in considerazione i requisiti seguenti:

- Cosa dovrebbe accadere al database dopo che è stato disaccoppiato?

  È possibile scegliere di creare uno snapshot del database e quindi terminarlo, mantenere il database operativo come database autonomo esterno a Elastic Beanstalk o eliminare

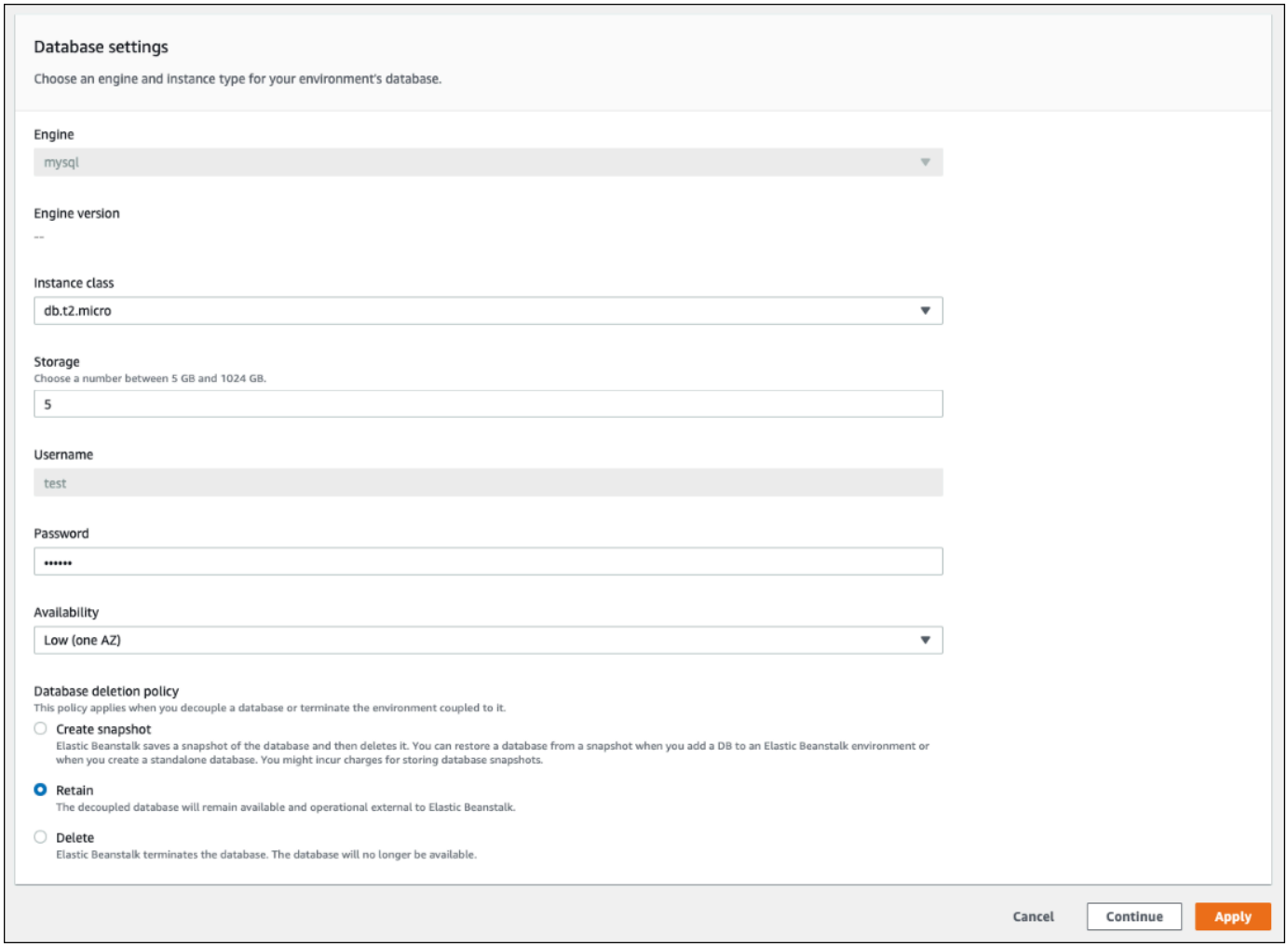
definitivamente il database. L'impostazione Database deletion policy (Policy di eliminazione del database) determina questo risultato. Per una descrizione dettagliata delle policy di eliminazione, consultare Ciclo di vita del database in questo stesso argomento.

- È necessario apportare modifiche alle impostazioni di configurazione del database prima di disaccoppiarlo?

  Se è necessario apportare modifiche alla configurazione del database, queste devono essere applicate prima del disaccoppiamento del database. Ciò include le modifiche apportate alla Database deletion policy (Policy di eliminazione del database). Eventuali modifiche in sospeso che vengono inviate contemporaneamente all'impostazione Decouple database (Disaccoppiamento del database) vengono ignorate, mentre viene applicata soltanto l'impostazione di disaccoppiamento.

Per disaccoppiare un'istanza database da un ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Esaminare tutti i valori delle configurazioni nella sezione Database settings (Impostazioni database), in particolare la Database deletion policy (Policy di eliminazione del database), che determina cosa succede al database dopo che è stato disaccoppiato.



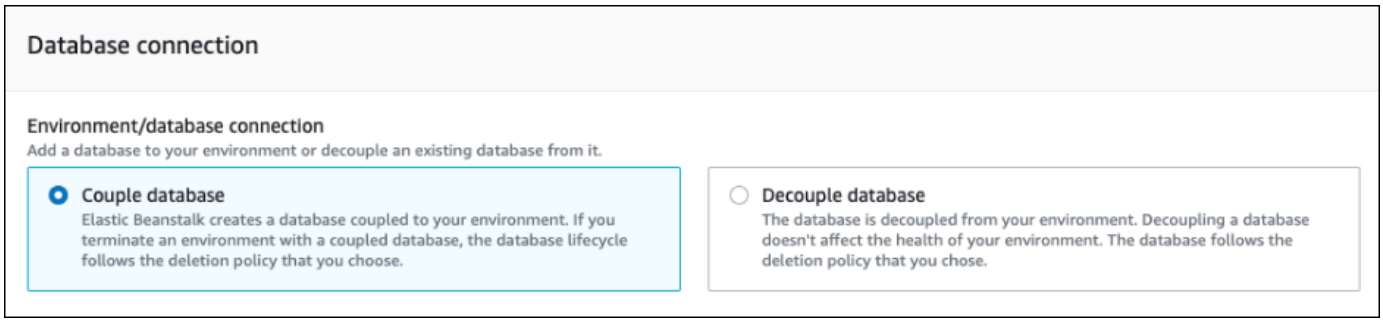
Se tutte le altre impostazioni di configurazione sono corrette, passare alla Fase 6 per disaccoppiare il database.

> **Warning**
>
> È importante applicare l'impostazione Database deletion policy (Policy di eliminazione del database) separatamente da Decouple database (Disaccoppiamento del database). Se si seleziona Apply (Applica) con l'intento di salvare sia Decouple database (Disaccoppiamento del database) sia una nuova Database deletion policy (Policy di eliminazione del database), la nuova policy di eliminazione selezionata viene ignorata. Elastic Beanstalk disaccoppia il database seguendo la policy di eliminazione precedente. Se la policy di eliminazione precedente è `Delete` o `Create Snapshot`, si rischia di perdere il database invece di seguire la policy in sospeso prevista.

Se una delle impostazioni di configurazione richiede aggiornamenti, procedere come segue:

1. Apportare le modifiche richieste nel pannello Database settings (Impostazioni database).
2. Scegli Applica. Per salvare le modifiche alla configurazione del database sono necessari alcuni minuti.
3. Tornare alla Fase 3 e scegliere Configuration (Configurazione) dal pannello di navigazione.

6. Accedere alla sezione del riquadro Database connection (Connessione database).



7. Scegliere Decouple database (Disaccoppia database).
8. Scegliere Apply (Applica) per avviare l'operazione di disaccoppiamento del database.

L'impostazione della policy di eliminazione determina il risultato del database e il periodo di tempo necessario per disaccoppiare il database.

- Se la policy di eliminazione è impostata su `Delete`, il database viene eliminato. L'operazione può richiedere circa 10-20 minuti, a seconda delle dimensioni del database.
- Se la policy di eliminazione è impostata su `Snapshot`, viene creata una snapshot del database. Quindi, il database viene eliminato. Il tempo necessario per questo processo varia a seconda delle dimensioni del database.
- Se la policy di eliminazione è impostata su `Retain`, il database rimane operativo all'esterno dell'ambiente Elastic Beanstalk. Di solito sono necessari meno di cinque minuti per disaccoppiare un database.

Se è stato deciso di mantenere il database esterno all'ambiente Elastic Beanstalk è necessario effettuare ulteriori passaggi per configurarlo. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon RDS. Se si prevede di utilizzare il database disaccoppiato per un ambiente di

produzione, verificare che il tipo di archiviazione utilizzato dal database sia adatto al carico di lavoro. Per ulteriori informazioni, consultare Archiviazione dell'istanza database e Modifica di un'istanza database nella Guida per l'utente di Amazon RDS.

Disaccoppiamento di un'istanza database RDS tramite i file di configurazione

È possibile disaccoppiare l'istanza database da un ambiente Elastic Beanstalk senza influire sull'integrità dell'ambiente. L'istanza database segue la policy di eliminazione del database che è stata applicata quando il database è stato disaccoppiato.

Entrambe le opzioni necessarie per disaccoppiare il database si trovano nello spazio dei nomi the section called “aws:rds:dbinstance”. Essi sono i seguenti:

- L'opzione `DBDeletionPolicy` imposta la policy di eliminazione. Questa può essere impostata su uno dei valori seguenti: `Snapshot`, `Delete` oppure `Retain`. Questi valori sono descritti in Ciclo di vita del database in questo stesso argomento.
- L'opzione `HasCoupledDatabase` determina se l'ambiente ha un database accoppiato.
  - Se è stato attivato `true`, Elastic Beanstalk crea una nuova istanza database accoppiata al proprio ambiente.
  - Se è stato attivato `false`, Elastic Beanstalk inizia a disaccoppiare l'istanza database dall'ambiente.

Se si desidera modificare la configurazione del database prima di disaccoppiarlo, applicare prima tutte le modifiche alla configurazione in un'operazione separata. Ciò include modificare la configurazione di `DBDeletionPolicy`. Dopo aver applicato le modifiche, eseguire un comando separato per impostare l'opzione di disaccoppiamento. Se si inviano contemporaneamente altre impostazioni di configurazione e l'impostazione di disaccoppiamento, le altre impostazioni delle opzioni di configurazione vengono ignorate mentre viene applicata l'impostazione di disaccoppiamento.

> **Warning**
>
> È importante eseguire i comandi per applicare le impostazioni `DBDeletionPolicy` e `HasCoupledDatabase` come due operazioni separate. Se la policy di eliminazione attiva è

già impostata su `Delete` o `Snapshot`, si rischia di perdere il database. Il database segue la policy di eliminazione attualmente attiva anziché la policy di eliminazione in sospeso prevista.

Per disaccoppiare un'istanza database da un ambiente

Seguire questi passaggi per disaccoppiare il database dall'ambiente Elastic Beanstalk. È possibile utilizzare l'EB CLI o il AWS CLI per completare i passaggi. Per ulteriori informazioni, consultare Personalizzazione avanzata dell'ambiente con i file di configurazione.

1. Se si desidera modificare la policy di eliminazione, impostare un file di configurazione nel formato seguente. In questo esempio la policy di eliminazione è impostata su conservazione.

   Example

   ```
   option_settings:
     aws:rds:dbinstance:
       DBDeletionPolicy: Retain
   ```

2. Eseguire il comando utilizzando lo strumento preferito per completare l'aggiornamento della configurazione.
3. Configurare un file di configurazione per impostare `HasCoupledDatabase` su `false`.

   Example

   ```
   option_settings:
     aws:rds:dbinstance:
       HasCoupledDatabase: false
   ```

4. Eseguire il comando utilizzando lo strumento preferito per completare l'aggiornamento della configurazione.

L'impostazione della policy di eliminazione determina il risultato del database e il periodo di tempo necessario per disaccoppiare il database.

- Se la policy di eliminazione è impostata su `Delete`, il database viene eliminato. L'operazione può richiedere circa 10-20 minuti, a seconda delle dimensioni del database.

- Se la policy di eliminazione è impostata su `Snapshot`, viene creata una snapshot del database. Quindi, il database viene eliminato. Il tempo necessario per questo processo varia a seconda delle dimensioni del database.
- Se la policy di eliminazione è impostata su `Retain`, il database rimane operativo all'esterno dell'ambiente Elastic Beanstalk. Di solito sono necessari meno di cinque minuti per disaccoppiare un database.

Se è stato deciso di mantenere il database esterno all'ambiente Elastic Beanstalk è necessario effettuare ulteriori passaggi per configurarlo. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon RDS. Se si prevede di utilizzare il database disaccoppiato per un ambiente di produzione, verificare che il tipo di archiviazione utilizzato dal database sia adatto al carico di lavoro. Per ulteriori informazioni, consultare Archiviazione dell'istanza database e Modifica di un'istanza database nella Guida per l'utente di Amazon RDS.

Tuo AWS Elastic Beanstalk sicurezza dell'ambiente

Elastic Beanstalk offre diverse opzioni che controllano l'accesso ai servizi (sicurezza) dell'ambiente e delle relative istanze Amazon EC2. In questo argomento viene descritta la configurazione di queste opzioni.

Sections

- Configurazione della sicurezza dell'ambiente
- Spazi dei nomi della configurazione di sicurezza degli ambienti

Configurazione della sicurezza dell'ambiente

Puoi modificare la configurazione di sicurezza dell'ambiente Elastic Beanstalk utilizzando la console Elastic Beanstalk.

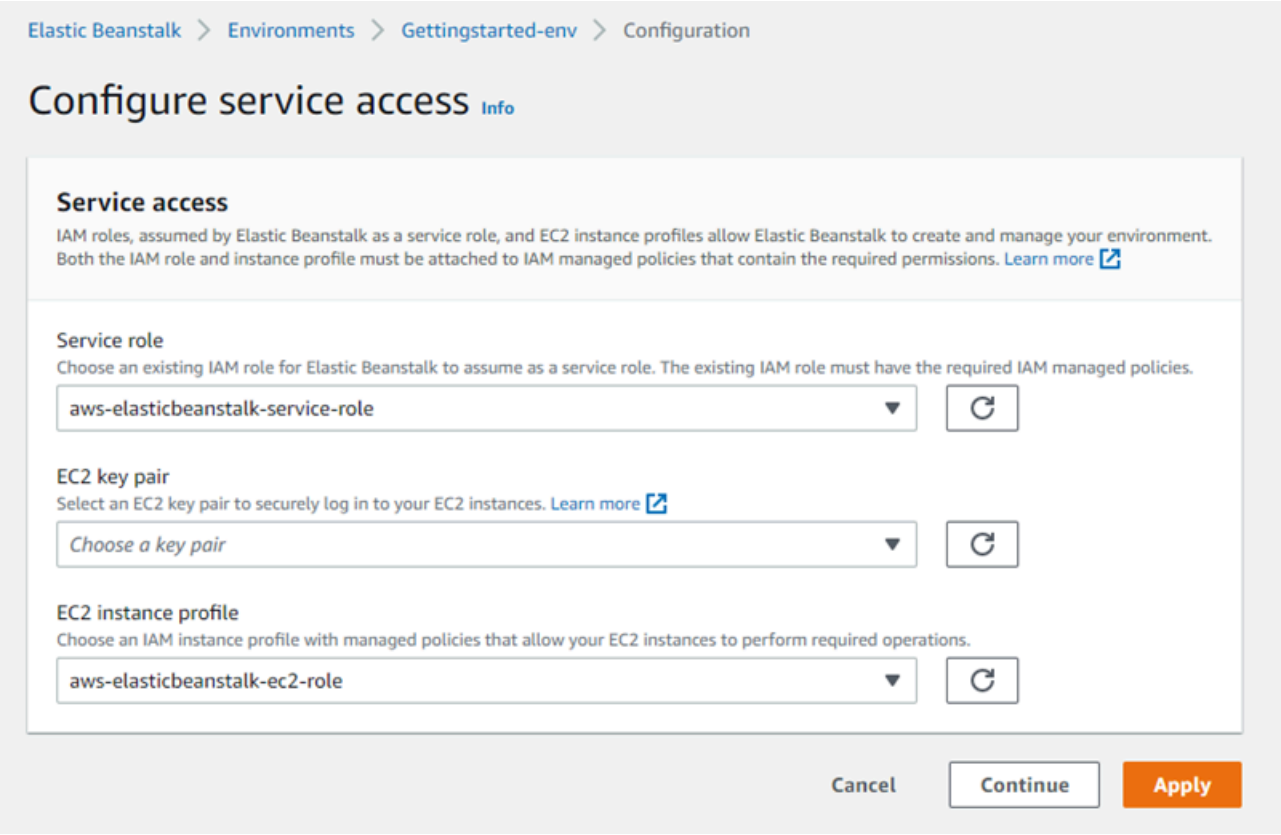
Configurazione dell'accesso ai servizi dell'ambiente (sicurezza) nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Service access (Accesso ai servizi), scegli Edit (Modifica).

Sono disponibili le impostazioni seguenti.

Settings

- Ruolo del servizio
- Coppia di chiavi EC2
- Profilo dell'istanza IAM



Ruolo del servizio

Seleziona un ruolo di servizio da associare all'ambiente Elastic Beanstalk. Elastic Beanstalk assume il ruolo di servizio quando accede ad altri servizi per conto dell'utente. AWS Per informazioni dettagliate, vedi Gestione dei ruoli del servizio Elastic Beanstalk.

Coppia di chiavi EC2

Puoi accedere in modo sicuro alle istanze Amazon Elastic Compute Cloud (Amazon EC2) fornite per l'applicazione Elastic Beanstalk con una coppia di chiavi Amazon EC2. Per istruzioni sulla creazione di una coppia di chiavi, consulta Creating a Key Pair Using Amazon EC2 nella Amazon EC2 User Guide.

> **Note**
>
> Quando crei una coppia di chiavi, Amazon EC2 memorizza una copia della chiave pubblica. Se non devi più usarla per connetterti alle istanze dell'ambiente, puoi eliminarla da Amazon EC2. Per i dettagli, consulta la sezione Eliminazione della coppia di chiavi nella Guida per l'utente di Amazon EC2.

Scegli una EC2 key pair (Coppia di chiavi EC2) dal menu a discesa per assegnarla alle istanze del tuo ambiente. Quando assegni una coppia di chiavi, la chiave pubblica viene archiviata sull'istanza per autenticare la chiave privata, che memorizzi in locale. La chiave privata non viene mai archiviata. AWS

Per ulteriori informazioni sulla connessione alle istanze Amazon EC2, consulta Connect to Your Instance e Connecting to Linux/UNIX Instances from Windows using PuTTY nella Amazon EC2 User Guide.

Profilo dell’istanza IAM

Un profilo dell'istanza EC2 è un ruolo IAM che viene applicato alle istanze avviate nell'ambiente Elastic Beanstalk. Le istanze Amazon EC2 assumono il ruolo di profilo dell'istanza per firmare le richieste AWS e accedere alle API, ad esempio per caricare i log su Amazon S3.

La prima volta che crei un ambiente nella console Elastic Beanstalk, Elastic Beanstalk chiede di creare un profilo dell'istanza con un set predefinito di autorizzazioni. Puoi aggiungere autorizzazioni a questo profilo per fornire alle tue istanze l'accesso ad altri servizi. AWS Per informazioni dettagliate, vedi Gestione dei profili dell'istanza Elastic Beanstalk.

> **Note**
>
> In precedenza Elastic Beanstalk creava un `aws-elasticbeanstalk-ec2-role` profilo di istanza EC2 predefinito denominato la prima volta AWS che un account creava un ambiente. Questo profilo dell'istanza includeva le policy gestite predefinite. Se il tuo account dispone già di questo profilo dell'istanza, rimarrà disponibile per poter essere assegnata ai tuoi ambienti. Tuttavia, le recenti linee guida AWS sulla sicurezza non consentono a un AWS servizio di creare automaticamente ruoli con policy di fiducia per altri AWS servizi, in questo caso EC2. A causa di queste linee guida sulla sicurezza, Elastic Beanstalk non crea più un profilo dell'istanza `aws-elasticbeanstalk-ec2-role` predefinito.

> **Note**
>
> Esiste un altro aspetto della sicurezza delle istanze EC2 che definisce le regole del firewall per le istanze EC2. Questo è controllato dai gruppi di sicurezza EC2. Per ulteriori informazioni, consulta Le EC2 istanze Amazon per il tuo ambiente Elastic Beanstalk.

Spazi dei nomi della configurazione di sicurezza degli ambienti

Elastic Beanstalk fornisce le opzioni di configurazione nei seguenti spazi dei nomi per consentirti di personalizzare la sicurezza del tuo ambiente:

- `aws:elasticbeanstalk:environment`: configura il ruolo di servizio dell'ambiente utilizzando l'opzione `ServiceRole`.
- `aws:autoscaling:launchconfiguration`— Configura le autorizzazioni per le istanze Amazon EC2 dell'ambiente utilizzando `EC2KeyName` le opzioni`IamInstanceProfile`,`DisableDefaultEC2SecurityGroup`, e. `SecurityGroups`

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Tagging delle risorse negli ambienti Elastic Beanstalk

Puoi applicare tag ai tuoi AWS Elastic Beanstalk ambienti. I tag sono coppie chiave-valore associate AWS alle risorse. Per informazioni sul tagging delle risorse, sui casi d'uso, sui vincoli delle chiavi e dei valori di tag e sui tipi di risorse supportati di Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Elastic Beanstalk applica i tag di ambiente alla risorsa di ambiente stessa e ad altre AWS risorse che Elastic Beanstalk crea per l'ambiente. È inoltre possibile usare i tag per gestire le autorizzazioni al livello della risorsa specifica all'interno di un ambiente. Per ulteriori informazioni, consulta Tagging Your Amazon EC2 Resources nella Amazon EC2 User Guide.

Per impostazione predefinita Elastic Beanstalk applica alcuni tag all'ambiente:

- `elasticbeanstalk:environment-name`: il nome dell'ambiente.
- `elasticbeanstalk:environment-id`: l'ID dell'ambiente.

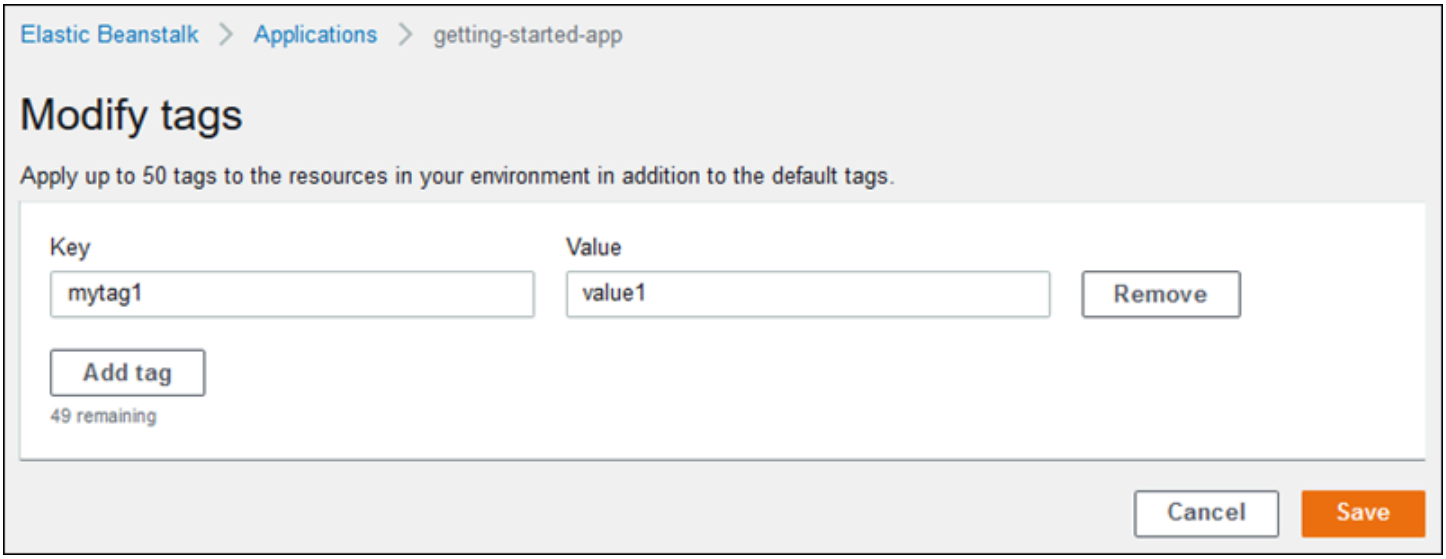
- `Name`— Inoltre il nome dell'ambiente. `Name`viene utilizzato nella EC2 dashboard di Amazon per identificare e ordinare le risorse.

Non è possibile modificare questi tag predefiniti.

Puoi specificare i tag quando crei l'ambiente Elastic Beanstalk. In un ambiente esistente, è possibile aggiungere o rimuovere i tag e aggiornare i valori dei tag esistenti. Un ambiente può avere fino a 50 tag, inclusi i tag predefiniti.

Aggiunta di tag durante la creazione dell'ambiente

Quando utilizzi la console Elastic Beanstalk per creare un ambiente, puoi specificare le chiavi e i valori dei tag nella pagina di configurazione Modify tags (Modifica tag) della procedura guidata Create New Environment (Crea nuovo ambiente).



Se utilizzi la CLI EB per creare un ambiente, utilizza l'opzione `--tags` con eb create per aggiungere i tag.

```
~/workspace/my-app$ eb create --tags mytag1=value1,mytag2=value2
```

Con il AWS CLI o altri client basati su API, usa il `--tags` parametro sul create-environment comando.

```
$ aws elasticbeanstalk create-environment \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
```

```
      --application-name my-app --environment-name my-env --cname-prefix my-app --
version-label v1 --template-name my-saved-config
```

Le configurazioni salvate includono i tag definiti dall'utente. Quando applichi una configurazione salvata che contiene i tag durante la creazione dell'ambiente, tali tag sono applicati al nuovo ambiente, a condizione che non specifichi nuovi tag. Se aggiungi i tag a un ambiente utilizzando uno dei metodi precedenti, tutti i tag definiti nella configurazione salvata vengono eliminati.

Gestione di tag di un ambiente esistente

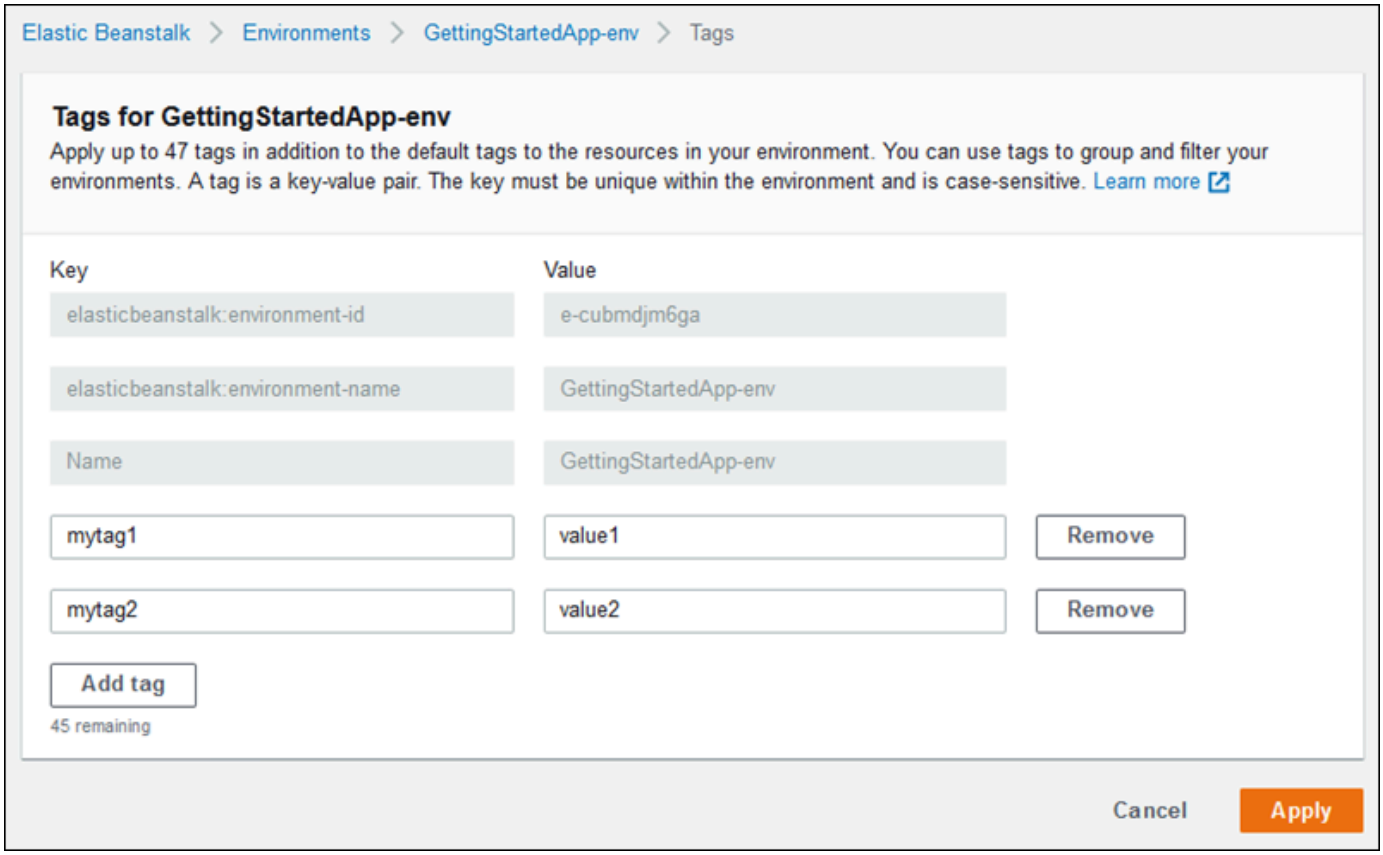
È possibile aggiungere, aggiornare ed eliminare i tag in un ambiente Elastic Beanstalk esistente. Elastic Beanstalk applica le modifiche alle risorse dell'ambiente.

Tuttavia, non è possibile modificare i tag predefiniti che Elastic Beanstalk applica all'ambiente.

Per gestire i tag di un ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione scegliere Tags (Tag).

   La pagina di gestione dei tag mostra l'elenco die tag esistenti nell'ambiente.



4. Aggiungi, aggiorna o elimina i tag:
   - Per aggiungere un tag, immetterlo nelle caselle vuote nella parte inferiore dell'elenco. Per aggiungere un altro tag, scegli Add tag (Aggiungi tag). Viene aggiunta un'altra coppia di caselle vuote da Elastic Beanstalk.
   - Per aggiornare la chiave o il valore di un tag, modificare la relativa casella nella riga del tag.
   - Per eliminare un tag, scegliere Remove (Rimuovi) accanto alla casella del valore del tag.
5. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Se utilizzi la CLI EB per aggiornare l'ambiente, utilizza eb tags per aggiungere, aggiornare, eliminare o elencare i tag.

Ad esempio, il comando seguente elenca i tag nell'ambiente predefinito.

```
~/workspace/my-app$ eb tags --list
```

Il comando seguente aggiorna il tag `mytag1` ed elimina il tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2
```

Per un elenco completo delle opzioni e per altri esempi, consulta `eb tags`.

Con il AWS CLI o altri client basati su API, usa il list-tags-for-resource comando per elencare i tag di un ambiente.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-app/my-env"
```

Utilizza il comando update-tags-for-resource per aggiungere, aggiornare o eliminare i tag in un ambiente.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-
app/my-env"
```

Specifica i tag da aggiungere e i tag da aggiornare nel parametro `--tags-to-add` di update-tags-for-resource. È stato aggiunto un tag non esistente e il valore di un tag esistente è stato aggiornato.

> **Note**
>
> Per utilizzare questi due AWS CLI comandi con un ambiente Elastic Beanstalk, è necessario l'ARN dell'ambiente. È possibile recuperare l'ARN utilizzando il seguente comando.
>
> ```
> $ aws elasticbeanstalk describe-environments
> ```

Variabili di ambiente e altre impostazioni software

La pagina di configurazione degli aggiornamenti, del monitoraggio e della registrazione consente di configurare il software sulle istanze Amazon Elastic Compute Cloud EC2 (Amazon) che eseguono l'applicazione. Puoi configurare variabili di ambiente, AWS X-Ray debug, archiviazione e streaming dei log delle istanze e impostazioni specifiche della piattaforma.

Argomenti

- Configurare le impostazioni specifiche della piattaforma
- Configurazione delle proprietà dell'ambiente (variabili di ambiente)
- Spazi dei nomi delle impostazioni software
- Accesso alle proprietà dell'ambiente
- Configurazione del debug AWS X-Ray
- Visualizzazione dei log dell'ambiente Elastic Beanstalk

Configurare le impostazioni specifiche della piattaforma

Oltre al set di opzioni standard disponibili per tutti gli ambienti, la maggior parte delle piattaforme Elastic Beanstalk consente di specificare le impostazioni specifiche per il linguaggio o il framework. Queste vengono visualizzati nella sezione Software della piattaforma della pagina Configurazione di aggiornamenti, monitoraggio e registrazione e possono assumere le seguenti forme.

- Proprietà dell'ambiente preimpostate: la piattaforma Ruby utilizza le proprietà dell'ambiente per le impostazioni del framework, ad esempio `RACK_ENV` e `BUNDLE_WITHOUT`.
- Proprietà dell'ambiente di tipo segnaposto: la piattaforma Tomcat definisce una proprietà dell'ambiente denominata `JDBC_CONNECTION_STRING` che non è impostata su nessun valore. Questo tipo di impostazione è più comune nelle versioni delle piattaforme meno recenti.
- Opzioni di configurazione: la maggior parte delle piattaforme definisce le opzioni di configurazione negli spazi dei nomi condivisi o specifici per la piattaforma, ad esempio `aws:elasticbeanstalk:xray` o `aws:elasticbeanstalk:container:python`.

Per configurare le impostazioni specifiche della piattaforma nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. In Software della piattaforma, apporta le modifiche necessarie alle impostazioni delle opzioni.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Per informazioni sulle opzioni specifiche della piattaforma e su come ottenere i valori di proprietà dell'ambiente nel codice, consulta l'argomento sulla piattaforma relativo al linguaggio o al framework in uso:

- Docker: the section called “Configurazione dell'ambiente”
- Go: Utilizzo della piattaforma Go di Elastic Beanstalk
- Java SE: Utilizzo della piattaforma Java SE di Elastic Beanstalk
- Tomcat: Utilizzo della piattaforma Tomcat di Elastic Beanstalk
- .NET Core su Linux: Utilizzo del core Elastic Beanstalk.NET su piattaforma Linux
- .NET: Utilizzo della piattaforma Windows Elastic Beanstalk .NET
- Node.js: Utilizzo della piattaforma Node.js di Elastic Beanstalk
- PHP: Utilizzo della piattaforma PHP di Elastic Beanstalk
- Python: Utilizzo della piattaforma Python di Elastic Beanstalk
- Ruby: Utilizzo della piattaforma Ruby di Elastic Beanstalk

Configurazione delle proprietà dell'ambiente (variabili di ambiente)

Puoi utilizzare le proprietà di ambiente, note anche come variabili di ambiente, per passare endpoint, impostazioni di debug e altre informazioni all'applicazione. Le variabili di ambiente consentono di eseguire l'applicazione in più ambienti per scopi diversi, come sviluppo, test, gestione temporanea e produzione.

Inoltre, quando aggiungi un database all'ambiente, Elastic Beanstalk imposta variabili di ambiente, `RDS_HOSTNAME` ad esempio quelle che puoi leggere nel codice dell'applicazione per costruire un oggetto o una stringa di connessione.

Per configurare le variabili di ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri verso il basso fino alle variabili di ambiente Runtime.

6. Seleziona Aggiungi variabile di ambiente.
7. Per Sorgente seleziona Testo normale.

> **Note**
>
> I valori Secrets Manager e SSM Parameter Store nel menu a discesa servono a configurare le variabili di ambiente come segreti per archiviare dati sensibili, come credenziali e chiavi API. Per ulteriori informazioni, consulta Utilizzo di Elastic Gestione dei segreti AWS AWS Systems Manager Beanstalk con e Parameter Store.

8. Immettere le coppie del nome della variabile di ambiente e del valore della variabile di ambiente.
9. Se è necessario aggiungere altre variabili, ripetere i passaggi da 6 a 8.
10. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Limitare le proprietà dell'ambiente

- Le chiavi possono contenere qualsiasi carattere alfanumerico e i simboli seguenti: `_ . : / + \ - @`

  I simboli elencati sono validi per le chiavi di proprietà dell'ambiente, ma potrebbero non esserlo per i nomi delle variabili di ambiente nella piattaforma dell'ambiente. Per assicurare la compatibilità con tutte le piattaforme, limita le proprietà dell'ambiente al modello seguente: `[A-Z_][A-Z0-9_]*`
- I valori possono contenere qualsiasi carattere alfanumerico, spazi vuoti e i simboli seguenti: `_ . : / = + \ - @ ' "`

> **Note**
>
> Occorre inserire caratteri di escape per alcuni caratteri inseriti nei valori delle proprietà di ambiente Usa il carattere barra rovesciata (\) per rappresentare alcuni caratteri speciali e caratteri di controllo. L'elenco seguente include esempi per la rappresentazione di alcuni caratteri che devono essere sottoposti a escape:
>
> - barra rovesciata (\) — per rappresentare l'uso `\\`
> - virgoletta singola (') — per rappresentare l'uso `\'`
> - virgolette doppie (") — per rappresentare l'uso `\"`

- Le chiavi e i valori fanno distinzione tra maiuscole e minuscole.

- La dimensione combinata di tutte le proprietà dell'ambiente non può superare i 4.096 byte se memorizzate come stringhe con il formato =. *key value*

Spazi dei nomi delle impostazioni software

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

Puoi utilizzare i file di configurazione di Elastic Beanstalk per impostare le proprietà e le opzioni di configurazione dell'ambiente nel codice sorgente. Utilizza lo spazio dei nomi`aws:elasticbeanstalk:application:environment` per definire le proprietà dell'ambiente.

Example.ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: www.example.com/api
```

Se si utilizzano file o CloudFormation modelli di configurazione per creare risorse personalizzate, è possibile utilizzare una CloudFormation funzione per ottenere informazioni sulla risorsa e assegnarle dinamicamente a una proprietà dell'ambiente durante la distribuzione. L'esempio seguente, tratto dal elastic-beanstalk-samples GitHub repository, utilizza la funzione Ref per ottenere l'ARN di un argomento Amazon SNS creato e lo assegna a una proprietà di ambiente denominata. `NOTIFICATION_TOPIC`

> **Note**
>
> - Se si utilizza una CloudFormation funzione per definire una proprietà di ambiente, la console Elastic Beanstalk visualizza il valore della proprietà prima che la funzione venga valutata. Puoi utilizzare lo script della piattaforma `get-config` per confermare i valori delle proprietà dell'ambiente disponibili per la tua applicazione.
> - La piattaforma Multicontainer Docker non viene utilizzata per creare risorse container. CloudFormation Di conseguenza, questa piattaforma non supporta la definizione delle proprietà dell'ambiente utilizzando le funzioni. CloudFormation

Example.Ebextensions/sns-topic.config

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

option_settings:
  aws:elasticbeanstalk:application:environment:
    NOTIFICATION_TOPIC: '`{"Ref" : "NotificationTopic"}`'
```

Puoi inoltre utilizzare questa caratteristica per propagare le informazioni dagli pseudoparametri di CloudFormation. Questo esempio ottiene la regione corrente e la assegna a una proprietà denominata `AWS_REGION`.

Example.Ebextensions/env-regionname.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    AWS_REGION: '`{"Ref" : "AWS::Region"}`'
```

La maggior parte delle piattaforme Elastic Beanstalk definisce spazi dei nomi aggiuntivi con opzioni per la configurazione del software in esecuzione sull'istanza, come il proxy inverso che inoltra le richieste all'applicazione. Per ulteriori informazioni sugli spazi dei nomi disponibili per la piattaforma in uso, consulta una delle sezioni seguenti:

- Go: Spazi dei nomi per la configurazione di Go
- Java SE: Spazio dei nomi per la configurazione di Java SE
- Tomcat: Spazio dei nomi della configurazione Tomcat
- .NET Core su Linux: NET Core nello spazio dei nomi di configurazione Linux
- .NET: Spazio dei nomi aws:elasticbeanstalk:container:dotnet:apppool
- Node.js: Spazi dei nomi per la configurazione di Node.js
- PHP: Namespace per la configurazione
- Python: Spazio dei nomi di configurazione di Python
- Ruby: Spazi dei nomi della configurazione Ruby

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la

console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Accesso alle proprietà dell'ambiente

Nella maggior parte dei casi, puoi accedere alle proprietà dell'ambiente nel codice dell'applicazione come accedi alle variabili di ambiente. In generale, tuttavia, le proprietà dell'ambiente vengono trasmesse soltanto all'applicazione e non possono essere visualizzate tramite la connessione di un'istanza nell'ambiente e l'esecuzione di `env`.

- Go: `os.Getenv`

```
endpoint := os.Getenv("API_ENDPOINT")
```

- Java SE: `System.getenv`

```
String endpoint = System.getenv("API_ENDPOINT");
```

- Tomcat — `System.getProperty` e `System.getenv`

  Le versioni della piattaforma Tomcat rilasciate a partire dal 26 marzo 2025 possono essere utilizzate anche `System.getenv` per accedere a variabili di ambiente in testo non crittografato. È possibile continuare a utilizzare per accedere alle variabili di ambiente in testo `System.getProperty` non crittografato. Tuttavia, le variabili di ambiente memorizzate come segrete sono disponibili solo utilizzando. `System.getenv`

```
String endpoint = System.getProperty("API_ENDPOINT");
```

```
String endpoint = System.getenv("API_ENDPOINT");
```

> **Important**
>
> L'aggiunta dell'`System.getenv`accesso per le variabili di ambiente nelle versioni della piattaforma Tomcat rilasciate il o dopo il 26 marzo 2025 può causare comportamenti imprevisti nelle applicazioni che attribuiscono alle variabili di ambiente la precedenza sulle proprietà del sistema Java o quando si passa esplicitamente da a. `System.getProperty` `System.getenv` Per ulteriori informazioni e azioni consigliate, consulta Utilizzo della piattaforma Elastic Beanstalk Tomcat.

- .NET Core su Linux: `Environment.GetEnvironmentVariable`

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

- .NET: `appConfig`

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

- Node.js – `process.env`

```
var endpoint = process.env.API_ENDPOINT
```

- PHP: `$_SERVER`

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

- Python – `os.environ`

```
import os
endpoint = os.environ['API_ENDPOINT']
```

- Ruby: `ENV`

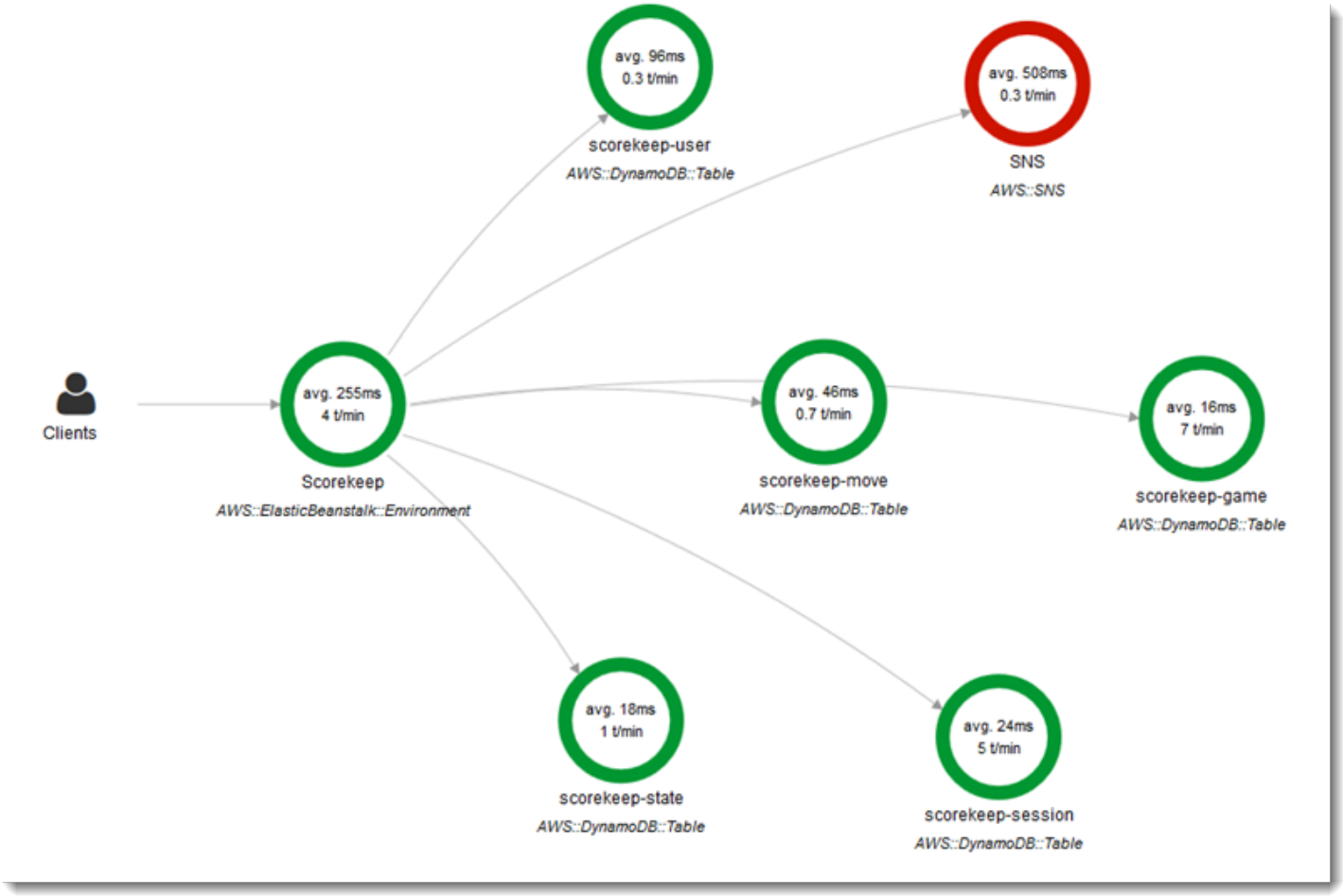
```
endpoint = ENV['API_ENDPOINT']
```

All'esterno del codice dell'applicazione, ad esempio in uno script eseguito durante la distribuzione, puoi accedere alle proprietà dell'ambiente con lo script di piattaforma `get-config`. Vedi il elastic-beanstalk-samples GitHub repository per esempi di configurazioni che utilizzano. `get-config`

Configurazione del debug AWS X-Ray

Puoi usare la AWS Elastic Beanstalk console o un file di configurazione per eseguire il AWS X-Ray demone sulle istanze del tuo ambiente. X-Ray è un AWS servizio che raccoglie dati sulle richieste servite dall'applicazione e li utilizza per creare una mappa dei servizi che è possibile utilizzare per identificare problemi con l'applicazione e opportunità di ottimizzazione.

> **Note**
>
> In alcune regioni non è possibile usufruire di X-Ray. Se crei un ambiente in una di queste regioni, non potrai eseguire il daemon X-Ray sulle istanze dell'ambiente.
> Per informazioni sui AWS servizi offerti in ciascuna regione, consulta la Tabella delle regioni.



X-Ray fornisce un SDK che è possibile utilizzare per il codice dell'applicazione e un'applicazione daemon che inoltra le informazioni di debug dall'SDK all'API X-Ray.

Piattaforme supportate

Puoi utilizzare SDK X-Ray con le seguenti piattaforme Elastic Beanstalk:

- Go - versione 2.9.1 e successive
- Java 8 - versione 2.3.0 e successive

- Java 8 con Tomcat 8 - versione 2.4.0 e successive
- Node.js - versione 3.2.0 e successive
- Windows Server - Tutte le versioni della piattaforma rilasciate a partire dal 18 dicembre 2016
- Python - versione 2.5.0 e successive

Nelle piattaforme supportate, è possibile utilizzare un'opzione di configurazione per eseguire il daemon X-Ray sulle istanze dell'ambiente. È possibile abilitare il daemon nella console di Elastic Beanstalk o utilizzando un file di configurazione.

Per caricare dati su X-Ray, il daemon X-Ray richiede le autorizzazioni IAM nella policy gestita. AWSXrayWriteOnlyAccess Queste autorizzazioni sono incluse nel profilo dell'istanza di Elastic Beanstalk. Se non utilizzi il profilo dell'istanza predefinito, consulta Assegnare al daemon l'autorizzazione per inviare i dati a X-Ray nella Guida per gli sviluppatori di AWS X-Ray .

Il debug con X-Ray richiede l'uso dell'SDK X-Ray. Per istruzioni e applicazioni di esempio, consulta Nozioni di base su AWS X-Ray nella Guida per gli sviluppatori di AWS X-Ray .

Se utilizzi una versione della piattaforma che non include il daemon, è comunque possibile eseguirlo con uno script in un file di configurazione. Per ulteriori informazioni, consulta Download ed esecuzione del daemon X-Ray in modalità manuale (avanzata) nella Guida per gli sviluppatori di AWS X-Ray .

Sections

- Configurazione del debug
- Spazio dei nomi aws:elasticbeanstalk:xray

Configurazione del debug

È possibile abilitare il daemon X-Ray su un ambiente in esecuzione nella console Elastic Beanstalk.

Per abilitare il debug nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).

4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Nella sezione Amazon X-Ray, seleziona Attivato.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

È anche possibile abilitare questa opzione durante la creazione dell'ambiente. Per ulteriori informazioni, consulta Procedura guidata per la creazione del nuovo ambiente.

Spazio dei nomi aws:elasticbeanstalk:xray

Puoi utilizzare l'opzione `XRayEnabled` nel namespace `aws:elasticbeanstalk:xray` per abilitare il debug.

Per abilitare il debug automaticamente quando distribuisci l'applicazione, imposta l'opzione in un file di configurazione del codice sorgente, come descritto di seguito.

Example.ebextensions/debugging.config

```
option_settings:
  aws:elasticbeanstalk:xray:
    XRayEnabled: true
```

Visualizzazione dei log dell'ambiente Elastic Beanstalk

AWS Elastic Beanstalk offre due modi per visualizzare regolarmente i log delle istanze Amazon EC2 che eseguono l'applicazione:

- Configurando l'ambiente Elastic Beanstalk per caricare i log delle istanze ruotati nel bucket Amazon S3 dell'ambiente.
- Configura l'ambiente per lo streaming dei log delle istanze su Amazon CloudWatch Logs.

Quando configuri lo streaming dei log delle istanze su CloudWatch Logs, Elastic CloudWatch Beanstalk crea gruppi di log di Logs per i log proxy e di distribuzione sulle istanze Amazon EC2 e trasferisce questi file di log a Logs in tempo reale. CloudWatch Per ulteriori informazioni sui log delle istanze, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Oltre ai log delle istanze, se abiliti l'integrità migliorata del tuo ambiente, puoi configurare l'ambiente in modo che trasmetta informazioni sullo stato di salute a Logs. CloudWatch Quando lo stato

dell'ambiente cambia, Elastic Beanstalk aggiunge un record a un gruppo di log sullo stato, con il nuovo stato e una descrizione della causa della modifica. Per informazioni sullo streaming delle informazioni sullo stato dell'ambiente, consulta Streaming delle informazioni sullo stato dell'ambiente Elastic Beanstalk su Amazon Logs CloudWatch.

Configurazione della visualizzazione dei log delle istanze

Per visualizzare i log delle istanze, puoi abilitare la rotazione dei log delle istanze e lo streaming dei log nella console Elastic Beanstalk.

Per configurare la rotazione dei log delle istanze e lo streaming dei log nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Nella sezione Archiviazione dei log S3, seleziona Attivato sotto Ruota log per abilitare il caricamento dei log ruotati su Amazon S3.
6. nella sezione Instance log streaming to CloudWatch Logs, configura le seguenti impostazioni:
   - Streaming di log: seleziona Attivato per abilitare lo streaming dei log.
   - Conservazione: specifica il numero di giorni in cui conservare i log nei CloudWatch registri.
   - Ciclo di vita: imposta su Elimina i log al termine per eliminare immediatamente i log dai CloudWatch log se l'ambiente è terminato, anziché attendere la loro scadenza.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Dopo aver abilitato lo streaming dei log, Elastic Beanstalk visualizza i gruppi di log in streaming CloudWatch nella console di gestione dell'ambiente, nella pagina Logs. Scegli un gruppo di log per visualizzarne gli eventi di registro in un visualizzatore integrato: Elastic Beanstalk seleziona automaticamente il flusso di log attivo più di recente e puoi cambiare stream, cercare e filtrare gli eventi e caricare gli eventi precedenti.

Configurazione della visualizzazione dei log sullo stato dell'ambiente

Per visualizzare i log sullo stato dell'ambiente, puoi abilitare lo streaming dei log sullo stato dell'ambiente nella console Elastic Beanstalk.

Per configurare lo streaming dei log sullo stato dell'ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri fino alla sezione Monitoraggio.
6. In Health event streaming to CloudWatch Logs, configura le seguenti impostazioni:
   - Streaming di log: scegli Attivato per abilitare lo streaming dei log.
   - Conservazione: specifica il numero di giorni in cui conservare i registri nei CloudWatch registri.
   - Ciclo di vita: imposta su Elimina i log al termine per eliminare immediatamente i log dai CloudWatch log se l'ambiente è terminato, anziché attendere la loro scadenza.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Spazi dei nomi per la visualizzazione dei log

Gli spazi dei nomi seguenti contengono le impostazioni per la visualizzazione dei log:

- `aws:elasticbeanstalk:hostmanager`: configura il caricamento dei log ruotati in Amazon S3.
- `aws:elasticbeanstalk:cloudwatch:logs`— Configurare lo streaming dei registri delle istanze su. CloudWatch
- `aws:elasticbeanstalk:cloudwatch:logs:health`— Configurare lo streaming dello stato dell'ambiente su CloudWatch.

Notifiche dell'ambiente Elastic Beanstalk con Amazon SNS

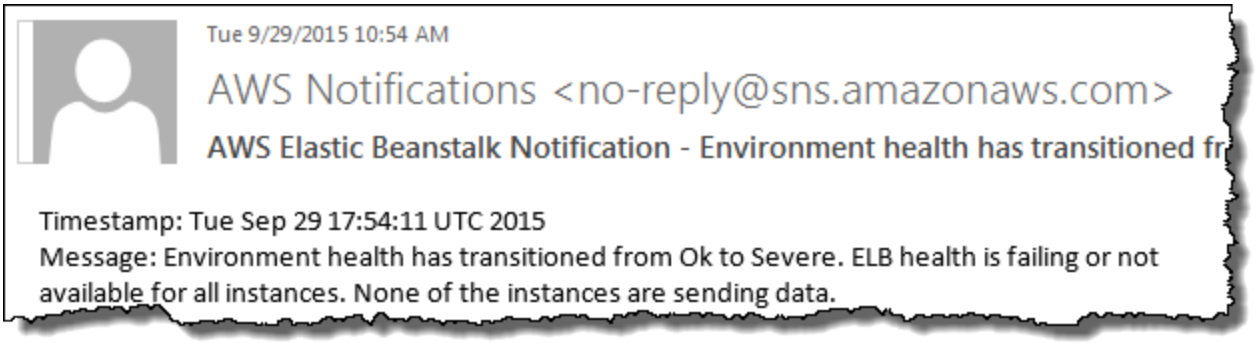
Puoi configurare il tuo AWS Elastic Beanstalk ambiente per utilizzare Amazon Simple Notification Service (Amazon SNS) per notificarti eventi importanti che influiscono sulla tua applicazione. Per

ricevere e-mail AWS ogni volta che si verifica un errore o lo stato dell'ambiente cambia, specifica un indirizzo e-mail quando crei un ambiente o in un secondo momento.

> **Note**
>
> Elastic Beanstalk utilizza Amazon SNS per le notifiche. Per informazioni sui prezzi di Amazon SNS, consulta https://aws.amazon.com/sns/ la pagina prezzi/.

Quando configuri le notifiche per il tuo ambiente, Elastic Beanstalk crea un argomento Amazon SNS per tuo conto. Per inviare messaggi a un argomento Amazon SNS, Elastic Beanstalk deve disporre dell'autorizzazione richiesta. Per ulteriori informazioni, consulta Configurazione delle autorizzazioni per l'invio di notifiche.

Quando si verifica un evento degno di nota, Elastic Beanstalk invia un messaggio all'argomento. Quindi, Amazon SNS inoltra i messaggi ricevuti agli abbonati dell'argomento. Gli eventi degni di nota includono gli errori di creazione di ambienti e tutte le modifiche allo stato dell'ambiente delle istanze. Gli eventi per le operazioni di Dimensionamento automatico Amazon EC2 (ad esempio, l'aggiunta e la rimozione di istanze dall'ambiente) e altri eventi informativi non attivano le notifiche.



È possibile immettere un indirizzo e-mail nella console Elastic Beanstalk quando si crea un ambiente o in un secondo momento. In questo modo verrà creato un argomento Amazon SNS a cui si effettuerà la sottoscrizione. Elastic Beanstalk gestisce il ciclo di vita dell'argomento e lo elimina quando l'ambiente viene terminato o quando l'indirizzo e-mail viene rimosso dalla console di gestione dell'ambiente.

Lo spazio dei nomi `aws:elasticbeanstalk:sns:topics` offre opzioni per la configurazione di un argomento Amazon SNS tramite i file di configurazione, una CLI o l'SDK. Utilizzando uno di questi metodi, puoi configurare il tipo di sottoscrittore e l'endpoint. Per tipo di sottoscrittore, puoi scegliere una coda Amazon SQS o un URL HTTP.

Le notifiche Amazon SNS possono essere solo attivate o disattivate. A seconda delle dimensioni e della composizione dell'ambiente, la frequenza delle notifiche inviate all'argomento può essere elevata. Per configurare le notifiche da inviare in circostanze specifiche, sono disponibili altre opzioni. Puoi configurare regole basate sugli eventi con Amazon EventBridge che ti avvisano quando Elastic Beanstalk emette eventi che soddisfano criteri specifici. Oppure, in alternativa, puoi configurare il tuo ambiente per pubblicare parametri personalizzati e impostare CloudWatch allarmi Amazon per avvisarti quando tali parametri raggiungono una soglia non corretta.

Configurazione delle notifiche utilizzando la console Elastic Beanstalk

La console Elastic Beanstalk ti consente di specificare un indirizzo e-mail per creare un argomento Amazon SNS per il tuo ambiente.

Per configurare le notifiche nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri verso il basso fino alla sezione Notifiche e-mail.
6. Immetti un indirizzo e-mail
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Quando specifichi un indirizzo e-mail per le notifiche, Elastic Beanstalk crea un argomento Amazon SNS per il tuo ambiente e aggiunge una sottoscrizione. Amazon SNS invia un'e-mail all'indirizzo che ha effettuato la sottoscrizione per conferma. È necessario fare clic sul link incluso nel messaggio e-mail di conferma per attivare la sottoscrizione e ricevere le notifiche.

Configurazione delle notifiche tramite le opzioni di configurazione

Utilizza le opzioni dello spazio dei nomi `aws:elasticbeanstalk:sns:topics` per configurare le notifiche Amazon SNS per il tuo ambiente. Puoi impostare queste opzioni utilizzando i file di configurazione, una CLI o un SDK.

- Endpoint di notifica: indirizzo e-mail, coda Amazon SQS o URL a cui inviare le notifiche. Se imposti questa opzione, vengono create una coda SQS e una sottoscrizione per l'endpoint specificato. Se l'endpoint non è un indirizzo e-mail, sarà necessario impostare anche l'opzione `Notification Protocol`. SNS convalida il valore di `Notification Endpoint` in base al valore di `Notification Protocol`. Impostando questa opzione più volte, si creano sottoscrizioni aggiuntive per l'argomento. Se rimuovi questa opzione, l'argomento sarà eliminato.
- Protocollo di notifica: il protocollo utilizzato per inviare notifiche a `Notification Endpoint`. L'impostazione predefinita di questa opzione è `email`. Imposta questa opzione su `email-json` per inviare messaggi e-mail in formato JSON `http` o `https` per pubblicare le notifiche in formato JSON in un endpoint HTTP oppure `sqs` per inviare notifiche a una coda SQS.

> **Note**
>
> AWS Lambda le notifiche non sono supportate.

- ARN dell'argomento di notifica: dopo aver impostato un endpoint di notifica per l'ambiente, leggi questa impostazione per ottenere l'ARN dell'argomento SNS. Puoi impostare questa opzione anche per utilizzare un argomento SNS esistente per le notifiche. Un argomento che colleghi al tuo ambiente utilizzando questa opzione non viene eliminato quando modifichi l'opzione o termini l'ambiente.

  Per configurare le notifiche Amazon SNS, devi disporre delle autorizzazioni necessarie. Se il tuo utente IAM utilizza la policy utente gestita da AdministratorAccessElastic AWSElastic Beanstalk - Beanstalk, allora dovresti già disporre delle autorizzazioni necessarie per configurare l'argomento Amazon SNS predefinito creato da Elastic Beanstalk per il tuo ambiente. Tuttavia, quando configuri un argomento Amazon SNS che non è gestito da Elastic Beanstalk, sarà necessario aggiungere la seguente policy al tuo ruolo utente.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:SetTopicAttributes",
        "sns:GetTopicAttributes",
        "sns:Subscribe",
```

```
        "sns:Unsubscribe",
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
  ]
}
```

- Nome dell'argomento di notifica: imposta questa opzione per personalizzare il nome dell'argomento Amazon SNS utilizzato per le notifiche dell'ambiente. Se esiste già un argomento con lo stesso nome, Elastic Beanstalk lo collega all'ambiente.

> **Warning**
>
> Se colleghi un argomento SNS esistente a un ambiente con `Notification Topic Name`, Elastic Beanstalk elimina l'argomento nel caso in cui successivamente l'ambiente viene terminato o questa impostazione viene modificata.

Se modifichi questa opzione, viene modificato anche il `Notification Topic ARN`. Se un argomento è già collegato all'ambiente, Elastic Beanstalk elimina il vecchio argomento, quindi ne crea uno nuovo insieme alla sottoscrizione.

Utilizzando un nome di argomento personalizzato, dovrai inoltre fornire un ARN di un argomento personalizzato creato esternamente. La policy degli utenti gestiti non rileva automaticamente un argomento con un nome personalizzato, pertanto agli utenti IAM dovrai fornire autorizzazioni Amazon SNS personalizzate. Utilizza una policy simile a quella utilizzata per l'ARN di un argomento personalizzato, ma includi quanto riportato di seguito:

- Includi altre due azioni nell'elenco `Actions`, in particolare `sns:CreateTopic`, `sns:DeleteTopic`
- Se stai cambiando il nome di un argomento personalizzato `Notification Topic Name` a un altro, devi includere anche entrambi gli argomenti nell'elenco. ARNs `Resource` In alternativa, includi un'espressione regolare che copra entrambi. In questo modo Elastic Beanstalk dispone delle autorizzazioni per eliminare il vecchio argomento e crearne uno nuovo.

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Configurazione delle autorizzazioni per l'invio di notifiche

In questa sezione vengono illustrate le considerazioni sulla sicurezza relative alle notifiche che utilizzano Amazon SNS. Ci sono due casi distinti:

- Utilizza l'argomento predefinito di Amazon SNS creato da Elastic Beanstalk per il tuo ambiente.
- Fornisci un argomento Amazon SNS esterno tramite le opzioni di configurazione.

La policy di accesso predefinita per un argomento Amazon SNS consente solo al proprietario dell'argomento di pubblicarlo o iscriversi. Tuttavia, attraverso la corretta configurazione delle policy, Elastic Beanstalk può essere autorizzato a pubblicare su un argomento Amazon SNS in uno dei due casi descritti in questa sezione. Le sezioni seguenti forniscono ulteriori informazioni.

Autorizzazioni per un argomento predefinito

Quando configuri le notifiche per il tuo ambiente, Elastic Beanstalk crea un argomento Amazon SNS per il tuo ambiente. Per inviare messaggi a un argomento Amazon SNS, Elastic Beanstalk deve disporre dell'autorizzazione richiesta. Se il tuo ambiente utilizza il ruolo di servizio appositamente generato dalla console Elastic Beanstalk o dalla CLI di EB oppure il ruolo collegato ai servizi di monitoraggio dell'account, non è necessario eseguire altre operazioni. Questi ruoli gestiti includono l'autorizzazione necessaria che consente a Elastic Beanstalk di inviare messaggi all'argomento Amazon SNS.

Tuttavia, se al momento della creazione dell'ambiente hai fornito un ruolo di servizio personalizzato, accertati che quest'ultimo includa la policy seguente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
```

```
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:ElasticBeanstalkNotifications*"
      ]
    }
  ]
}
```

Autorizzazioni per un argomento esterno

In Configurazione delle notifiche tramite le opzioni di configurazione, spieghiamo come è possibile sostituire l'argomento Amazon SNS fornito da Elastic Beanstalk con un altro argomento Amazon SNS. Se l'argomento viene sostituito, Elastic Beanstalk dovrà verificare che nel nuovo argomento SNS sia presente l'autorizzazione per la pubblicazione in modo da poterlo associare all'ambiente. Dovrebbe essere presente `sns:Publish`. Il ruolo di servizio utilizza la stessa autorizzazione. Per verificarlo, Elastic Beanstalk invia una notifica di prova a SNS nell'ambito delle operazioni di creazione o aggiornamento dell'ambiente. Se questo test non riesce, anche il tentativo di creare o aggiornare l'ambiente avrà esito negativo. Elastic Beanstalk restituisce quindi un messaggio che spiega il motivo di questo errore.

Se fornisci un ruolo di servizio personalizzato per il tuo ambiente, assicurati che il ruolo del servizio personalizzato includa la policy riportata di seguito per consentire a Elastic Beanstalk di inviare messaggi all'argomento Amazon SNS. Nel codice seguente, sostituisci *`sns_topic_name`* con il nome dell'argomento Amazon SNS fornito nelle opzioni di configurazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
  ]
```

```
}
```

Per maggiori informazioni sul controllo degli accessi di Amazon SNS, consulta Esempi di casi per il controllo degli accessi Amazon SNS nella Guida per sviluppatori di Amazon Simple Notification Service.

Configurazione di Amazon Virtual Private Cloud (Amazon VPC) con Elastic Beanstalk

Amazon Virtual Private Cloud (Amazon VPC) è il servizio di rete che indirizza il traffico in modo sicuro verso EC2 le istanze che eseguono l'applicazione in Elastic Beanstalk. Se non configuri un VPC all'avvio del tuo ambiente, Elastic Beanstalk utilizza il VPC predefinito.

Puoi avviare l'ambiente in un VPC personalizzato per specificare le impostazioni di rete e di sicurezza. Elastic Beanstalk ti consente di scegliere quali sottoreti utilizzare per le risorse e come configurare gli indirizzi IP per le istanze e il sistema di bilanciamento del carico nel tuo ambiente. Quando viene creato, un ambiente è bloccato in un VPC, ma è possibile modificare le impostazioni relative a sottorete e indirizzo IP su un ambiente di esecuzione.

Configurazione delle impostazioni VPC nella console Elastic Beanstalk

Se durante la creazione dell'ambiente hai scelto un VPC personalizzato, puoi modificarne le impostazioni nella console Elastic Beanstalk.

Per configurare le impostazioni VPC dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Network (Rete), scegliere Edit (Modifica).

Sono disponibili le impostazioni seguenti.

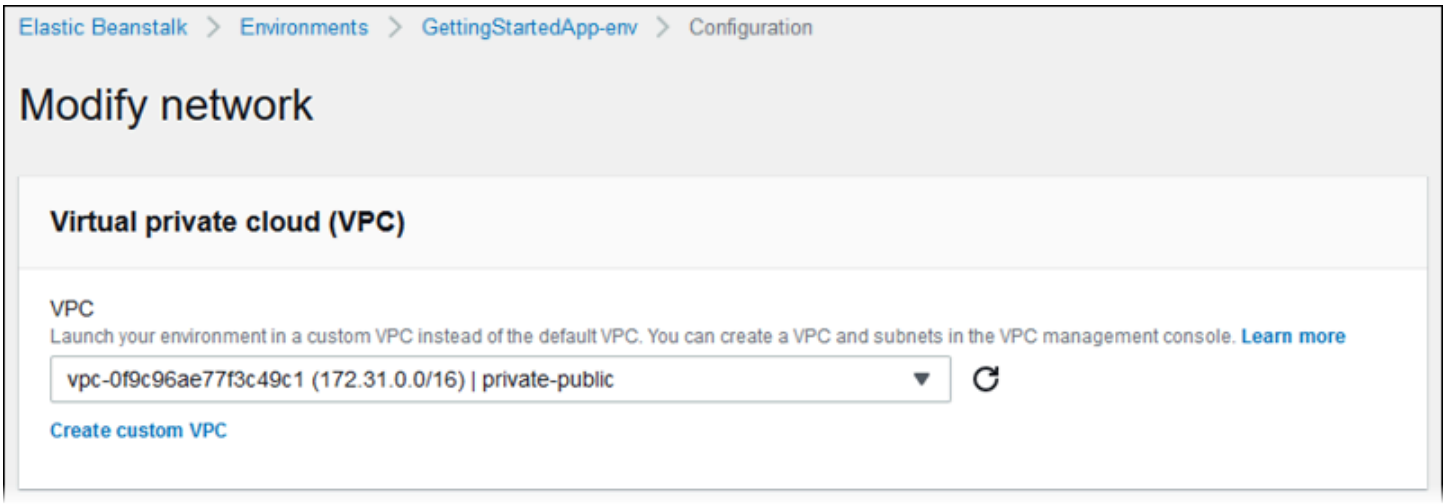
Opzioni

- VPC

- Visibilità del sistema di bilanciamento del carico
- Sottoreti del sistema di bilanciamento del carico
- Indirizzo IP pubblico delle istanze
- Sottoreti delle istanze
- Sottoreti del database

VPC

Scegli un VPC per l'ambiente. Puoi modificare questa impostazione soltanto durante la creazione dell'ambiente.

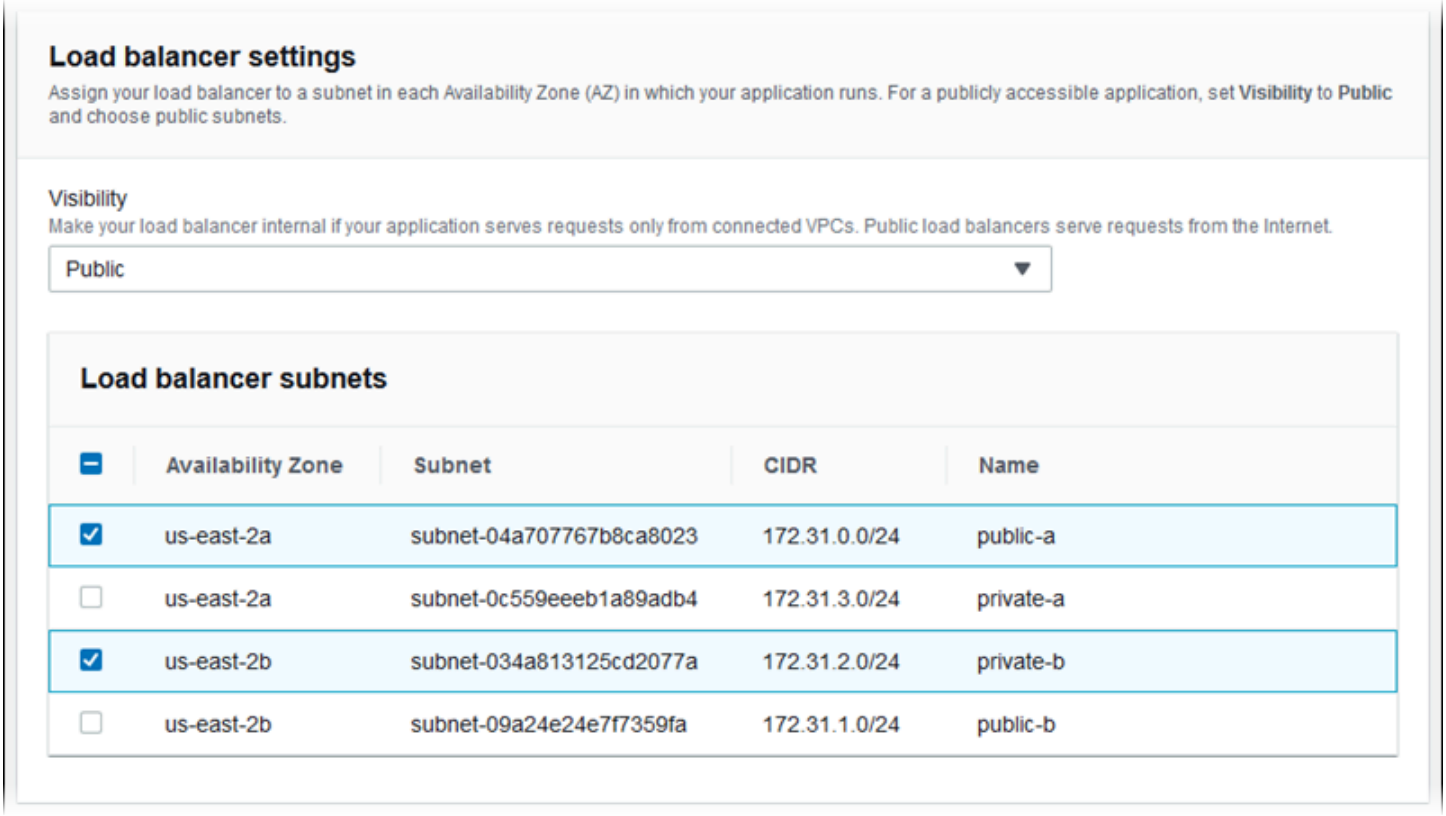


Visibilità del sistema di bilanciamento del carico

Per un ambiente con sistema di bilanciamento del carico, scegli il relativo schema. Per impostazione predefinita, il sistema di bilanciamento del carico è pubblico, con indirizzo IP e nome di dominio pubblici. Se la tua applicazione gestisce solo il traffico dall'interno del tuo VPC o da una connessione VPN, deseleziona questa opzione e scegli le sottoreti private per il tuo sistema di bilanciamento del carico, in modo da rendere quest'ultimo interno e disabilitare l'accesso da Internet.

Sottoreti del sistema di bilanciamento del carico

Per un ambiente con sistema di bilanciamento del carico, scegli le sottoreti che il sistema di bilanciamento del carico impiega per trasmettere il traffico. Per un'applicazione pubblica, scegli sottoreti pubbliche. Per un'elevata disponibilità, utilizza sottoreti in più zone di disponibilità. Per un'applicazione interna, scegli sottoreti private e disabilita la visibilità del sistema di bilanciamento del carico.

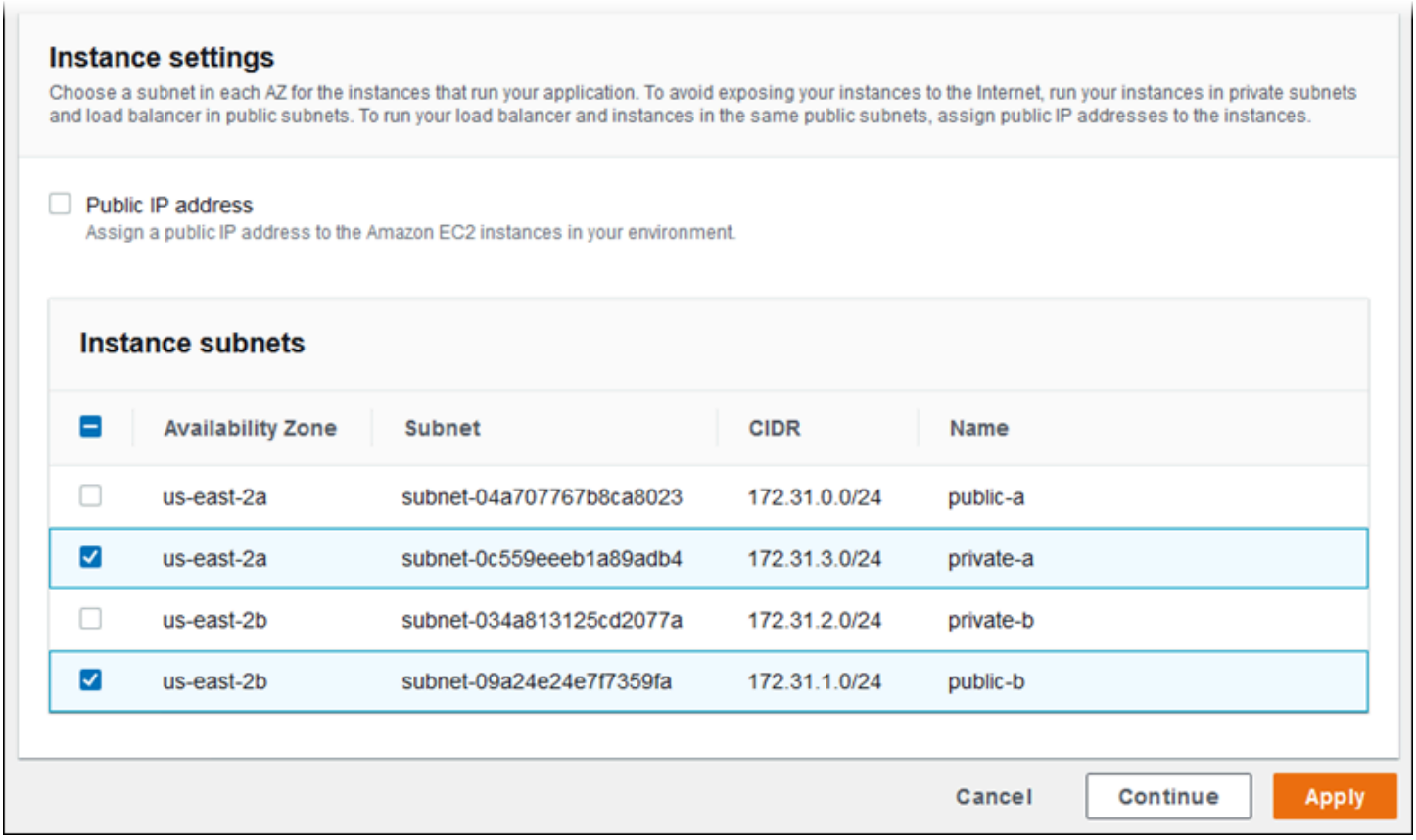


Indirizzo IP pubblico delle istanze

Se scegli sottoreti pubbliche per le istanze della tua applicazione, abilita gli indirizzi IP pubblici per renderle instradabili da Internet.

Sottoreti delle istanze

Scegli le sottoreti per le istanze dell'applicazione. Scegli almeno una sottorete per ogni zona di disponibilità utilizzata dal tuo sistema di bilanciamento del carico. Se scegli sottoreti private per le istanze, il tuo VPC deve disporre di un gateway NAT in una sottorete pubblica che le istanze possono utilizzare per accedere a Internet.



Sottoreti del database

Quando esegui un database Amazon RDS collegato al tuo ambiente Elastic Beanstalk, scegli le sottoreti per le istanze database. Per un'elevata disponibilità, rendi il database AZ multiple e scegli una sottorete per ogni zona di disponibilità. Per garantire che la tua applicazione sia in grado di connettersi al database, esegui entrambi nella stessa sottorete.

Spazio dei nomi aws:ec2:vpc

Puoi utilizzare le opzioni di configurazione nello spazio dei nomi `aws:ec2:vpc` per configurare le impostazioni di rete dell'ambiente.

Il seguente file di configurazione utilizza le opzioni di questo spazio dei nomi per impostare il VPC e le sottoreti dell'ambiente per una configurazione di tipo pubblico-privato. Per impostare l'ID VPC in un file di configurazione, il file deve essere incluso nel bundle di origine dell'applicazione durante la creazione dell'ambiente. Consulta Impostazione delle opzioni di configurazione durante la creazione di un ambiente per altri metodi di configurazione di queste impostazioni durante la creazione dell'ambiente.

Example.ebextensions/vpc.config - Pubblico-privato

```
option_settings:
   aws:ec2:vpc:
      VPCId: vpc-087a68c03b9c50c84
      AssociatePublicIpAddress: 'false'
      ELBScheme: public
      ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
      Subnets: subnet-026c6117b178a9c45,subnet-0839e902f656e8bd1
```

Questo esempio mostra una configurazione pubblica-pubblica, in cui il sistema di bilanciamento del carico e le EC2 istanze vengono eseguiti nelle stesse sottoreti pubbliche.

Example.ebextensions/vpc.config - Pubblico-pubblico

```
option_settings:
   aws:ec2:vpc:
      VPCId: vpc-087a68c03b9c50c84
      AssociatePublicIpAddress: 'true'
      ELBScheme: public
      ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
      Subnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
```

Migrazione degli ambienti Elastic EC2 Beanstalk da -Classic a un VPC

Questo argomento descrive diverse opzioni su come migrare gli ambienti Elastic Beanstalk EC2 da una piattaforma di rete -Classic a una rete Amazon Virtual Private Cloud (Amazon VPC).

Se hai creato il tuo AWS account prima del 4 dicembre 2013, in alcuni ambienti potresti avere ambienti che utilizzano la EC2 configurazione di rete -Classic. Regioni AWS Tutti AWS gli account creati a partire dal 4 dicembre 2013 sono già riservati al solo VPC in ogni regione. AWS Le uniche esenzioni sono se Amazon EC2 -Classic è stato abilitato a seguito di una richiesta di supporto.

> **Note**
>
> È possibile visualizzare le impostazioni di configurazione di rete dell'ambiente nella categoria Network configuration (Configurazione di rete) nella pagina Panoramica della configurazione della console Elastic Beanstalk.

Perché è necessario eseguire la migrazione

EC2Amazon-Classic raggiungerà la fine del supporto standard il 15 agosto 2022. Per evitare interruzioni dei carichi di lavoro, ti consigliamo di migrare da Amazon EC2 -Classic a un VPC prima del 15 agosto 2022. Ti chiediamo inoltre di non lanciare alcuna AWS risorsa su Amazon EC2 -Classic in futuro e di utilizzare invece Amazon VPC.

Quando esegui la migrazione degli ambienti Elastic Beanstalk da EC2 Amazon -Classic ad Amazon VPC, devi creare un nuovo account. AWS Inoltre, devi ricreare gli ambienti -Classic nel tuo AWS EC2 nuovo account. AWS Non è necessario eseguire ulteriori operazioni di configurazione per gli ambienti per utilizzare il VPC predefinito. Se il VPC predefinito non soddisfa i requisiti, crea manualmente un VPC personalizzato e associalo agli ambienti.

In alternativa, se il tuo AWS account esistente dispone di risorse che non puoi migrare su un nuovo AWS account, aggiungi un VPC al tuo account corrente. Quindi, configura gli ambienti in modo che utilizzi il VPC.

Per ulteriori informazioni, consulta il post sul blog EC2-Classic Networking is Retiring - Here's How to Prepare.

Migrare un ambiente da EC2 -Classic a un nuovo AWS account (consigliato)

Se non disponi già di un AWS account creato il 4 dicembre 2013 o dopo tale data, crea un nuovo account. Eseguirai la migrazione degli ambienti in questo nuovo account.

1. Il tuo nuovo AWS account fornisce un VPC predefinito ai suoi ambienti. Se non hai bisogno di creare un VPC personalizzato, passa alla fase 2.

   È possibile creare un VPC personalizzato in uno dei seguenti modi:

   - Crea velocemente un VPC utilizzando la procedura guidata della console Amazon VPC con una delle opzioni di configurazione disponibili. Per ulteriori informazioni, consulta Configurazioni della procedura guidata della console Amazon VPC.
   - Crea un VPC personalizzato sulla console Amazon VPC se hai requisiti più specifici per il VPC. Si consiglia di farlo, ad esempio, se il caso d'uso richiede un numero specifico di sottoreti. Per ulteriori informazioni, consulta VPCs e sottoreti.
   - Crea un VPC utilizzando l'elastic-beanstalk-samplesarchivio sul GitHub sito Web se preferisci utilizzare i AWS CloudFormation modelli con i tuoi ambienti Elastic Beanstalk. Questo

repository include modelli. CloudFormation Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon VPC.

> **Note**
>
> Puoi anche creare un VPC personalizzato mentre ricrei l'ambiente nel tuo nuovo AWS account utilizzando la procedura guidata per la creazione di un nuovo ambiente. Se usi la procedura guidata e scegli di creare un VPC personalizzato, sarai reindirizzato alla console Amazon VPC.

2. Nel tuo nuovo AWS account, crea un nuovo ambiente. Ti consigliamo che l'ambiente includa la stessa configurazione dell'ambiente esistente nell' AWS account da cui stai effettuando la migrazione. Puoi effettuare questa operazione utilizzando uno dei seguenti approcci:

> **Note**
>
> Se il nuovo ambiente deve utilizzare lo stesso CNAME dopo la migrazione, interrompi l'ambiente originale sulla piattaforma -Classic. EC2 Questo rilascia il CNAME per l'utilizzo. Tuttavia, ciò può causare tempi di inattività per quell'ambiente e può anche rischiare che un altro cliente scelga il CNAME tra la chiusura dell'ambiente EC2 -Classic e la creazione di uno nuovo. Per ulteriori informazioni, consulta Terminazione di un ambiente Elastic Beanstalk.
> Per gli ambienti che hanno il proprio nome di dominio proprietario, il CNAME non presenta questo problema. Puoi semplicemente aggiornare il Domain Name System (DNS) per inoltrare le richieste al nuovo CNAME.

- Utilizza la procedura guidata per la creazione del nuovo ambiente nella console Elastic Beanstalk. La procedura guidata fornisce un'opzione per creare un VPC personalizzato. Se non si sceglie di creare un VPC personalizzato, verrà assegnato un VPC predefinito.
- Usa l'interfaccia a riga di comando di Elastic Beanstalk (EB CLI) per ricreare l'ambiente nel tuo nuovo account. AWS Uno degli esempi nella descrizione del comando eb create illustra la creazione di un ambiente in un VPC personalizzato. Se non si fornisce un ID VPC, l'ambiente utilizzerà il VPC predefinito.

  Utilizzando questo approccio, puoi utilizzare un file di configurazione salvato tra i due account. AWS Di conseguenza, non è necessario inserire manualmente tutte le informazioni

di configurazione. Tuttavia, è necessario salvare le impostazioni di configurazione per l'ambiente EC2 -Classic che si sta migrando con il comando eb config save. Copiare il file di configurazione salvato in una nuova directory per il nuovo ambiente dell'account.

> **Note**
>
> Devi modificare alcuni dei dati nel file di configurazione salvato prima di poterli utilizzare nel nuovo account. Devi aggiornare anche le informazioni relative al tuo account precedente con i dati corretti del nuovo account. Ad esempio, è necessario sostituire l'Amazon Resource Name (ARN) del ruolo AWS Identity and Access Management (IAM) con l'ARN del ruolo IAM per il nuovo account.

Se si utilizza il comando eb create con il `cfg`, il nuovo ambiente viene creato utilizzando il file di configurazione salvato specificato. Per ulteriori informazioni, consulta Utilizzo delle configurazioni salvate di Elastic Beanstalk.

Esegui la migrazione di un ambiente da EC2 -Classic all'interno del tuo stesso account AWS

Il tuo AWS account esistente potrebbe avere risorse che non puoi migrare su un nuovo account. AWS In questo caso, è necessario ricreare gli ambienti e configurare manualmente un VPC per ogni ambiente creato.

Eseguire la migrazione degli ambienti a un VPC personalizzato

Prerequisiti

Prima di iniziare, è necessario disporre di un VPC. È possibile creare un VPC non predefinito (personalizzato) in uno dei seguenti modi:

- Crea velocemente un VPC utilizzando la procedura guidata della console Amazon VPC con una delle opzioni di configurazione disponibili. Per ulteriori informazioni, consulta Configurazioni della procedura guidata della console Amazon VPC.
- Crea un VPC personalizzato sulla console Amazon VPC se hai requisiti più specifici per il VPC. Si consiglia di farlo, ad esempio, se il caso d'uso richiede un numero specifico di sottoreti. Per ulteriori informazioni, consulta VPCs e sottoreti.

- Crea un VPC utilizzando l'elastic-beanstalk-samplesarchivio sul GitHub sito Web se preferisci utilizzare i AWS CloudFormation modelli con i tuoi ambienti Elastic Beanstalk. Questo repository include modelli. CloudFormation Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon VPC.

Nei passaggi seguenti, si utilizzano l'ID VPC e la sottorete generati IDs quando si configura il VPC nel nuovo ambiente.

1. Creazione di un nuovo ambiente che includa la stessa configurazione dell'ambiente esistente. Puoi effettuare questa operazione utilizzando uno dei seguenti approcci:

   > **Note**
   >
   > La caratteristica Configurazioni salvate consente di ricreare gli ambienti nel nuovo account. Questa funzione consente di salvare la configurazione di un ambiente, in modo da poterla applicare quando si creano o si aggiornano altri ambienti. Per ulteriori informazioni, consulta Utilizzo delle configurazioni salvate di Elastic Beanstalk.

   - Utilizzando la console Elastic Beanstalk, applica una configurazione salvata EC2 dal tuo ambiente -Classic quando configuri il nuovo ambiente. Questa configurazione utilizzerà il VPC. Per ulteriori informazioni, consultare Utilizzo delle configurazioni salvate di Elastic Beanstalk.
   - Utilizzando l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) esegui il comando eb create per ricreare l'ambiente. Fornire i parametri dell'ambiente originale e l'identificatore del VPC. Uno degli esempi nella descrizione del comando eb create mostra la creazione di un ambiente in un VPC personalizzato.
   - Usa AWS Command Line Interface (AWS CLI) e ricrea il tuo ambiente usando il comando. elasticbeanstalk create-environment Fornire i parametri dell'ambiente originale con l'identificatore del VPC. Per istruzioni, consulta Creazione di ambienti Elastic Beanstalk con la CLI AWS.

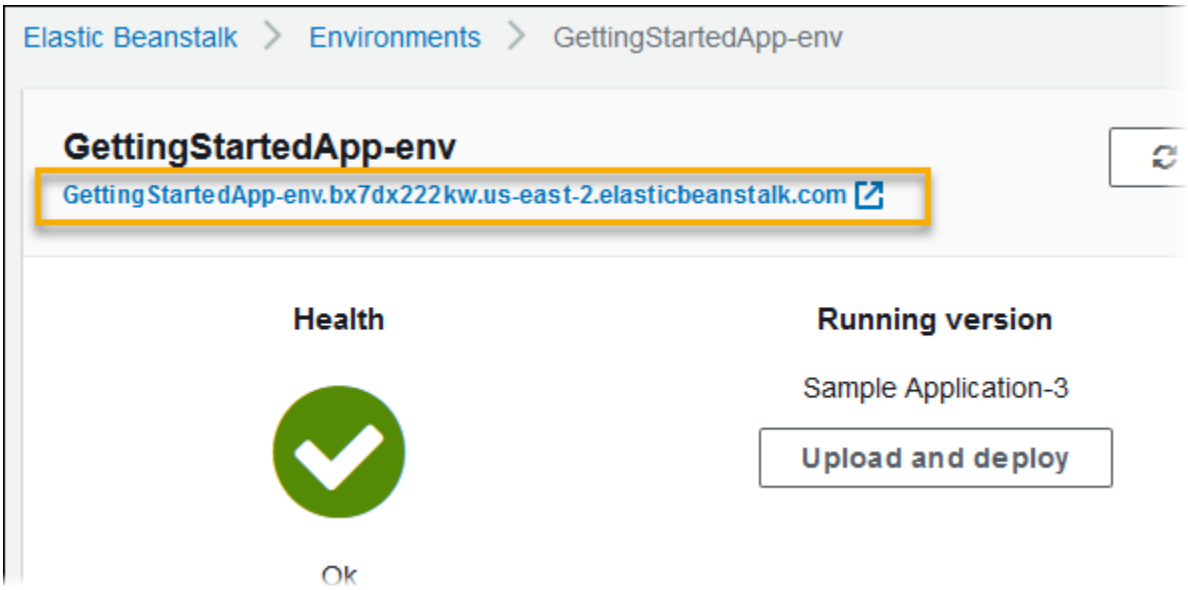
2. Scambia CNAMEs l'ambiente esistente con il nuovo ambiente. In questo modo, può essere fatto riferimento al nuovo ambiente creato con l'indirizzo conosciuto. È possibile utilizzare la CLI EB o AWS CLI.

- Utilizzando l'EB CLI, scambiate l'ambiente CNAMEs eseguendo il comando. eb swap Per ulteriori informazioni, consulta Configurazione dell'interfaccia a riga di comando EB (EB CLI) per gestire Elastic Beanstalk.
- Usando AWS CLI, scambia l'ambiente CNAMEs con il comando elasticbeanstalk. swap-environment-cnames Per ulteriori informazioni, consulta la sezione relativa alle informazioni di riferimento ai comandi di AWS CLI.

Nome di dominio dell'ambiente Elastic Beanstalk

Per impostazione predefinita, l'ambiente è disponibile per gli utenti in un sottodominio di `elasticbeanstalk.com`. Quando crei un ambiente, puoi scegliere un nome host per l'applicazione. Il sottodominio e il dominio vengono popolati automaticamente in *`region`*`.elasticbeanstalk.com`.

Per indirizzare gli utenti all'ambiente, Elastic Beanstalk registra un record CNAME che punta al sistema di bilanciamento del carico dell'ambiente. È possibile visualizzare l'URL dell'applicazione dell'ambiente con il valore corrente del CNAME nella pagina di panoramica dell'ambiente della console Elastic Beanstalk.



Scegliere l'URL nella pagina panoramica oppure scegliere Go to environment (Vai all'ambiente) nel riquadro di navigazione per accedere alla pagina Web dell'applicazione.

Puoi modificare il CNAME nel tuo ambiente scambiandolo con il CNAME di un altro ambiente. Per istruzioni, consulta Distribuzioni blu/verde con Elastic Beanstalk.

Se disponi di un nome di dominio, puoi utilizzare Amazon Route 53 per risolverlo nell'ambiente. È possibile acquistare un nome di dominio con Amazon Route 53 oppure utilizzarne uno acquistato da un altro provider.

Per acquistare un nome di dominio con Route 53, consultare Registrazione di un nuovo dominio nella Guida per gli sviluppatori di Amazon Route 53.

Per ulteriori informazioni sull'utilizzo di un dominio personalizzato, consulta Routing del traffico a un ambiente AWS Elastic Beanstalk nella Guida per gli sviluppatori di Amazon Route 53.

> **Important**
>
> Se un ambiente viene terminato, è necessario anche eliminare eventuali mappature CNAME create, poiché altri clienti possono riutilizzare un nome host disponibile. Assicurati di eliminare i record DNS che puntano al tuo ambiente terminato per evitare una voce DNS inesatta. Una voce DNS inesatta può esporre il traffico Internet destinato al tuo dominio a vulnerabilità di sicurezza. Può presentare anche altri rischi.
> Per ulteriori informazioni, consulta l'argomento relativo alla Protezione dai record di delega inesatti in Route 53 nella Guida per sviluppatori di Amazon Route 53. Puoi anche saperne di più sulle voci DNS sospese in Enhanced Domain Protections for Amazon CloudFront Requests nel Security Blog.AWS

Configurazione degli ambienti Elastic Beanstalk (avanzata)

Quando crei un AWS Elastic Beanstalk ambiente, Elastic Beanstalk effettua il provisioning e configura tutte le AWS risorse necessarie per eseguire e supportare l'applicazione. Oltre a configurare i metadati e il comportamento di aggiornamento dell'ambiente, puoi personalizzare queste risorse specificando valori per le opzioni di configurazione. Ad esempio, potresti voler aggiungere una coda Amazon SQS e un allarme sulla profondità della coda oppure potresti voler aggiungere un cluster Amazon. ElastiCache

La maggior parte delle opzioni di configurazione presenta valori predefiniti che vengono applicati automaticamente da Elastic Beanstalk. Puoi sostituire queste impostazioni predefinite con file di configurazione, configurazioni salvate, opzioni della riga di comando o chiamando direttamente l'API Elastic Beanstalk. La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per alcune opzioni.

Includendo un file di configurazione nel bundle di origine, puoi personalizzare agevolmente il tuo ambiente in contemporanea con la distribuzione della versione dell'applicazione. Quando si personalizza il software sulla propria istanza, è più vantaggioso utilizzare un file di configurazione piuttosto che creare un'AMI personalizzata perché non è necessario mantenere un set di. AMIs

Quando distribuisci le applicazioni puoi personalizzare e configurare il software da cui dipende l'applicazione. Questi file potrebbero essere dipendenze richieste dall'applicazione (ad esempio, pacchetti aggiuntivi dal repository yum) oppure file di configurazione (ad esempio, un file che sostituisce httpd.conf e annulla specifici valori predefiniti da AWS Elastic Beanstalk).

Argomenti

- Opzioni di configurazione
- Personalizzazione avanzata dell'ambiente con i file di configurazione (.ebextensions)
- Utilizzo delle configurazioni salvate di Elastic Beanstalk
- Manifest di ambiente (env.yaml)
- Utilizzo di un'immagine Amazon Machine Image (AMI) personalizzata nell'ambiente Elastic Beanstalk
- Fornire i file statici
- Configurazione del protocollo HTTPS per l'ambiente Elastic Beanstalk

Opzioni di configurazione

Elastic Beanstalk definisce un numero elevato di opzioni di configurazione che puoi utilizzare per configurare il comportamento dell'ambiente e le risorse che contiene. Le opzioni di configurazione sono organizzate in spazi dei nomi, ad esempio `aws:autoscaling:asg`, che definiscono le opzioni per un gruppo Auto Scaling dell'ambiente.

La console Elastic Beanstalk e la CLI EB impostano le opzioni di configurazione al momento della creazione di un ambiente, incluse le opzioni che imposti in modo esplicito e i valori consigliati definiti dal client. Puoi anche impostare le opzioni di configurazione in configurazioni salvate e nei file di configurazione. Se la stessa opzione è impostata in più posizioni, il valore utilizzato viene stabilito in base all'ordine di precedenza.

Le impostazioni delle opzioni di configurazione possono essere in formato di testo e salvate prima della creazione dell'ambiente, applicate durante la creazione dell'ambiente utilizzando qualsiasi client supportato e aggiunte, modificate o rimosse dopo la creazione dell'ambiente. Per un elenco dettagliato di tutti i metodi disponibili per l'utilizzo delle opzioni di configurazione in ciascuna di queste tre fasi, consulta i seguenti argomenti:

- Impostazione delle opzioni di configurazione prima della creazione di un ambiente
- Impostazione delle opzioni di configurazione durante la creazione di un ambiente
- Impostazione delle opzioni di configurazione dopo la creazione di un ambiente

Per un elenco completo delle opzioni e dei namespace, inclusi i valori supportati e quelli predefiniti, vedi Opzioni generali per tutti gli ambienti e Opzioni specifiche della piattaforma.

Priorità

Durante la creazione dell'ambiente, le opzioni di configurazione vengono applicate da più origini con le seguenti priorità, dalla più alta alla più bassa:

- Impostazioni applicate direttamente all'ambiente: impostazioni specificate durante un'operazione di creazione o aggiornamento dell'ambiente sull'API Elastic Beanstalk da qualsiasi client, inclusa la console Elastic Beanstalk, EB CLI e. AWS CLI SDKs La CLI EB e la console Elastic Beanstalk applicano anche valori consigliati per alcune opzioni a questo livello, a meno che non siano sostituite.
- Configurazioni salvate: le impostazioni per le opzioni non applicate direttamente all'ambiente vengono caricate da una configurazione salvata, se specificato.

- File di configurazione (.ebextensions): le impostazioni per le opzioni non applicate direttamente all'ambiente e non specificate in una configurazione salvata vengono caricate dai file di configurazione nella cartella `.ebextensions` alla radice del bundle di origine dell'applicazione.

  I file di configurazione vengono eseguiti in ordine alfabetico. Ad esempio, `.ebextensions/01run.config` viene eseguito prima di `.ebextensions/02do.config`.
- Valori predefiniti: se un'opzione di configurazione dispone di un valore predefinito, viene applicato solo quando l'opzione non è impostata su nessuno dei suddetti livelli.

Se la stessa opzione di configurazione è definita in più di un percorso, viene applicata l'impostazione con la priorità massima. Quando un'impostazione è applicata da una configurazione salvata o le impostazioni vengono applicate direttamente all'ambiente, l'impostazione viene memorizzata come parte della configurazione dell'ambiente. Queste impostazioni possono essere annullate con la AWS CLI o con la CLI EB.

Le impostazioni nei file di configurazione non vengono applicate direttamente all'ambiente e non possono essere rimosse senza modificare i file di configurazione e distribuire una nuova versione dell'applicazione. Se un'impostazione applicata con uno degli altri metodi viene rimossa, la stessa impostazione sarà caricata dai file di configurazione nel bundle di origine.

Ad esempio, è possibile impostare il numero minimo di istanze nell'ambiente su 5 al momento della creazione dell'ambiente, utilizzando la console Elastic Beanstalk, un'opzione della riga di comando o una configurazione salvata. Il bundle di origine per l'applicazione include anche un file di configurazione che consente di impostare il numero minimo di istanze su 2.

Quando si crea l'ambiente, Elastic Beanstalk imposta l'opzione `MinSize` nello spazio dei nomi `aws:autoscaling:asg` su 5. Se quindi rimuovi l'opzione dalla configurazione dell'ambiente, viene caricato il valore del file di configurazione e il numero minimo di istanze è impostato su 2. Se elimini il file di configurazione dal bundle di origine e ridistribuisci, Elastic Beanstalk utilizza l'impostazione predefinita di 1.

Valori consigliati

L'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) e la console Elastic Beanstalk forniscono i valori consigliati per alcune opzioni di configurazione. Questi valori possono essere diversi dai valori predefiniti e vengono impostati a livello di API al momento della creazione dell'ambiente. I valori consigliati permettono a Elastic Beanstalk di migliorare la configurazione predefinita dell'ambiente senza applicare modifiche non compatibili con la versione precedente all'API.

Ad esempio, sia l'EB CLI che la console Elastic Beanstalk impostano l'opzione di configurazione EC2 per il tipo di istanza (nello spazio dei nomi). `InstanceType` `aws:autoscaling:launchconfiguration` Ogni client fornisce un modo diverso di ignorare l'impostazione predefinita. Nella console è possibile scegliere un altro tipo di istanza da un menu a discesa nella pagina Configuration Details (Dettagli della configurazione) della procedura guidata Create New Environment (Crea nuovo ambiente). Con la CLI EB, puoi usare il parametro `--instance_type` per eb create.

Poiché i valori consigliati vengono impostati a livello di API, questi sostituiranno i valori per le stesse opzioni che hai impostato nei file di configurazione o nelle configurazioni salvate. Vengono configurate le opzioni seguenti:

Console Elastic Beanstalk

- Spazio dei nomi: `aws:autoscaling:launchconfiguration`

  Nomi opzioni: `IamInstanceProfile`, `EC2KeyName`, `InstanceType`
- Spazio dei nomi: `aws:autoscaling:updatepolicy:rollingupdate`

  Nomi opzioni: `RollingUpdateType` e `RollingUpdateEnabled`
- Spazio dei nomi: `aws:elasticbeanstalk:application`

  Nomi opzioni: `Application Healthcheck URL`
- Spazio dei nomi: `aws:elasticbeanstalk:command`

  Nomi opzioni: `DeploymentPolicy`, `BatchSize` e `BatchSizeType`
- Spazio dei nomi: `aws:elasticbeanstalk:environment`

  Nomi opzioni: `ServiceRole`
- Spazio dei nomi: `aws:elasticbeanstalk:healthreporting:system`

  Nomi opzioni: `SystemType` e `HealthCheckSuccessThreshold`
- Spazio dei nomi: `aws:elasticbeanstalk:sns:topics`

  Nomi opzioni: `Notification Endpoint`
- Spazio dei nomi: `aws:elasticbeanstalk:sqsd`

  Nomi opzioni: `HttpConnections`
- Spazio dei nomi: `aws:elb:loadbalancer`

Nomi opzioni: `CrossZone`

- Spazio dei nomi: `aws:elb:policies`

  Nomi opzioni: `ConnectionDrainingTimeout` e `ConnectionDrainingEnabled`

CLI EB

- Spazio dei nomi: `aws:autoscaling:launchconfiguration`

  Nomi opzioni: `IamInstanceProfile`, `InstanceType`
- Spazio dei nomi: `aws:autoscaling:updatepolicy:rollingupdate`

  Nomi opzioni: `RollingUpdateType` e `RollingUpdateEnabled`
- Spazio dei nomi: `aws:elasticbeanstalk:command`

  Nomi opzioni: `BatchSize` e `BatchSizeType`
- Spazio dei nomi: `aws:elasticbeanstalk:environment`

  Nomi opzioni: `ServiceRole`
- Spazio dei nomi: `aws:elasticbeanstalk:healthreporting:system`

  Nomi opzioni: `SystemType`
- Spazio dei nomi: `aws:elb:loadbalancer`

  Nomi opzioni: `CrossZone`
- Spazio dei nomi: `aws:elb:policies`

  Nomi opzioni: `ConnectionDrainingEnabled`

Impostazione delle opzioni di configurazione prima della creazione di un ambiente

AWS Elastic Beanstalk supporta un gran numero di opzioni di configurazione che consentono di modificare le impostazioni applicate alle risorse dell'ambiente. Molte di queste opzioni dispongono di valori predefiniti che possono essere sostituiti per personalizzare l'ambiente. Possono essere configurate altre opzioni per abilitare funzionalità aggiuntive.

Elastic Beanstalk supporta due metodi di salvataggio delle impostazioni delle opzioni di configurazione. I file di configurazione in formato JSON o YAML possono essere inclusi nel codice sorgente della tua applicazione in una directory denominata `.ebextensions` e distribuiti come parte del tuo bundle di origine dell'applicazione. È possibile creare e gestire i file di configurazione in locale.

Le configurazioni salvate sono modelli creati da un ambiente in esecuzione o da un file di opzioni JSON e memorizzati in Elastic Beanstalk. Le configurazioni salvate esistenti possono anche essere estese per creare una nuova configurazione.

> **Note**
>
> Le impostazioni definite nei file di configurazione e le configurazioni salvate hanno una priorità inferiore rispetto alle impostazioni configurate durante o dopo la creazione dell'ambiente, inclusi i valori consigliati applicati dalla console Elastic Beanstalk e dalla CLI EB. Per informazioni dettagliate, consulta Priorità.

Le opzioni possono anche essere specificate in un documento in formato JSON e fornite direttamente a Elastic Beanstalk quando crei o aggiorni un ambiente con la CLI EB o AWS CLI. Le opzioni fornite direttamente a Elastic Beanstalk in questo modo sostituiscono tutti gli altri metodi.

Per un elenco completo delle opzioni disponibili, consulta Opzioni di configurazione.

Metodi

- File di configurazione (.ebextensions)
- Configurazioni salvate
- Documento JSON
- Configurazione CLI EB

File di configurazione (`.ebextensions`)

Utilizza `.ebextensions` per configurare le opzioni necessarie per far funzionare l'applicazione, e fornisci valori predefiniti per altre opzioni che possono essere sostituite a un livello più elevato di priorità. Le opzioni specificate in `.ebextensions` hanno il livello più basso di priorità e vengono sostituite dalle impostazioni in qualsiasi altro livello.

Per utilizzare i file di configurazione, crea una cartella chiamata `.ebextensions` al primo livello del codice sorgente del progetto. Aggiungi un file con l'estensione `.config` e specifica le opzioni nel modo seguente:

```
option_settings:
  - namespace:  namespace
    option_name:  option name
    value:  option value
  - namespace:  namespace
    option_name:  option name
    value:  option value
```

Ad esempio, il seguente file di configurazione imposta l'url di controllo dello stato dell'applicazione su `/health`:

`healthcheckurl.config`

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application
    option_name:  Application Healthcheck URL
    value:  /health
```

In JSON:

```
{
 "option_settings" :
    [
      {
        "namespace" : "aws:elasticbeanstalk:application",
        "option_name" : "Application Healthcheck URL",
        "value" : "/health"
      }
    ]
}
```

In questo modo viene configurato il sistema di bilanciamento del carico di Elastic Load Balancing nell'ambiente Elastic Beanstalk per effettuare una richiesta HTTP `/health` al percorso di ciascuna istanza EC2 per determinare se è integro o meno.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Includi la directory `.ebextensions` nel bundle di origine dell'applicazione e distribuiscila in un ambiente Elastic Beanstalk nuovo o esistente.

I file di configurazione supportano diverse sezioni, oltre a `option_settings` per la personalizzazione del software e dei file che vengono eseguiti sui server nel tuo ambiente. Per ulteriori informazioni, consultare .Ebextensions.

Configurazioni salvate

Crea una configurazione salvata per salvare le impostazioni applicate a un ambiente esistente durante o dopo la creazione dello stesso utilizzando la console Elastic Beanstalk, la CLI EB oppure l AWS CLI. Le configurazioni salvate appartengono a un'applicazione e possono essere applicate ad ambienti nuovi o esistenti per quell'applicazione.

Client

- Console Elastic Beanstalk
- CLI EB
- AWS CLI

Console Elastic Beanstalk

Per creare una configurazione salvata (console Elastic Beanstalk)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegli Actions (Operazioni), quindi Save configuration (Salva configurazione).
4. Utilizza la finestra di dialogo su schermo per completare l'operazione.

Le configurazioni salvate sono memorizzate nel bucket S3 di Elastic Beanstalk in una cartella che ha lo stesso nome dell'applicazione. Ad esempio, le configurazioni per un'applicazione denominata `my-app` nella regione us-west-2 per l'account numero 123456789012 sono disponibili in `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app`.

CLI EB

La CLI EB fornisce inoltre sottocomandi per l'interazione con le configurazioni salvate in eb config:

Per creare una configurazione salvata (CLI EB)

1. Salva la configurazione attuale dell'ambiente collegato:

```
~/project$ eb config save --cfg my-app-v1
```

   La CLI EB salva la configurazione su `~/project/.elasticbeanstalk/saved_configs/my-app-v1.cfg.yml`

2. Modifica la configurazione salvata in locale, se necessario.
3. Carica la configurazione salvata in S3:

```
~/project$ eb config put my-app-v1
```

AWS CLI

Creazione di una configurazione salvata da un ambiente di esecuzione con `aws elasticbeanstalk create-configuration-template`

Per creare una configurazione salvata (AWS CLI)

1. Individua l'ID ambiente dell'ambiente Elastic Beanstalk in uso con `describe-environments`:

```
$ aws elasticbeanstalk describe-environments --environment-name my-env
{
    "Environments": [
        {
            "ApplicationName": "my-env",
            "EnvironmentName": "my-env",
            "VersionLabel": "89df",
            "Status": "Ready",
```

```
            "Description": "Environment created from the EB CLI using \"eb create
\"",
            "EnvironmentId": "e-vcghmm2zwk",
            "EndpointURL": "awseb-e-v-AWSEBLoa-1JUM8159RA11M-43V6ZI1194.us-
west-2.elb.amazonaws.com",
            "SolutionStackName": "64bit Amazon Linux 2015.03 v2.0.2 running Multi-
container Docker 1.7.1 (Generic)",
            "CNAME": "my-env-nfptuqaper.elasticbeanstalk.com",
            "Health": "Green",
            "AbortableOperationInProgress": false,
            "Tier": {
                "Version": " ",
                "Type": "Standard",
                "Name": "WebServer"
            },
            "HealthStatus": "Ok",
            "DateUpdated": "2015-10-01T00:24:04.045Z",
            "DateCreated": "2015-09-30T23:27:55.768Z"
        }
    ]
}
```

2. Salva la configurazione attuale dell'ambiente con `create-configuration-template`:

```
$ aws elasticbeanstalk create-configuration-template --environment-id e-vcghmm2zwk
 --application-name my-app --template-name v1
```

Elastic Beanstalk salva la configurazione nel bucket di Elastic Beanstalk in Amazon S3.

Documento JSON

Se utilizzi la AWS CLI per creare e aggiornare ambienti, puoi anche fornire opzioni di configurazione in formato JSON. Una libreria di file di configurazione in JSON è utile se la utilizzi AWS CLI per creare e gestire ambienti.

Ad esempio, il seguente documento JSON imposta l'url di controllo dello stato dell'applicazione su `/health`:

~ /ebconfigs/healthcheckurl.json

```
[
  {
```

```
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
    "Value": "/health"
  }
]
```

Configurazione CLI EB

Oltre a supportare le configurazioni salvate e la configurazione diretta dell'ambiente con i comandi eb config, la CLI EB dispone di un file di configurazione con un'opzione denominata `default_ec2_keyname` che puoi utilizzare per specificare una coppia di chiavi Amazon EC2 per l'accesso SSH alle istanze nell'ambiente. La CLI EB si avvale di questa opzione per impostare l'opzione di configurazione `EC2KeyName` nel namespace `aws:autoscaling:launchconfiguration`.

~/ .yml workspace/my-app/.elasticbeanstalk/config

```
branch-defaults:
  master:
    environment: my-env
  develop:
    environment: my-env-dev
deploy:
  artifact: ROOT.war
global:
  application_name: my-app
  default_ec2_keyname: my-keypair
  default_platform: Tomcat 8 Java 8
  default_region: us-west-2
  profile: null
  sc: git
```

Impostazione delle opzioni di configurazione durante la creazione di un ambiente

Quando crei un AWS Elastic Beanstalk ambiente utilizzando la console Elastic Beanstalk, EB AWS CLI CLI, un SDK o l'API Elastic Beanstalk, puoi fornire valori per le opzioni di configurazione per personalizzare l'ambiente e le risorse che vengono lanciate al suo interno. AWS

Per tutto ciò di diverso da una modifica di configurazione una tantum, è possibile archiviare i file di configurazione in locale, nel bundle di origine o in Amazon S3.

Questo argomento include le procedure per tutti i metodi di impostazione delle opzioni di configurazione durante la creazione dell'ambiente.

Client

- Nella console Elastic Beanstalk
- Utilizzo della CLI EB
- Usando il AWS CLI

Nella console Elastic Beanstalk

Quando crei un ambiente Elastic Beanstalk nella console Elastic Beanstalk, è possibile fornire le opzioni di configurazione utilizzando i file di configurazione, le configurazioni salvate e i moduli nella procedura guidata Create New Environment (Crea nuovo ambiente).

Metodi

- Utilizzo dei file di configurazione (.ebextensions)
- Utilizzo di una configurazione salvata
- Utilizzo della procedura guidata per la creazione del nuovo ambiente

Utilizzo dei file di configurazione (`.ebextensions`)

Includi i file `.config` nel bundle di origine dell'applicazione in una cartella chiamata `.ebextensions`.

Per informazioni dettagliate sui file di configurazione, consulta .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Carica normalmente il bundle sorgente su Elastic Beanstalk durante la creazione dell'ambiente.

La console Elastic Beanstalk applica valori consigliati per alcune opzioni di configurazione e campi dei moduli per le altre. Le impostazioni configurate dalla console Elastic Beanstalk vengono applicate direttamente all'ambiente e sovrascrivono le impostazioni nei file di configurazione.

Utilizzo di una configurazione salvata

Quando crei un nuovo ambiente utilizzando la console Elastic Beanstalk, uno dei primi passaggi è scegliere una configurazione. La configurazione può essere una Predefined configuration (Configurazione predefinita), generalmente la versione più recente di una piattaforma, ad esempio PHP o Tomcat oppure una Saved configuration (Configurazione salvata).

Per applicare una configurazione salvata durante la creazione dell'ambiente (console Elastic Beanstalk)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Saved configurations (Configurazioni salvate).
4. Selezionare la configurazione salvata che si desidera applicare, quindi scegliere Launch environment (Avvia ambiente).
5. Procedi seguendo la procedura guidata per creare l'ambiente.

Le configurazioni salvate sono specifiche dell'applicazione. Consulta Configurazioni salvate per dettagli sulla creazione di configurazioni salvate.

Utilizzo della procedura guidata per la creazione del nuovo ambiente

La maggior parte delle opzioni di configurazione standard sono riportate nella pagina Configure more options (Configurazione di più opzioni) della procedura guidata Create New Environment (Crea un nuovo ambiente). Se crei un database Amazon RDS o configuri un VPC per l'ambiente in uso, le opzioni di configurazione aggiuntive sono disponibili per tali risorse.

Per impostare le opzioni di configurazione durante la creazione dell'ambiente (console Elastic Beanstalk)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applications (Applicazioni).
3. Seleziona o crea un'applicazione.
4. Selezionare Actions (Operazioni) e scegliere Create environment (Crea ambiente).
5. Continua con la procedura guidata e scegli Configure more options (Configura altre opzioni).

6. Scegliere una delle configuration presets (impostazioni di configurazione predefinite) e successivamente scegliere Edit (Modifica) in una o più categorie di configurazione per modificare un gruppo di opzioni di configurazione correlate.
7. Una volta selezionate le opzioni, scegli Create environment (Crea ambiente).

Le opzioni impostate nella procedura guidata del nuovo ambiente sono impostate direttamente sull'ambiente e sovrascrivono le impostazioni delle opzioni nelle configurazioni salvate o nei file di configurazione (`.ebextensions`) che applichi. Dopo la creazione dell'ambiente, è possibile annullare le impostazioni utilizzando la CLI EB o la AWS CLI per permettere la visualizzazione delle impostazioni nelle configurazioni salvate o nei file di configurazione.

Per ulteriori informazioni sulla procedura guidata per il nuovo ambiente, consultare Procedura guidata per la creazione del nuovo ambiente.

Utilizzo della CLI EB

Metodi

- Utilizzo dei file di configurazione (.ebextensions)
- Utilizzo delle configurazioni salvate
- Utilizzo delle opzioni della riga di comando

Utilizzo dei file di configurazione (`.ebextensions`)

Includi i file `.config` nella tua cartella di progetto in `.ebextensions` per distribuirli con il codice della tua applicazione.

Per informazioni dettagliate sui file di configurazione, consulta .Ebextensions.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   `-- config.yml
|-- index.php
`-- styles.css
```

Crea un ambiente e distribuisci il codice sorgente nello stesso con eb create.

```
~/workspace/my-app$ eb create my-env
```

Utilizzo delle configurazioni salvate

Per applicare una configurazione salvata al momento della creazione di un ambiente con eb create, utilizza l'opzione `--cfg`.

```
~/workspace/my-app$ eb create --cfg savedconfig
```

Puoi archiviare la configurazione salvata nella cartella del progetto o nella posizione di storage Elastic Beanstalk in Amazon S3. Nell'esempio precedente, la CLI EB prima cerca un file di configurazione salvato denominato `savedconfig.cfg.yml` nella cartella `.elasticbeanstalk/saved_configs/`. Non includere le estensioni dei nomi dei file (`.cfg.yml`) quando applichi una configurazione salvata con `--cfg`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   |-- saved_configs
|   |   `-- savedconfig.cfg.yml
|   `-- config.yml
|-- index.php
`-- styles.css
```

Se la CLI EB non trova la configurazione localmente, viene eseguita la ricerca nella posizione di storage Elastic Beanstalk in Amazon S3. Per informazioni dettagliate sulla creazione, sulla modifica e sul caricamento delle configurazioni salvate, consulta Configurazioni salvate.

Utilizzo delle opzioni della riga di comando

Il comando eb create della CLI EB presenta diverse opzioni che è possibile utilizzare per impostare le opzioni di configurazione durante la creazione dell'ambiente. Queste opzioni possono essere utilizzate per aggiungere un database RDS all'ambiente, configurare un VPC o sostituire i valori consigliati.

Ad esempio, la CLI EB utilizza il tipo di istanza `t2.micro` per impostazione predefinita. Per scegliere un tipo di istanza diverso, utilizza l'opzione `--instance_type`.

```
$ eb create my-env --instance_type t2.medium
```

Per creare un'istanza database Amazon RDS e collegarla all'ambiente, utilizza le opzioni `--database`.

```
$ eb create --database.engine postgres --database.username dbuser
```

Se ometti il nome dell'ambiente, la password del database o altri parametri necessari per creare l'ambiente, la CLI EB richiede di immetterli.

Vedi eb create per un elenco completo delle opzioni disponibili e degli esempi di utilizzo.

Usando il AWS CLI

Quando si utilizza il `create-environment` comando per creare un ambiente Elastic Beanstalk AWS CLI con AWS CLI , non viene applicato alcun valore consigliato. Tutte le opzioni di configurazione sono definite nei file di configurazione del bundle di origine da te specificati.

Metodi

- Utilizzo dei file di configurazione (.ebextensions)
- Utilizzo di una configurazione salvata
- Utilizzo delle opzioni della riga di comando

Utilizzo dei file di configurazione (**`.ebextensions`**)

Per applicare i file di configurazione a un ambiente creato con AWS CLI, includili nel bundle di sorgenti dell'applicazione che carichi su Amazon S3.

Per informazioni dettagliate sui file di configurazione, consulta .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Per caricare un bundle di sorgenti dell'applicazione e creare un ambiente con AWS CLI

1. Se non hai già un bucket Elastic Beanstalk in Amazon S3, creane uno con `create-storage-location`.

```
$ aws elasticbeanstalk create-storage-location
{
    "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Carica il bundle di origine dell'applicazione su Amazon S3.

```
$ aws s3 cp sourcebundle.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
sourcebundle.zip
```

3. Per creare la versione dell'applicazione.

```
$ aws elasticbeanstalk create-application-version --application-name my-app --
version-label v1 --description MyAppv1 --source-bundle S3Bucket="elasticbeanstalk-
us-west-2-123456789012",S3Key="my-app/sourcebundle.zip" --auto-create-application
```

4. Creazione dell'ambiente

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --version-label v1 --solution-stack-name "64bit Amazon Linux 2015.03
 v2.0.0 running Tomcat 8 Java 8"
```

Utilizzo di una configurazione salvata

Per applicare una configurazione salvata a un ambiente durante la creazione, utilizza il parametro `--template-name`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name
 my-env --template-name savedconfig --version-label v1
```

Quando specifichi una configurazione salvata, non devi anche specificare un nome dello stack di soluzioni. Le configurazioni salvate già specificano uno stack di soluzioni ed Elastic Beanstalk restituisce un errore se tenti di utilizzare entrambe le opzioni.

Utilizzo delle opzioni della riga di comando

Utilizza il parametro `--option-settings` per specificare le opzioni di configurazione in formato JSON.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name
 my-env --version-label v1 --template-name savedconfig --option-settings '[
  {
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
    "Value": "/health"
  }
]
```

Per caricare il formato JSON da un file, utilizza il prefisso `file://`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --version-label v1 --template-name savedconfig --option-settings file://
healthcheckurl.json
```

Elastic Beanstalk applica le impostazioni delle opzioni specificate con l'opzione `--option-settings` direttamente nell'ambiente. Se le stesse opzioni vengono specificate in una configurazione salvata o in un file di configurazione, `--option-settings` sostituisce tali valori.

Impostazione delle opzioni di configurazione dopo la creazione di un ambiente

Puoi modificare le impostazioni delle opzioni in un ambiente in esecuzione applicando le configurazioni salvate e caricando un nuovo bundle di origine con i file di configurazione (`.ebextensions`) o utilizzando un documento in formato JSON. Anche la console Elastic Beanstalk e la CLI EB hanno funzionalità specifiche del client per la configurazione e l'aggiornamento delle opzioni di configurazione.

Quando imposti o modifichi un'opzione di configurazione, puoi attivare un aggiornamento completo dell'ambiente, a seconda dell'importanza della modifica. Ad esempio, le modifiche delle opzioni in `aws:autoscaling:launchconfiguration`, ad esempio `InstanceType`, richiedono di nuovo l'esecuzione del provisioning delle istanze Amazon EC2 nell'ambiente. Questa operazione attiva un aggiornamento in sequenza. Altre modifiche di configurazione possono essere applicate senza alcuna interruzione o provisioning.

È possibile rimuovere le impostazioni delle opzioni da un ambiente con l'interfaccia a riga di comando o AWS CLI i comandi EB. La rimozione di un'opzione che è stata impostata direttamente in un ambiente a livello di API consente a impostazioni nei file di configurazione, che sono altrimenti

mascherate da impostazioni applicate direttamente a un ambiente, di emergere e di diventare effettive.

Le impostazioni delle configurazioni salvate e i file di configurazione possono essere sostituiti impostando la stessa opzione direttamente nell'ambiente con uno degli altri metodi di configurazione. Tuttavia, questi possono essere rimossi completamente applicando un file di configurazione o una configurazione salvata aggiornati. Quando un'opzione non è impostata in una configurazione salvata, in un file di configurazione oppure direttamente su un ambiente, viene applicato il valore predefinito, se presente. Per informazioni dettagliate, consulta Priorità.

Client

- Console Elastic Beanstalk
- CLI EB
- La AWS CLI

Console Elastic Beanstalk

Puoi aggiornare le impostazioni delle opzioni di configurazione nella console Elastic Beanstalk distribuendo un bundle di origine dell'applicazione che contiene i file di configurazione, applicando una configurazione salvata o modificando l'ambiente direttamente nella pagina Configuration (Configurazione) nella console di gestione dell'ambiente.

Metodi

- Utilizzo dei file di configurazione (.ebextensions)
- Utilizzo di una configurazione salvata
- Utilizzo della console Elastic Beanstalk

Utilizzo dei file di configurazione (`.ebextensions`)

Aggiorna i file di configurazione nella directory di origine, crea un nuovo bundle di origine e distribuisci la nuova versione nell'ambiente Elastic Beanstalk in uso per applicare le modifiche.

Per informazioni dettagliate sui file di configurazione, consulta .Ebextensions.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Le modifiche apportate ai file di configurazione non sostituiranno le impostazioni delle opzioni in configurazioni salvate o in impostazioni applicate direttamente all'ambiente a livello di API. Vedi Priorità per dettagli.

Utilizzo di una configurazione salvata

Applica una configurazione salvata a un ambiente in esecuzione per applicare impostazioni di opzioni da essa definite.

Per applicare una configurazione salvata a un ambiente in esecuzione (console Elastic Beanstalk)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Saved configurations (Configurazioni salvate).
4. Selezionare la configurazione salvata che si desidera applicare, quindi scegliere Load (Carica).
5. Seleziona un ambiente, quindi scegli Load (Carica).

Le impostazioni definite in una configurazione salvata sovrascrivono le impostazioni nei file di configurazione e vengono sostituite dalle impostazioni configurate tramite la console di gestione dell'ambiente.

Consulta Configurazioni salvate per dettagli sulla creazione di configurazioni salvate.

Utilizzo della console Elastic Beanstalk

La console Elastic Beanstalk presenta varie opzioni di configurazione nella pagina Configuration (Configurazione) per ogni ambiente.

Per modificare le opzioni di configurazione in un ambiente in esecuzione (console Elastic Beanstalk)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Trova la pagina della configurazione che desideri modificare:
   - Se vedi l'opzione a cui sei interessato o conosci la categoria di configurazione in cui si trova, seleziona Edit (Modifica) nella categoria di configurazione.
   - Per cercare un'opzione, attiva Table View (Visualizzazione tabella), quindi immetti i termini di ricerca nella casella di ricerca. Mentre digiti, l'elenco diventa più breve e mostra solo le opzioni che soddisfano i termini di ricerca.

     Quando vedi l'opzione che hai cercato, scegli Edit (Modifica) nella categoria di configurazione che la contiene.

5. Modifica le impostazioni e successivamente scegli Save (Salva).
6. Ripeti i due passaggi precedenti su altre categorie di configurazione in base alle esigenze.
7. Scegliere Apply (Applica).

Le modifiche apportate alle opzioni di configurazione nella console di gestione dell'ambiente vengono applicate direttamente all'ambiente. Queste modifiche sostituiscono le impostazioni per le stesse opzioni nei file di configurazione o nelle configurazioni salvate. Vedi Priorità per dettagli.

Per ulteriori informazioni su come modificare le opzioni di configurazione in un ambiente di esecuzione utilizzando la console Elastic Beanstalk, consulta gli argomenti disponibili in Configurazione degli ambienti Elastic Beanstalk.

CLI EB

È possibile aggiornare le impostazioni delle opzioni di configurazione con la CLI EB distribuendo un codice sorgente che contiene i file di configurazione, applicando le impostazioni di una configurazione salvata o modificando la configurazione dell'ambiente direttamente con il comando eb config.

Metodi

- Utilizzo dei file di configurazione (.ebextensions)
- Utilizzo di una configurazione salvata
- Utilizzo eb config
- Utilizzo eb setenv

Utilizzo dei file di configurazione (`.ebextensions`)

Includi i file `.config` nella tua cartella di progetto in `.ebextensions` per distribuirli con il codice della tua applicazione.

Per informazioni dettagliate sui file di configurazione, consulta .Ebextensions.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   `-- config.yml
|-- index.php
`-- styles.css
```

Distribuisci il tuo codice sorgente con eb deploy.

```
~/workspace/my-app$ eb deploy
```

Utilizzo di una configurazione salvata

È possibile utilizzare il comando eb config per applicare una configurazione salvata a un ambiente in esecuzione. Utilizza l'opzione `--cfg` opzione con il nome della configurazione salvata per applicare le impostazioni al tuo ambiente.

```
$ eb config --cfg v1
```

In questo esempio, `v1` è il nome di un file di configurazione precedentemente creato e salvato.

Le impostazioni applicate a un ambiente con questo comando sostituiranno le impostazioni applicate durante la creazione dell'ambiente e le impostazioni definite nei file di configurazione del tuo bundle di origine dell'applicazione.

Utilizzo eb config

Il comando eb config della CLI EB consente di impostare e rimuovere impostazioni delle opzioni direttamente su un ambiente utilizzando un editor di testo.

Quando esegui eb config, la CLI EB mostra le impostazioni applicate al tuo ambiente da tutte le origini, inclusi i file di configurazione, le configurazioni salvate, i valori consigliati, le opzioni impostate direttamente sull'ambiente e le API di default.

> **Note**
>
> eb config non visualizza le proprietà di ambiente. Per impostare proprietà di ambiente che puoi leggere dalla tua applicazione, utilizza eb setenv.

L'esempio seguente mostra le impostazioni applicate nello spazio del nome `aws:autoscaling:launchconfiguration`. Queste impostazioni includono:

- Due valori consigliati, per `IamInstanceProfile` e `InstanceType`, applicati dalla CLI EB durante la creazione dell'ambiente.
- L'opzione `EC2KeyName`, impostata direttamente nell'ambiente durante la creazione in base alla configurazione del repository.
- Valori predefiniti dell'API per le altre opzioni.

```
ApplicationName: tomcat
```

```
DateUpdated: 2015-09-30 22:51:07+00:00
EnvironmentName: tomcat
SolutionStackName: 64bit Amazon Linux 2015.03 v2.0.1 running Tomcat 8 Java 8
settings:
...
aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: null
    EC2KeyName: my-key
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    ImageId: ami-1f316660
    InstanceType: t2.micro
...
```

Per impostare o modificare le opzioni di configurazione con eb config

1. Esegui eb config per visualizzare la configurazione dell'ambiente.

   ```
   ~/workspace/my-app/$ eb config
   ```

2. Modifica uno qualsiasi dei valori delle impostazioni utilizzando l'editor di testo predefinito.

   ```
   aws:autoscaling:launchconfiguration:
       BlockDeviceMappings: null
       EC2KeyName: my-key
       IamInstanceProfile: aws-elasticbeanstalk-ec2-role
       ImageId: ami-1f316660
       InstanceType: t2.medium
   ```

3. Salva il file di configurazione temporaneo ed esci.
4. La CLI EB aggiorna la configurazione dell'ambiente.

Impostando le opzioni di configurazione con eb config vengono sovrascritte le impostazioni di tutte le altre sorgenti.

Puoi anche eliminare le opzioni dal tuo ambiente con eb config.

Per rimuovere le opzioni di configurazione (CLI EB)

1. Esegui eb config per visualizzare la configurazione dell'ambiente.

   ```
   ~/workspace/my-app/$ eb config
   ```

2. Sostituisci qualsiasi valore mostrato con la stringa `null`. È anche possibile eliminare l'intera riga contenente l'opzione che desideri rimuovere.

```
aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: null
    EC2KeyName: my-key
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    ImageId: ami-1f316660
    InstanceType: null
```

3. Salva il file di configurazione temporaneo ed esci.
4. La CLI EB aggiorna la configurazione dell'ambiente.

Se elimini le opzioni dal tuo ambiente con eb config le impostazioni per le stesse opzioni potranno essere visibili dai file di configurazione nel tuo bundle di origine dell'applicazione. Vedi Priorità per dettagli.

Utilizzo eb setenv

Per impostare le proprietà dell'ambiente con la CLI EB, utilizza eb setenv.

```
~/workspace/my-app/$ eb setenv ENVVAR=TEST
INFO: Environment update is starting.
INFO: Updating environment my-env's configuration settings.
INFO: Environment health has transitioned from Ok to Info. Command is executing on all
 instances.
INFO: Successfully deployed new configuration to environment.
```

Questo comando imposta le proprietà dell'ambiente nel namespace `aws:elasticbeanstalk:application:environment`. Le proprietà dell'ambiente impostate con eb setenv sono disponibili per la tua applicazione, dopo un breve processo di aggiornamento.

Visualizza le proprietà dell'ambiente impostate nel tuo ambiente con eb printenv.

```
~/workspace/my-app/$ eb printenv
 Environment Variables:
     ENVVAR = TEST
```

La AWS CLI

È possibile aggiornare le impostazioni delle opzioni di configurazione con il AWS CLI distribuendo un pacchetto sorgente che contiene i file di configurazione, applicando una configurazione salvata memorizzata in remoto o modificando l'ambiente direttamente con il comando. `aws elasticbeanstalk update-environment`

Metodi

- Utilizzo dei file di configurazione (.ebextensions)
- Utilizzo di una configurazione salvata
- Utilizzo delle opzioni della riga di comando

Utilizzo dei file di configurazione (`.ebextensions`)

Per applicare i file di configurazione a un ambiente in esecuzione con AWS CLI, includili nel pacchetto sorgente dell'applicazione che carichi su Amazon S3.

Per informazioni dettagliate sui file di configurazione, consulta .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Per caricare un bundle fonte dell'applicazione e applicarlo a un ambiente in esecuzione (AWS CLI)

1. Se non hai già un bucket Elastic Beanstalk in Amazon S3, creane uno con `create-storage-location`:

   ```
   $ aws elasticbeanstalk create-storage-location
   {
       "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
   }
   ```

2. Carica il bundle di origine dell'applicazione su Amazon S3.

```
$ aws s3 cp sourcebundlev2.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
sourcebundlev2.zip
```

3. Per creare la versione dell'applicazione.

```
$ aws elasticbeanstalk create-application-version --application-
name my-app --version-label v2 --description MyAppv2 --source-bundle
 S3Bucket="elasticbeanstalk-us-west-2-123456789012",S3Key="my-app/
sourcebundlev2.zip"
```

4. Aggiornamento dell'ambiente.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --version-label
 v2
```

Utilizzo di una configurazione salvata

Puoi applicare una configurazione salvata a un ambiente in esecuzione con l'opzione `--template-name` nel comando `aws elasticbeanstalk update-environment`.

La configurazione salvata deve essere nel bucket Elastic Beanstalk in un percorso denominato in base al nome dell'applicazione in uso in `resources/templates`. Ad esempio, il modello `v1` per l'applicazione `my-app` nella regione Stati Uniti occidentali (Oregon). (us-west-2) per l'account 123456789012 si trova in `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/v1`.

Per applicare una configurazione salvata a un ambiente in esecuzione (AWS CLI)

- Specifica la configurazione salvata in una chiamata `update-environment` con l'opzione `--template-name`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --template-
name v1
```

Elastic Beanstalk colloca le configurazioni salvate in questa posizione quando le crei con `aws elasticbeanstalk create-configuration-template`. Puoi anche modificare le configurazioni salvate in locale e inserirle in questa posizione.

Utilizzo delle opzioni della riga di comando

Per modificare le opzioni di configurazione con un documento JSON (AWS CLI)

1. Definisci le tue impostazioni delle opzioni in formato JSON in un file locale.
2. Esegui `update-environment` con l'opzione `--option-settings`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://~/ebconfigs/as-zero.json
```

In questo esempio, `as-zero.json` definisce le opzioni che configurano l'ambiente con un minimo e un massimo di istanze pari a zero. Questo arresta le istanze nell'ambiente senza terminare l'ambiente.

`~/ebconfigs/as-zero.json`

```
[
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MinSize",
        "Value": "0"
    },
    {
        "Namespace": "aws:autoscaling:asg",
        "OptionName": "MaxSize",
        "Value": "0"
    },
    {
        "Namespace": "aws:autoscaling:updatepolicy:rollingupdate",
        "OptionName": "RollingUpdateEnabled",
        "Value": "false"
    }
]
```

> **Note**
>
> Impostando le opzioni di configurazione con `update-environment` vengono sovrascritte le impostazioni di tutte le altre sorgenti.

Puoi anche eliminare le opzioni dal tuo ambiente con `update-environment`.

Per annullare le opzioni di configurazione (AWS CLI)

- Esegui il comando `update-environment` con l'opzione `--options-to-remove`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --options-to-
remove Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType
```

Se elimini le opzioni dal tuo ambiente con `update-environment` le impostazioni per le stesse opzioni potranno essere visibili dai file di configurazione nel tuo bundle di origine dell'applicazione. Se un'opzione non è configurata utilizzando uno di questi metodi, si applica il valore predefinito dell'API, se esiste. Vedi Priorità per dettagli.

Opzioni generali per tutti gli ambienti

Spazi dei nomi

- aws:autoscaling:asg
- aws:autoscaling:launchconfiguration
- aws:autoscaling:scheduledaction
- aws:autoscaling:trigger
- aws:autoscaling:updatepolicy:rollingupdate
- aws:ec2:instances
- aws:ec2:vpc
- aws:elasticbeanstalk:application
- aws:elasticbeanstalk:application:environment
- aws: elasticbeanstalk:application:environmentsecrets
- aws:elasticbeanstalk:cloudwatch:logs
- aws:elasticbeanstalk:cloudwatch:logs:health
- aws:elasticbeanstalk:command
- aws:elasticbeanstalk:environment
- aws:elasticbeanstalk:environment:process:default
- aws:elasticbeanstalk:environment:process:process_name
- aws:elasticbeanstalk:environment:proxy:staticfiles

- aws:elasticbeanstalk:healthreporting:system
- aws:elasticbeanstalk:hostmanager
- aws:elasticbeanstalk:managedactions
- aws:elasticbeanstalk:managedactions:platformupdate
- aws:elasticbeanstalk:monitoring
- aws:elasticbeanstalk:sns:topics
- aws:elasticbeanstalk:sqsd
- aws:elasticbeanstalk:trafficsplitting
- aws:elasticbeanstalk:xray
- aws:elb:healthcheck
- aws:elb:loadbalancer
- aws:elb:listener
- aws:elb:listener:listener_port
- aws:elb:policies
- aws:elb:policies:policy_name
- aws:elbv2:listener:default
- aws:elbv2:listener:listener_port
- aws:elbv2:listenerrule:rule_name
- aws:elbv2:loadbalancer
- aws:rds:dbinstance

aws:autoscaling:asg

Configura il gruppo Auto Scaling dell'ambiente. Per ulteriori informazioni, consulta the section called "Gruppo Auto Scaling".

Spazio dei nomi: `aws:autoscaling:asg`

Nome	Descrizione	Impostazione predefinita	Valori validi
Availability Zones	Le zone di disponibilità (AZ) sono ubicazioni distinte all'interno di una AWS regione progettate per essere	`Any`	`Any` `Any 1`

Nome	Descrizione	Impostazione predefinita	Valori validi
	isolate dai guasti in altre AZ. Offrono una connettività di rete non costosa e a bassa latenza ad altre zone di disponibilità nella stessa regione. Scegli il numero di zone di disponibilità per le istanze.		`Any 2` `Any 3`
Cooldown	I periodi di attesa impediscono ad Amazon EC2 Auto Scaling di avviare ulteriori attività di dimensionamento prima che gli effetti delle attività precedenti siano visibili. La quantità di tempo, in secondi, tra il completamento di un'attività di dimensionamento e l'inizio di un'altra attività di questo tipo.	`360`	`0` Da a `10000`

Custom Availability Zones	Definisci le zone di disponibilità per le istanze.	Nessuno	`us-east-1a`
			`us-east-1b`
			`us-east-1c`
			`us-east-1d`
			`us-east-1e`
			`eu-central-1`

Nome	Descrizione	Impostazione predefinita	Valori validi

EnableCapacityRebalancing	Specifica se abilitare la caratteristica di ribilanciamento della capacità per le istanze Spot nel gruppo Auto Scaling. Per ulteriori informazioni, consulta Ribilanciamento della capacità nella Guida per l'utente di Amazon EC2 Auto Scaling.
	
	Questa opzione è rilevante solo se `EnableSpot` è impostato su `true` nello spazio dei nomi `aws:ec2:instances` e c'è almeno un'istanza Spot nel gruppo Auto Scaling.
		`false`	`true`
			
			`false`

MinSize	Il numero minimo di istanze desiderate nel gruppo Auto Scaling.	1	1 Da a `10000`
MaxSize	Il numero massimo di istanze desiderate nel gruppo Auto Scaling.	4	1 Da a `10000`

aws:autoscaling:launchconfiguration

Configura le istanze Amazon Elastic Compute Cloud (Amazon EC2) dell'ambiente.

Le istanze utilizzate per l'ambiente vengono create utilizzando un modello di avvio Amazon EC2 o una risorsa di configurazione di avvio del gruppo Auto Scaling. Le opzioni seguenti funzionano con entrambi questi tipi di risorse.

Per ulteriori informazioni, consulta the section called “ EC2 Istanze Amazon”. Puoi anche fare riferimento a ulteriori informazioni su Amazon Elastic Block Store (EBS) nel capitolo Amazon EBS della Amazon EC2 User Guide.

Spazio dei nomi: aws:autoscaling:launchconfiguration

Nome	Descrizione	Impostazione predefinita	Valori validi

DisableDefaultEC2SecurityGroup	Se impostato sul valore predefinito di`false`, Elastic Beanstalk crea un gruppo di sicurezza predefinito che consente il traffico da Internet o dal sistema di bilanciamento del carico sulle porte standard per HTTP (80). Collega questo gruppo di sicurezza alle istanze EC2 dell'ambiente quando crea l'ambiente.
	
	Se impostato su `true` Elastic Beanstalk, non assegnerà il gruppo di sicurezza predefinito alle istanze EC2 per un nuovo ambiente. Per un ambiente esistente, Elastic Beanstalk annullerà l'assegnazione del gruppo di sicurezza EC2 predefinito dalle istanze EC2 dell'ambiente. Di conseguenza, è necessario impostare anche le seguenti configurazioni:
	
	• L'`SecurityGroups` opzione di questo namespace richiederà almeno un valore per definire i gruppi di sicurezza personalizzati.
	• Per gli ambienti con un sistema di bilanciamento del carico, sarà inoltre necessario impostare le `SecurityGroups` opzioni in un altro namespace per configurare gruppi di sicurezza personalizzati per il sistema di bilanciamento del carico. Per i sistemi di bilanciam
		`false`	`true`
			
			`false`

Nome	Descrizione	Impostazione predefinita	Valori validi
	ento del carico delle applicazioni, imposta l'opzione nel namespace. aws:elbv2:loadbalancer Per i sistemi di bilanciamento del carico classici, impostate l'opzione nel namespace. aws:elb:loadbalancer • Per ulteriori informazioni, consulta Gestione dei gruppi di sicurezza EC2. Se viene specificato un valore per EC2KeyName un ambiente `DisableDefaultEC2SecurityGroup` impostato su un gruppo di sicurezza predefinito non verrà associato alle `true` istanze EC2.		

Nome	Descrizione	Impostazione predefinita	Valori validi

DisableIMDSv1	Impostato per `true` disabilitare Instance Metadata Service Version 1 (IMDSv1) e applicare IMDSv2.
	
	Impostato per abilitare sia IMDSv1 che IMDSv2. `false`
	
	Le istanze per l'ambiente sono predefinite come segue, in base al sistema operativo della piattaforma:
	
	• Windows server, AL2 e versioni precedenti: abilita sia IMDSv1 che IMDSv2 (impostazione predefinita su) DisableIMDSv1 `false`
	• AL2023: abilita solo IMDSv2 (impostazione predefinita su) DisableIMDSv1 `true`
	
	Per ulteriori informazioni, consulta Configurazione del servizio di metadati dell'istanza.
	
	Important
	Questa impostazione delle opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessari e le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono
		`false`: piattaforme basate su server Windows, Amazon Linux 2 e versioni precedenti
		
		`true`: piattaforme basate su Amazon Linux 2023
			`true`
			
			`false`

Nome	Descrizione	Impostazione predefinita	Valori validi
	incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornam enti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambi ente. Per ulteriori informazi oni e altre considerazioni, consultaMigrazione dell'ambi ente Elastic Beanstalk ai modelli di avvio.		

Nome	Descrizione	Impostazione predefinita	Valori validi
EC2KeyName	È possibile utilizzare una coppia di chiavi per accedere in modo sicuro all'istanza EC2. Se viene specificato un valore per `EC2KeyName` un ambiente `DisableDefaultEC2SecurityGroup` impostato su `true` un gruppo di sicurezza predefinito, non verrà associato alle istanze EC2. **Note** Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console sostituisce questa opzione con un valore consigliato.	Nessuno	

Nome	Descrizione	Impostazione predefinita	Valori validi
IamInstanceProfile	Un profilo di istanza consente agli utenti e ai AWS servizi AWS Identity and Access Management (IAM) di accedere a credenziali di sicurezza temporanee per effettuare AWS chiamate API. Specifica il nome del profilo dell'istanza o il suo ARN. Esempi: • `aws-elasticbeanstalk-ec2-role` • `arn:aws:iam::123456789012:instance-profile/aws-elasticbeanstalk-ec2-role` **Note** Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.	Nessuna	Nome del profilo dell'istanza o ARN.
ImageId	È possibile sostituire l'Amazon Machine Image (AMI) di default con il tuo ID AMI. Ad esempio: `ami-1f316660`	Nessuno	

Nome	Descrizione	Impostazione predefinita	Valori validi
InstanceType	Il tipo di istanza utilizzato per eseguire l'applicazione in un ambiente Elastic Beanstalk. **Important** L'opzione `InstanceType` è obsoleta. È sostituita dall'opzione `InstanceTypes` più nuova e potente nello spazio dei nomi `aws:ec2:instances`. È possibile utilizzare la nuova opzione per specificare un elenco di uno o più tipi di istanza per l'ambiente. Il primo valore dell'elenco è equivalente al valore dell'opzione `InstanceType` inclusa nello spazio dei nomi `aws:autoscaling:launchconfiguration` descritto qui. Si consiglia di utilizzare la nuova opzione per specificare i tipi di istanza. Se specificata, la nuova opzione ha la precedenza su quella precedente. Per ulteriori informazioni, consulta the section called "Spazio dei nomi aws:ec2:instances". I tipi di istanza disponibili dipendono dalle zone di disponibilità e dalla	Varia in base all'account e alla regione.	Un tipo di istanza EC2. Varia in base all'account, alla regione e alla zona di disponibilità. È possibile ottenere un elenco di tipi di istanza Amazon EC2 filtrati in base a questi valori. Per ulteriori informazioni, consulta la sezione Tipi di istanze disponibili nella Guida per l'utente di Amazon EC2.

Nome	Descrizione	Impostazione predefinita	Valori validi
	regione utilizzate. Se si sceglie una sottorete, la zona di disponibilità che contiene tale sottorete determina i tipi di istanza disponibili. • Elastic Beanstalk non supporta i tipi di istanza Amazon EC2 Mac. • Per ulteriori informazioni sulle famiglie e i tipi di istanze di Amazon EC2, consulta Tipi di istanze nella Amazon EC2 User Guide. • Per ulteriori informazioni sui tipi di istanze disponibili nelle diverse regioni, consulta la sezione Tipi di istanze disponibili nella Amazon EC2 User Guide. **Note** Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.		

Nome	Descrizione	Impostazione predefinita	Valori validi

LaunchTemplateTagPropagationEnabled	Imposta su `true` per abilitare la propagazione di tag dell'ambiente ai modelli di avvio per risorse specifiche fornite all'ambiente.
	
	Elastic Beanstalk è in grado di propagare i tag ai modelli di avvio solo per le risorse seguenti:
	
	• Volumi EBS
	• Istanze EC2
	• Interfacce di rete EC2
	• CloudFormation modelli di avvio che definiscono una risorsa
	
	Questo vincolo esiste perché consente CloudFormation solo l'uso di tag per la creazione di modelli per risorse specifiche. Per ulteriori informazioni, consulta la TagSpecificationGuida per l'AWS CloudFormation utente.
	
	Important
	
	• La modifica del valore di questa opzione da `false` a `true` per un ambiente esistente può costituire una modifica rivoluzionaria per i tag esistenti.
	• Quando questa funzionalità è abilitata, la propagazione di tag richiederà la
		`false`	`true`
			
			`false`

Nome	Descrizione	Impostazione predefinita	Valori validi
	sostituzione di EC2, con conseguenti tempi di inattivit à. Per applicare le modifiche alla configurazione in batch e prevenire i tempi di inattività durante il processo di aggiornamento, puoi abilitare gli aggiornamenti in sequenza. Per ulteriori informazioni, consulta Modifiche di configurazione. Per ulteriori informazioni sui modelli di avvio, consulta le risorse seguenti: • Modelli di avvio nella Guida per l'utente di Dimensionamento automatico Amazon EC2 • Utilizzo dei modelli nella Guida per l'utente di AWS CloudFormation • Frammenti di modello Elastic Beanstalk nella Guida per l'utente di AWS CloudFormation Per ulteriori informazioni su questa opzione, consulta Propagazione di tag ai modelli di avvio.		

MonitoringInterval	L'intervallo (in minuti) a cui desideri che vengano restituiti i CloudWatch parametri di Amazon.	`5 minute`	`1 minute`
			
			`5 minute`

Nome	Descrizione	Impostazione predefinita	Valori validi
SecurityGroups	Elenca gli ID dei gruppi di sicurezza Amazon EC2 da assegnare alle istanze EC2 nel gruppo Auto Scaling per definire le regole del firewall per le istanze. Utilizza questa opzione insieme `DisableDefaultEC2SecurityGroup` a per collegare i tuoi gruppi di sicurezza personalizzati che definiscono le regole del firewall per le istanze EC2. Per ulteriori informazioni, consulta Ambienti con carico bilanciato (multiistanza). **Important** Potrebbe essere necessario completare alcune configurazioni aggiuntive per evitare che il traffico in entrata verso le istanze EC2 venga bloccato. Ciò si applica solo agli ambienti a più istanze con gruppi di sicurezza EC2 personalizzati. I gruppi di sicurezza EC2 devono includere una regola in entrata che garantisca l'accesso al traffico instradato dal sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Gestione	`elasticbeanstalk-default`	

Nome	Descrizione	Impostazione predefinita	Valori validi
	dei gruppi di sicurezza EC2 in ambienti a più istanze. Puoi fornire una singola stringa di valori separati da virgole che contengono gli ID dei gruppi di sicurezza Amazon EC2 esistenti o riferimenti alle risorse AWS: :EC2:: create nel modello. SecurityGroup È necessario fornire almeno un valore per questa opzione se `DisableDefaultEC2SecurityGroup` for questo namespace è impostato su. `true`		

Nome	Descrizione	Impostazione predefinita	Valori validi

SSHSource Restriction	Consente di bloccare l'accesso SSH a un ambiente. Ad esempio, è possibile bloccare l'accesso SSH alle istanze EC2, di modo che solo un host bastione sia in grado di accedere alle istanze nella sottorete privata.
	
	Questa stringa è espressa nella seguente forma:
	
	`protocol, fromPort, toPort, source_restriction`
	
	`protocol`
	
	Il protocollo per la regola in ingresso.
	`fromPort`
	
	Il numero di porta iniziale.
	`toPort`
	
	Il numero di porta finale.
	`source_restriction`
	
	L'intervallo Classless Inter-Domain Routing (CIDR) o il gruppo di sicurezza attraverso il quale il traffico deve essere instradato. Specificare il gruppo di sicurezza con l'ID del gruppo di sicurezza.
	
	Per specificare un gruppo di sicurezza di un altro account, includi l'ID dell' AWS account prima
		Nessuno	

Nome	Descrizione	Impostazione predefinita	Valori validi
	dell'ID del gruppo di sicurezza , separato da una barra. L'altro account deve trovarsi nella stessa AWS regione. Nota la sintassi: *aws-account-id* /*security-group-id* . Ad esempio: **123456789012 /sg-99999999** Esempi: • `tcp, 22, 22, 54.240.196.185/32` • `tcp, 22, 22, my-security-group-id` • `tcp, 22, 22, 123456789012/their-security-group-id`		

Nome	Descrizione	Impostazione predefinita	Valori validi
BlockDeviceMappings	Collega ulteriori volumi Amazon EBS o volumi di instance store su tutte le istanze del gruppo Auto Scaling. **Important** Questa impostazione delle opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessari e le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio. Quando si esegue la mappatura dei volumi di instance store, è necessario mappare solo il nome del dispositivo a un nome di volume. Tuttavia, quando	Nessuno	• dimensione — deve essere compresa tra 500 e 16384 GiB • velocità effettiva: deve essere compresa tra 125 e 1000 mebibyte al secondo () MiB/s

Nome	Descrizione	Impostazione predefinita	Valori validi
	si mappano volumi Amazon EBS, si consiglia di specificare alcuni o tutti i seguenti campi (ogni campo deve essere separato da due punti): • ID snapshot • dimensioni, in GB • eliminare al termine (`true` o `false`) • tipo di archiviazione (solo per `gp3`, `gp2`, `standard`, `st1`, `sc1` o `io1`) • IOPS (solo per `gp3` o `io1`) • velocità di trasmissione effettiva (solo per `gp3`) L'esempio seguente collega tre volumi Amazon EBS, un volume gp2 vuoto di 100 GB e uno snapshot, uno volume io1 vuoto di 20 GB con 2.000 provision ed IOPS e un volume di instance store `ephemeral0` . Più volumi di instance store possono essere collegati se il tipo di istanza lo supporta. `/dev/sdj=:100:true:gp2,/dev/sdh=snap-51eef269,/dev/sdi=:20:true:io1:2000,/dev/sdb=ephemeral0`		

Nome	Descrizione	Impostazione predefinita	Valori validi
RootVolumeType	Tipo di volume (magnetico, general purpose SSD o provisioned IOPS SSD) da utilizzare per il volume Amazon EBS root collegato alle istanze EC2 dell'ambiente. **Important** Questa impostazione delle opzioni può far sì che Elastic Beanstalk migri un ambiente esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessari e le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consulta Migrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.	Varia in base alla piattaforma.	`standard` per l'archiviazione magnetica. `gp2` o `gp3` per SSD per scopo generico. `io1` per SSD con capacità di IOPS allocata.

Nome	Descrizione	Impostazione predefinita	Valori validi
RootVolumeSize	La capacità di storage del volume Amazon EBS root in GB. Obbligatorio se imposti `RootVolumeType` su provisioned IOPS SSD. Ad esempio, `"64"`.	Varia a seconda della piattaforma per lo storage magnetico e general purpose SSD. Nessuno per provisioned IOPS SSD.	Da `10` a `16384` GB per general purpose SSD e provisioned IOPS SSD. Da `8` a `1024` GB per storage magnetico.
RootVolumeIOPS	Le input/output operazioni al secondo (IOPS) desiderate per un volume root SSD IOPS fornito o per un volume radice SSD generico. `gp3` La proporzione massima tra IOPS e dimensione del volume è 500 a 1. Ad esempio, un volume con 3.000 IOPS deve avere almeno 6 GiB.	Nessuno	Da `100` a `20000` per volumi root SSD io1 con capacità di IOPS allocata. Da `3000` a `16000` per volumi root SSD `gp3` per scopo generico.
RootVolumeThroughput	Il throughput desiderato di mebibyte al secondo (MiB/s) per il provisioning del volume root di Amazon EBS collegato all'istanza EC2 del tuo ambiente. **Note** Questa opzione è applicabile solo ai tipi di archiviazione `gp3`.	Nessuna	`125` Da a `1000`

aws:autoscaling:scheduledaction

Configura operazioni programmate per il gruppo Auto Scaling dell'ambiente. Per ogni azione, specifica un `resource_name` oltre a nome di opzione, namespace e valore per ogni impostazione. Per esempi, consulta Spazio dei nomi aws:autoscaling:scheduledaction.

`Spazio dei` nomi: aws:autoscaling:scheduledaction

Nome	Descrizione	Impostazione predefinita	Valori validi
StartTime	Per le azioni singole, scegli la data e l'ora di esecuzione. Per le azioni ricorrenti, scegli quando attivare l'azione.	Nessuno	Un timestamp ISO-8601 unico per tutte le azioni di ridimensionamento pianificate.
EndTime	Una data e un'ora future (nel fuso UTC/GMT orario) in cui si desidera che l'azione di ridimensionamento pianificata smetta di ripetersi. Se non si specifica un EndTime, l'azione viene ripetuta in base all'espressione `Recurrence` Ad esempio: `2015-04-28T04:07:2Z` Quando un'operazione pianificata termina, le impostazioni precedenti di Amazon EC2 Auto Scaling non vengono ripristinate automaticamente. Configura una seconda azione programmata per ripristinare le impostazioni originali come necessario.	Nessuno	Un ISO-8601 timestamp unico per tutte le azioni di ridimensionamento pianificate.

Nome	Descrizione	Impostazione predefinita	Valori validi
MaxSize	Il numero massimo di istanze da applicare all'esecuzione dell'azione.	Nessuna	`0` Da a `10000`
MinSize	Il numero minimo di istanze da applicare all'esecuzione dell'azione.	Nessuna	`0` Da a `10000`
DesiredCapacity	Imposta la capacità iniziale desiderata per il gruppo Auto Scaling. Dopo l'applicazione dell'azione programmata, i trigger regolano la capacità desiderata in base alle relative impostazioni.	Nessuna	`0` Da a `10000`
Recurrence	La frequenza alla quale si desidera eseguire l'azione pianificata. Se non si specifica una ricorrenza, l'azione di ridimensionamento viene eseguita una sola volta, come specificato da `StartTime` .	Nessuna	Un'espressione Cron.
Suspend	Imposta su `true` per disattivare temporaneamente un'azione programmata ricorrente.	`false`	`true` `false`

aws:autoscaling:trigger

Configura trigger di dimensionamento per il gruppo Auto Scaling dell'ambiente.

> **Note**
>
> Tre opzioni in questo spazio dei nomi determinano per quanto tempo il parametro per un trigger può restare oltre i suoi limiti definiti prima dell'attivazione del trigger. Queste opzioni sono correlate come segue:
> `BreachDuration = Period * EvaluationPeriods`

I valori predefiniti per queste opzioni (5, 5 e 1, rispettivamente) soddisfano questa equazione. Se specifichi valori incoerenti, Elastic Beanstalk potrebbe modificare uno dei valori in modo che l'equazione risulti ancora soddisfatta.

Spazio dei nomi: aws:autoscaling:trigger

Nome	Descrizione	Impostazione predefinita	Valori validi
BreachDuration	Il periodo di tempo, in minuti, durante il quale un parametro può superare il relativo limite definito (come specificato in `UpperThreshold` e `LowerThreshold`) prima che il trigger venga richiamato.	`5`	`1` Da a `600`
LowerBreachScaleIncrement	Il numero di istanze Amazon EC2 da rimuovere durante l'esecuzione di un'attività di dimensionamento.	`-1`	
LowerThreshold	Se la misurazione è inferiore alla durata dell'utilizzo fuori limite impostata, il trigger viene richiamato.	`2000000`	`0` Da a `20000000`

MeasureName	Il parametro utilizzato per il trigger di Auto Scaling.
	
	Note
	`HealthyHostCount` , `UnhealthyHostCount` e `TargetResponseTime` sono applicabili solo per ambienti con un sistema di bilanciamento del carico dedicato. Questi valori
		`NetworkOut`	`CPUUtilization`
			
			`NetworkIn`
			
			`NetworkOut`
			
			`DiskWriteOps`
			
			`DiskReadBytes`
			
			`DiskReadOps`

Nome	Descrizione	Impostazione predefinita	Valori validi

	non sono validi per gli ambienti configurati con un servizio di bilanciamento del carico condiviso. Per ulteriori informazioni sui tipi di bilanciamento del carico, consulta Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk.		`DiskWriteBytes`
			`Latency`
			`RequestCount`
			`HealthyHostCount`
			`UnhealthyHostCount`
			`TargetResponseTime`

Period	Specifica la frequenza con cui Amazon CloudWatch misura le metriche per il tuo trigger. Il valore è il numero di minuti tra due periodi consecutivi.	`5`	`1` Da a `600`
EvaluationPeriods	Il numero di periodi di valutazione consecutivi utilizzati per determinare se è in corso una violazione.	`1`	`1` Da a `600`

Statistic	La statistica utilizzata dal trigger, ad esempio `Average`.	`Average`	`Minimum`
			`Maximum`
			`Sum`
			`Average`

Nome	Descrizione	Impostazione predefinita	Valori validi

Unit	L'unità per la misura del trigger, ad esempio `Bytes`.	`Bytes`	`Seconds`
			`Percent`
			`Bytes`
			`Bits`
			`Count`
			`Bytes/Second`
			`Bits/Second`
			`Count/Second`
			`None`

UpperBreachScaleIncrement	Specifica il numero di istanze Amazon EC2 da aggiungere durante l'esecuzione di un'attività di ridimensionamento.	1	
UpperThreshold	Se la misurazione è superiore alla durata dell'utilizzo fuori limite impostata, il trigger viene richiamato.	`6000000`	`0` Da a `20000000`

aws:autoscaling:updatepolicy:rollingupdate

Configura gli aggiornamenti in sequenza del gruppo Auto Scaling dell'ambiente.

Spazio dei nomi: `aws:autoscaling:updatepolicy:rollingupdate`

Nome	Descrizione	Impostazione predefinita	Valori validi
MaxBatchSize	Il numero di istanze incluse in ogni batch dell'aggiornamento in sequenza.	One-third della dimensione minima del gruppo Auto Scaling, arrotondata al numero intero successivo più alto.	`1` Da a `10000`
MinInstancesInService	Il numero minimo di istanze che devono essere in servizio nel gruppo Auto Scaling mentre altre istanze vengono terminate.	La dimensione minima del gruppo Auto Scaling o una dimensione inferiore alla dimensione massima del gruppo Auto Scaling, a seconda di quale sia più bassa.	`0` Da a `9999`
RollingUpdateEnabled	Se `true`, abilita gli aggiornamenti in sequenza per un ambiente. Gli aggiornamenti in sequenza sono utili quando è necessario effettuare aggiornamenti frequenti e minori all'applicazione software	`false`	`true` `false`

Nome	Descrizione	Impostazione predefinita	Valori validi
	Elastic Beanstalk e si desidera evitare tempi di inattività delle applicazioni. L'impostazione di questo valore su true attiva automaticamente le opzioni `MaxBatchSize` , `MinInstancesInService` e `PauseTime` . L'impostazione di una di queste opzioni, inoltre, imposta automaticamente il valore dell'opzione `RollingUpdateEnabled` su `true`. L'impostazione su `false` disattiva gli aggiornamenti in sequenza. **Note** Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente,		

Nome	Descrizione	Impostazione predefinita	Valori validi
	non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.		

Nome	Descrizione	Impostazione predefinita	Valori validi

RollingUpdateType	Sono inclusi tre tipi: aggiornamenti in sequenza basati sul tempo, aggiornamenti in sequenza basati sulla salute e aggiornamenti immutabili.
	
	Time-based gli aggiornamenti continui si applicano PauseTime tra i batch. Health-basedgli aggiornamenti continui attendono che le nuove istanze superino i controlli di integrità prima di passare al batch successivo. Gli aggiornamenti non modificabili eseguono un set completo di istanze in un nuovo gruppo Auto Scaling.
	
	Note
	Se utilizzi la console Elastic Beanstalk o la CLI EB
		`Time`	`Time`
			
			`Health`
			
			`Immutable`

Nome	Descrizione	Impostazione predefinita	Valori validi
	per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.		
PauseTime	Intervallo di attesa (in secondi, minuti o ore) del servizio Elastic Beanstalk dopo il completamento degli aggiornamenti a un batch di istanze e prima di passare al batch successivo.	Calcolato automaticamente in base al tipo di istanza e al container.	Da `PT0S*` (0 secondi) a `PT1H` (1 ora)

Nome	Descrizione	Impostazione predefinita	Valori validi
Timeout	Il periodo di attesa massimo (in minuti o ore) per il superamento dei controlli dello stato da parte di tutte le istanze in un batch di istanze prima che l'aggiornamento venga annullato.	`PT30M` (30 minuti)	Da `PT5M*` (5 minuti) a `PT1H` (1 ora) * Formato di durata ISO8601: `PT#H#M#S` dove ogni # è rispettivamente il numero di ore, minuti, and/or secondi.

aws:ec2:instances

Configurare le istanze dell'ambiente, incluse le opzioni Spot. Questo spazio dei nomi integra `aws:autoscaling:launchconfiguration` e `aws:autoscaling:asg`.

Per ulteriori informazioni, consulta the section called “Gruppo Auto Scaling”.

`Spazio dei nomi: aws:ec2:instances`

Nome	Descrizione	Impostazione predefinita	Valori validi

EnableSpot	Abilitare le richieste di istanze Spot per l'ambiente in uso. Quando `false`, alcune opzioni in questo spazio dei nomi non hanno effetto.
	
	Important
	Questa impostazione delle opzioni può far sì che Elastic Beanstalk migri un ambiente
		`false`	`true`
			
			`false`

Nome	Descrizione	Impostazione predefinita	Valori validi
	esistente con configurazioni di avvio verso modelli di avvio. A tale scopo sono necessari e le autorizzazioni necessarie per gestire i modelli di lancio. Queste autorizzazioni sono incluse nella nostra politica gestita. Se utilizzi politiche personalizzate anziché le nostre politiche gestite, la creazione o gli aggiornamenti dell'ambiente potrebbero non riuscire quando aggiorni la configurazione dell'ambiente. Per ulteriori informazioni e altre considerazioni, consultaMigrazione dell'ambiente Elastic Beanstalk ai modelli di avvio.		

Nome	Descrizione	Impostazione predefinita	Valori validi
InstanceTypes	Un elenco separato da virgole di tipi di istanza che si desidera utilizzare nell'ambiente (ad esempio, `t2.micro,t3.micro`). Quando `EnableSpot` è impostato su `true` ed `SpotAllocationStrategy` è impostato su `capacity-optimized-prioritized` , l'elenco di valori specificato in questa opzione determina la priorità del tipo di istanza per la strategia di allocazione delle istanze Spot. Quando le istanze Spot non sono attivate (`EnableSpot` è `false`), viene utilizzato solo il primo tipo di istanza nell'elenco. Il primo tipo di istanza nell'elenco di questa opzione è equivalente al valore dell'opzione `InstanceType` nello spazio dei nomi `aws:autoscaling:launchconfiguration` . Non si sconsiglia di utilizzare quest'ultima opzione perché è obsoleta. Se si specificano entrambi, viene utilizzato il primo tipo di istanza nell'elenco nell'opzione `InstanceTypes` e `InstanceType` viene ignorato.	Un elenco di due tipi di istanza Varia in base all'account e alla regione	Da uno a quaranta tipi di istanze EC2. (almeno due consigliati) Varia in base all'account, alla regione e alla zona di disponibilità. È possibile ottenere un elenco di tipi di istanza Amazon EC2 filtrati in base a questi valori. Per ulteriori informazioni, consulta la sezione Tipi di istanze disponibili nella Guida per l'utente di Amazon EC2. I tipi di istanza devono essere tutti parte della stessa architettura (`arm64`, `x86_64`, `i386`). Anche `SupportedArchitectures` fa parte di questo spazio dei nomi. Se fornisci dei valori per `SupportedArchitectures` , i valori che inserisci per `InstanceTypes` devono appartenere a una sola delle

Nome	Descrizione	Impostazione predefinita	Valori validi
	I tipi di istanza disponibili dipendono dalle zone di disponibilità e dalla regione utilizzate. Se si sceglie una sottorete, la zona di disponibilità che contiene tale sottorete determina i tipi di istanza disponibili. • Elastic Beanstalk non supporta i tipi di istanza Amazon EC2 Mac. • Per ulteriori informazioni sulle famiglie e i tipi di istanze di Amazon EC2, consulta Tipi di istanze nella Amazon EC2 User Guide. • Per ulteriori informazioni sui tipi di istanze disponibili nelle diverse regioni, consulta la sezione Tipi di istanze disponibili nella Amazon EC2 User Guide. **Note** Alcuni AWS account meno recenti potrebbero fornire a Elastic Beanstalk tipi di istanze predefiniti che non supportano le istanze Spot (ad esempio, t1.micro). Se si attivano le richieste di istanza Spot e viene visualizzato un errore su un tipo di istanza che non supporta Spot, assicurarsi di		architetture che fornisci per `Supported Architectures` .

Nome	Descrizione	Impostazione predefinita	Valori validi
	configurare i tipi di istanza che supportano Spot. Per scegliere i tipi di istanze Spot, utilizza Spot Instance Advisor. Quando si aggiorna la configurazione dell'ambiente e si rimuove uno o più tipi di istanza dall'opzione `InstanceTypes` , Elastic Beanstalk termina tutte le istanze Amazon EC2 in esecuzione su uno qualsiasi dei tipi di istanza rimossi. Il gruppo Auto Scaling dell'ambiente avvia quindi nuove istanze, se necessario per completare la capacità desiderata, utilizzando i tipi di istanza specificati correnti.		

Nome	Descrizione	Impostazione predefinita	Valori validi

SpotAllocationStrategy	Speciifica la strategia di allocazione delle istanze spot che determina il modo in cui le istanze Spot vengono allocate dai pool di capacità spot disponibili.
	
	Se impostato su `capacity-optimized-prioritized`, l'ordine dei valori in `InstanceTypes` imposta la priorità del tipo di istanza per l'allocazione.
	
	Questa opzione è rilevante solo quando `EnableSpot` è `true`.
		`capacity-optimized`	`capacity-optimized`
			
			`price-capacity-optimized`
			
			`capacity-optimized-prioritized`
			
			`lowest-price`

SpotFleetOnDemandBase	Il numero minimo di On-Demand istanze che il gruppo Auto Scaling fornisce prima di prendere in considerazione le istanze Spot man mano che l'ambiente cresce. Questa opzione è rilevante solo quando `EnableSpot` è `true`.	`0`	Da `0` all'opzione `MaxSize` nello spazio dei nomi `aws:autoscaling:asg`

Nome	Descrizione	Impostazione predefinita	Valori validi
SpotFleetOnDemandAboveBasePercentage	La percentuale di On-Demand istanze come parte della capacità aggiuntiva fornita dal gruppo Auto Scaling oltre `SpotOnDemandBase` alle istanze. Questa opzione è rilevante solo quando `EnableSpot` è `true`.	`0` per un ambiente a istanza singola `70` per un ambiente con bilanciamento del carico	`0` Da a `100`

Nome	Descrizione	Impostazione predefinita	Valori validi

SpotMaxPrice	Il prezzo massimo orario per unità che si è disposti a pagare per un'istanza Spot, in dollari americani. Per consigli sulle opzioni di prezzo massimo per le istanze Spot, consulta la cronologia dei prezzi delle istanze Spot nella Amazon EC2 User Guide.
	
	Questa opzione è rilevante solo quando `EnableSpot` è `true`.
		On-Demand prezzo per ogni tipo di istanza. Il valore dell'opzione in questo caso è `null`.	`0.001` Da a `20.0`
			
			`null`

Nome	Descrizione	Impostazione predefinita	Valori validi

SupportedArchitectures	Un elenco separato da virgole di tipi di istanza Amazon EC2 che userai nell'ambiente.
	
	Elastic Beanstalk supporta i tipi di istanza basati sulle seguenti architetture del processore:
	
	• AWS Architettura Arm Graviton a 64 bit (arm64)
	• Architettura a 64 bit (x86_64)
	• Architettura a 32-bit (i386)
	
	Per ulteriori informazioni sull'architettura del processore e sui tipi di istanza Amazon EC2, consulta the section called “Tipi di EC2 istanze Amazon”.
		Nessun	`arm64`
			
			`x86_64`
			
			`i386`
			
			Note
			L'architettura a 32 bit `i386` non è più supportata dalla maggior parte delle piattaforme Elastic Beanstalk. Consigliamo di scegliere invece i tipi di architettura `x86_64` o `arm64`.

aws:ec2:vpc

Configura l'ambiente per lanciare risorse in un Amazon Virtual Private Cloud personalizzato (Amazon VPC). Se non configuri le impostazioni in questo spazio dei nomi, Elastic Beanstalk avvia le risorse nel VPC di default.

Spazio dei nomi: aws:ec2:vpc

Nome	Descrizione	Impostazione predefinita	Valori validi
VPCId	L'ID dell'istanza Amazon VPC in uso.	Nessuna	
Subnets	Gli ID della sottorete o delle sottoreti del gruppo Auto Scaling. Se disponi di più sottoreti, specifica il valore come stringa di ID di sottorete delimitati da virgole (ad esempio, `"subnet-11111111,subnet-22222222"`).	Nessuno	
ELBSubnets	Gli ID della sottorete o delle sottoreti per l'Elastic Load Balancer. Se disponi di più sottoreti, specifica il valore come stringa di ID di sottorete delimitati da virgole (ad esempio, `"subnet-11111111,subnet-22222222"`).	Nessuna	
ELBScheme	Specifica `internal` se intendi creare un sistema di bilanciamento del carico interno in Amazon VPC, in modo che non sia possibile accedere all'applicazione Elastic Beanstalk dall'esterno di Amazon VPC. Se specifichi un valore diverso a `public` o `internal`, Elastic Beanstalk ignorerà il valore.	`public`	`public` `internal`
DBSubnets	Contiene gli ID delle sottoreti del database. Viene utilizzato solo se intendi aggiungere un'istanza database di Amazon RDS come parte dell'applicazione. Se disponi di più sottoreti, specifica il valore come stringa di ID di sottorete delimitati da virgole (ad esempio, `"subnet-11111111,subnet-22222222"`).	Nessuna	
AssociatePublicIpAddress	Specifica se avviare le istanze con indirizzi IP pubblici in Amazon VPC. Le istanze con indirizzi IP pubblici non necessitano di un dispositivo NAT per comunicare con Internet. Devi impostare il valore su `true` per includere	Nessuna	`true` `false`

Nome	Descrizione	Impostazione predefinita	Valori validi
	il sistema di bilanciamento del carico e le istanze in un'unica sottorete pubblica. Questa opzione non ha effetto su un ambiente a istanza singola, che ha sempre una singola istanza Amazon EC2 con un indirizzo IP elastico. L'opzione è rilevante per gli ambienti scalabili e con bilanciamento del carico.		

aws:elasticbeanstalk:application

Configura il percorso dei controlli dello stato per l'applicazione. Per ulteriori informazioni, consulta Report dello stato di base.

Spazio dei nomi: **`aws:elasticbeanstalk:application`**

Nome	Descrizione	Impostazione predefinita	Valori validi

URL dei controlli dello stato dell'applicazione	Il percorso al quale vengono inviate le richieste di controllo dello stato. Se questo percorso non è impostato, il servizio di bilanciamento del carico tenta di stabilire una connessione TCP sulla porta 80 per verificare lo stato dell'applicazione. Imposta un percorso che inizia con `/` per inviare una richiesta HTTP GET a quel percorso. Puoi anche includere un protocollo (HTTP, HTTPS, TCP o SSL) e la porta prima del percorso per verificare la	Nessu	I valori validi includono:
			
			`/` (HTTP GET al percorso root)
			
			`/health`
			
			`HTTPS:443/`
			
			`HTTPS:443/ health`

Nome	Descrizione	Impostazione predefinita	Valori validi
	connettività HTTPS o utilizzare una porta non di default. **Note** Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console sostituisce questa opzione con un valore consigliato.		

La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

aws:elasticbeanstalk:application:environment

Configura le proprietà di ambiente per l'applicazione.

Spazio dei nomi: **`aws:elasticbeanstalk:application:environment`**

Nome	Descrizione	Impostazione predefinita	Valori validi
Qualsiasi nome di variabile di ambiente.	Da passare in coppie chiave-valore.	Nessuna	Qualsiasi valore di variabile di ambiente.

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

aws: elasticbeanstalk:application:environmentsecrets

Configura le variabili di ambiente in modo che fungano da segreti di ambiente per la tua applicazione. I segreti di ambiente memorizzano Gestione dei segreti AWS segreti o AWS Systems Manager parametri di Parameter Store.

`Spazio dei nomi: aws:elasticbeanstalk:application:environmentsecrets`

Nome	Descrizione	Impostazione predefinita	Valori validi
OptionName	Speciifica il nome della variabile di ambiente che contiene l'archivio segreto o il valore dell'archivio dei parametri.	Nessun	Qualsiasi nome di variabile di ambiente.
Valore	Specifica l'ARN per il valore memorizzato Gestione dei segreti AWS nel AWS Systems Manager nostro Parameter Store. Durante il bootstrap dell'istanza, Elastic Beanstalk avvia la variabile di ambiente sul valore memorizzato in questa risorsa ARN. **Note** Assicurati che siano disponibili le autorizzazioni necessarie e per il ruolo del profilo di istanza EC2 del tuo ambiente per accedere agli ARN segreti e parametrici. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste.	Nessun	Valore ARN valido per un valore del AWS Systems Manager parametro Gestione dei segreti AWS segreto o Parameter Store.

Per ulteriori informazioni, consulta Configurazione dei segreti come variabili di ambiente.

aws:elasticbeanstalk:cloudwatch:logs

Configura lo streaming dei log delle istanze per l'applicazione.

`Spazio dei nomi: aws:elasticbeanstalk:cloudwatch:logs`

Nome	Descrizione	Impostazione predefinita	Valori validi
StreamLogs	Speciifica se creare gruppi in Logs per i log di proxy e distribuzione e per i log di streaming da ogni istanza dell' CloudWatch ambiente.	`false`	`true` `false`
DeleteOnTerminate	Specifica se eliminare i gruppi di log quando l'ambiente viene terminato. Se l'opzione è impostata su `false`, i log vengono conservati per `RetentionInDays` giorni.	`false`	`true` `false`
RetentionInDays	Il numero di giorni durante i quali conservare gli eventi di log prima della scadenza.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:cloudwatch:logs:health

Configura lo streaming dei log sullo stato dell'ambiente per l'applicazione.

Spazio dei **nomi**: aws:elasticbeanstalk:cloudwatch:logs:health

Nome	Descrizione	Impostazione predefinita	Valori validi
HealthStreamingEnabled	Per gli ambienti con report sullo stato avanzato abilitati, specifica se creare un gruppo in CloudWatch Logs for environment health e archiviare i dati sullo stato dell'ambiente Elastic Beanstalk. Per informazioni sull'abilitazione di informazioni sullo stato avanzate, consulta `aws:elasticbeanstalk:healthreporting:system`.	`false`	`true` `false`
DeleteOnTerminate	Specifica se eliminare il gruppo di log quando viene terminato l'ambiente. Se è `false`, i dati sullo stato vengono conservati per `RetentionInDays` giorni.	`false`	`true` `false`
RetentionInDays	Numero di giorni per cui conservare i dati sullo stato archiviati prima che scadano.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:command

Configura la policy di distribuzione per il codice dell'applicazione. Per ulteriori informazioni, consulta the section called “Opzioni di implementazione”.

Spazio dei nomi: aws:elasticbeanstalk:command

Nome	Descrizione	Impostazione predefinita	Valori validi

DeploymentPolicy	Scegli una policy di distribuzione per le distribuzioni della versione dell'applicazione.
	
	Note
	Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console sostituisce questa opzione con un valore consigliato.
		`AllAtOnce`	`AllAtOnce`
			
			`Rolling`
			
			`RollingWithAdditionalBatch`
			
			`Immutable`
			
			`TrafficSplitting`

Timeout	Il tempo di attesa, espresso in secondi, per il completamento dell'esecuzione dei comandi da parte di un'istanza. Elastic Beanstalk aggiunge internamente 240 secondi (quattro minuti) al valore `Timeout`. Ad esempio, il timeout effettivo per impostazione predefinita è 840 secondi (600+240) o 14 minuti.	`600`	1 Da a `3600`

BatchSizeType	Il tipo di numero specificato in. BatchSize
	
	Note
	Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e
		`Percentage`	`Percentage`
			
			`Fixed`

Nome	Descrizione	Impostazione predefinita	Valori validi
	CLI EB sostituiscono questa opzione con un valore consigliato.		
BatchSize	La percentuale o il numero fisso di istanze Amazon EC2 nel gruppo Auto Scaling su cui eseguire distribuzioni simultaneamente. I valori validi variano a seconda dell'BatchSize Typeimpostazione utilizzata. **Note** Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.	`100`	Da 1 a `100` (`Percentage`). 1a aws:autoscaling:asg:: () MaxSize `Fixed`
IgnoreHealthCheck	Non annullare una distribuzione non riuscita a causa del mancato superamento dei controlli dello stato.	`false`	`true` `false`

aws:elasticbeanstalk:environment

Configura il ruolo di servizio e l'architettura dell'ambiente.

`Spazio dei nomi: aws:elasticbeanstalk:environment`

Nome	Descrizione	Impostazione predefinita	Valori validi
EnvironmentType	Imposta il valore `SingleInstance` per avviare un'istanza EC2 senza sistema di bilanciamento del carico.	`LoadBalanced`	`SingleInstance`

Nome	Descrizione	Impostazione predefinita	Valori validi
			`LoadBalanced`
ServiceRole	Il nome di un ruolo IAM che Elastic Beanstalk utilizza per gestire le risorse per l'ambiente. Specifica il nome di un ruolo (facoltativamente preceduto da un percorso personalizzato) o il suo ARN. Esempi: • `aws-elasticbeanstalk-service-role` • *`custom-path /custom-role`* • `arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role` **Note** Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.	Nessuno	nome del ruolo IAM o ARN path/name

LoadBalancerType	Il tipo di sistema di bilanciamento del carico per l'ambiente. Per ulteriori informazioni, consultare the section called “Bilanciatore del carico”.	`classic`	`classic`
			
			`application`
			
			`network`

Nome	Descrizione	Impostazione predefinita	Valori validi

LoadBalancerIsShared	Specifica se il bilanciamento del carico dell'ambiente è dedicato o condiviso. Questa opzione può essere impostata solo per un sistema Application Load Balancer. Non può essere modificata dopo che l'ambiente è stato creato.
	
	Se è `false`, l'ambiente dispone di un proprio sistema di bilanciamento del carico dedicato, creato e gestito da Elastic Beanstalk. Quando `true`, l'ambiente utilizza un sistema di bilanciamento del carico condiviso, creato dall'utente e specificato nell'opzione `SharedLoadBalancer` dello spazio dei nomi aws:elbv2:loadbalancer.
		`false`	`true`
			
			`false`

aws:elasticbeanstalk:environment:process:default

Configura il processo di default dell'ambiente.

`Spazio dei nomi: aws:elasticbeanstalk:environment:process:default`

Nome	Descrizione	Impostazione predefinita	Valori validi
DeregistrationDelay	Il tempo di attesa, espresso in secondi, necessario al completamento delle richieste attive prima che venga annullata la registrazione di un'istanza.	`20`	`0` Da a `3600`

Nome	Descrizione	Impostazione predefinita	Valori validi
HealthCheckInterval	L'intervallo di tempo, espresso in secondi, nel quale Elastic Load Balancing controllerà lo stato delle istanze Amazon EC2 dell'applicazione.	Con Classic Load Balancer o Application Load Balancer: `15` Con Network Load Balancer: `30`	Con Classic Load Balancer o Application Load Balancer: da `5` a `300` Con Network Load Balancer: `10`, `30`
HealthCheckPath	Il percorso al quale vengono inviate le richieste HTTP per i controlli dello stato.	`/`	Un percorso instradabile.
HealthCheckTimeout	Il tempo di attesa, in secondi, per una risposta durante un controllo dello stato. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	`5`	`1` Da a `60`
HealthyThresholdCount	Il numero delle richieste consecutive riuscite prima che Elastic Load Balancing modifichi lo stato dell'istanza.	Con Classic Load Balancer o Application Load Balancer: `3` Con Network Load Balancer: `5`	`2` Da a `10`

Nome	Descrizione	Impostazione predefinita	Valori validi
MatcherHTTPCode	Elenco con valori separati da virgola di codici HTTP che indicano che un'istanza è integra. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer o una rete.	`200`	Con Application Load Balancer: da `200` a `499` Con Network Load Balancer: da `200` a `399`
Port	La porta sulla quale il processo è in ascolto.	`80`	`1` Da a `65535`

Protocol	Il protocollo utilizzato dal processo.
	
	Con un Application Load Balancer, puoi impostare questa opzione solo su `HTTP` o `HTTPS`.
	
	Con un Network Load Balancer, puoi impostare questa opzione solo su `TCP`.
		Con Classic Load Balancer o Application Load Balancer: `HTTP`
		
		Con Network Load Balancer: `TCP`
			`TCP`
			
			`HTTP`
			
			`HTTPS`

Nome	Descrizione	Impostazione predefinita	Valori validi

StickinessEnabled	Imposta il valore true per attivare le sticky session.
	
	Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.
		`'false'`	`'false'`
			
			`'true'`

StickinessLBCookie Duration	La durata, in secondi, del cookie delle sticky session. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	`86400` (un giorno)	`1` Da a `604800`
StickinessType	Imposta il valore `lb_cookie` per utilizzare cookie per le sticky session. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	`lb_cookie`	`lb_cookie`

Nome	Descrizione	Impostazione predefinita	Valori validi
UnhealthyThresholdCount	Il numero delle richieste consecutive non riuscite prima che Elastic Load Balancing modifichi lo stato dell'istanza.	`5`	`2` Da a `10`

aws:elasticbeanstalk:environment:process:process_name

Configura ulteriori processi per l'ambiente.

Spazio dei nomi: **`aws:elasticbeanstalk:environment:process:`*`nome_processo`***

Nome	Descrizione	Impostazione predefinita	Valori validi
DeregistrationDelay	Il tempo di attesa, espresso in secondi, necessario al completamento delle richieste attive prima che venga annullata la registrazione di un'istanza.	`20`	`0` Da a `3600`
HealthCheckInterval	L'intervallo, in secondi, in cui Elastic Load Balancing controlla lo stato delle istanze Amazon EC2 per la tua applicazione.	Con Classic Load Balancer o Application Load Balancer: `15` Con Network Load Balancer: `30`	Con Classic Load Balancer o Application Load Balancer: da `5` a `300` Con Network Load Balancer: `10`, `30`
HealthCheckPath	Il percorso al quale vengono inviate le	`/`	Un percorso instradabile.

Nome	Descrizione	Impostazione predefinita	Valori validi
	richieste HTTP per i controlli dello stato.		
HealthCheckTimeout	Il tempo di attesa, in secondi, per una risposta durante un controllo dello stato. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	5	1 Da a 60
HealthyThresholdCount	Il numero delle richieste consecutive riuscite prima che Elastic Load Balancing modifichi lo stato dell'istanza.	Con Classic Load Balancer o Application Load Balancer: 3 Con Network Load Balancer: 5	2 Da a 10
MatcherHTTPCode	Elenco con valori separati da virgola di codici HTTP che indicano che un'istanza è integra. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer o una rete.	200	Con Application Load Balancer: da 200 a 499 Con Network Load Balancer: da 200 a 399
Port	La porta su cui il processo è in ascolto.	80	1 Da a 65535

Nome	Descrizione	Impostazione predefinita	Valori validi

Protocol	Il protocollo utilizzato dal processo.
	
	Con un Application Load Balancer, puoi impostare questa opzione solo su `HTTP` o `HTTPS`.
	
	Con un Network Load Balancer, puoi impostare questa opzione solo su `TCP`.
		Con Classic Load Balancer o Application Load Balancer: `HTTP`
		
		Con Network Load Balancer: `TCP`
			`TCP`
			
			`HTTP`
			
			`HTTPS`

StickinessEnabled	Imposta il valore true per attivare le sticky session.
	
	Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.
		`'false'`	`'false'`
			
			`'true'`

StickinessLBCookie Duration	La durata, in secondi, del cookie delle sticky session. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	`86400` (un giorno)	`1` Da a `604800`

Nome	Descrizione	Impostazione predefinita	Valori validi
StickinessType	Imposta il valore `lb_cookie` per utilizzare cookie per le sticky session. Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	`lb_cookie`	`lb_cookie`
UnhealthyThreshold Count	Il numero delle richieste consecutive non riuscite prima che Elastic Load Balancing modifichi lo stato dell'istanza.	5	2 Da a `10`

aws:elasticbeanstalk:environment:proxy:staticfiles

Puoi utilizzare il namespace seguente per configurare il server proxy per l'elaborazione dei file statici. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione. In questo modo viene ridotta la quantità di richieste eseguibili dall'applicazione.

Effettua il mapping di un percorso servito dal server proxy su una cartella nel codice sorgente contenente asset statici. Ciascuna opzione da te definita in questo namespace mappa un percorso differente.

> **Note**
>
> Questo spazio dei nomi si applica alle ramificazioni della piattaforma basate su Amazon Linux 2 e versioni successive. Se l'ambiente utilizza una versione della piattaforma basata su AMI

Amazon Linux (precedente Amazon Linux 2), consulta the section called "Opzioni specifiche della piattaforma"per gli spazi dei nomi dei file statici specifici della piattaforma.

Spazio dei nomi: `aws:elasticbeanstalk:environment:proxy:staticfiles`

Nome	Valore
Il percorso in cui il server proxy servirà i file. Iniziare il valore con `/`. Ad esempio, specificare `/images` per servire i file in *`subdomain`* `.eleasticbeanstalk.com/images` .	Il nome della cartella contenente i file. Ad esempio, specificare `staticimages` per servire i file da una cartella denominata `staticimages` al primo livello del bundle di origine.

aws:elasticbeanstalk:healthreporting:system

Configura il reporting sullo stato migliorato per l'ambiente.

Spazio dei nomi: **`aws:elasticbeanstalk:healthreporting:system`**

Nome	Descrizione	Impostazione predefinita	Valori validi

SystemType	Il sistema di reporting dello stato (di base o migliorato). Il reporting sullo stato migliorato richiede un ruolo di servizio e una versione della piattaforma v2 o successiva.
	
	Note
	Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB
		`basic`	`basic`
			
			`enhanced`

Nome	Descrizione	Impostazione predefinita	Valori validi
	sostituiscono questa opzione con un valore consigliato.		
ConfigDocument	Un documento JSON che descrive l'ambiente e le metriche dell'istanza su cui pubblicare. CloudWatch	Nessuno	

EnhancedHealthAuthEnabled	Abilita l'autorizzazione per l'API interna utilizzata da Elastic Beanstalk per comunicare informazioni avanzate sullo stato dalle istanze dell'ambiente al servizio Elastic Beanstalk.
	
	Per ulteriori informazioni, consultare the section called “Ruoli dello stato migliorato”.
	
	Note
	Questa opzione è applicabile solo alle segnalazioni avanzate sullo stato (ad esempio, quando `SystemType` è impostato su `enhanced`).
		`true`	`true`
			
			`false`

Nome	Descrizione	Impostazione predefinita	Valori validi

HealthCheckSuccessThreshold	Abbassa la soglia dei controlli dello stato per le istanze.
	
	Note
	Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console sostituisce questa opzione con un valore consigliato.
		`Ok`	`Ok`
			`Warning`
			`Degraded`
			`Severe`

aws:elasticbeanstalk:hostmanager

Configura le istanze EC2 nell'ambiente per caricare i log ruotati in Amazon S3.

`Spazio dei nomi: aws:elasticbeanstalk:hostmanager`

Nome	Descrizione	Impostazione predefinita	Valori validi
LogPublicationControl	Copia i file di log delle istanze Amazon EC2 dell'applicazione nel bucket Amazon S3 associato all'applicazione.	`false`	`true` `false`

aws:elasticbeanstalk:managedactions

Configura gli aggiornamenti gestiti della piattaforma per l'ambiente.

Spazio dei nomi: **`aws:elasticbeanstalk:managedactions`**

Nome	Descrizione	Impostazione predefinita	Valori validi

ManagedActionsEnabled	Attiva gli aggiornamenti gestiti della piattaforma.
	
	Quando imposti il valore `true`, devi anche specificare `PreferredStartTime` e `UpdateLevel`.
		`false`	`true`
			
			`false`

PreferredStartTime	Configura una finestra di manutenzione per le azioni gestite in UTC. Ad esempio, `"Tue:09:00"`.	Nessuna	Giorno e ora nel formato *day*:*hour*:*minute* .
ServiceRoleForManagedUpdates	Nome di un ruolo IAM utilizzato da Elastic Beanstalk per eseguire aggiornamenti della piattaforma gestita per l'ambiente in uso. È possibile utilizzare lo stesso ruolo specificato per l'opzione `ServiceRole` dello spazio dei nomi `aws:elasticbeanstalk:environment` oppure il ruolo collegato al servizio degli aggiornamenti gestiti dell'account. In quest'ultimo caso, se l'account non dispone ancora di un ruolo collegato al servizio degli aggiornamenti gestiti,	Nessuna	Come `ServiceRole`. oppure `AWSServiceRoleForElasticBeanstalkManagedUpdates`

Nome	Descrizione	Impostazione predefinita	Valori validi
	Elastic Beanstalk crea tale ruolo.		

aws:elasticbeanstalk:managedactions:platformupdate

Configura gli aggiornamenti gestiti della piattaforma per l'ambiente.

Spazio dei nomi: **`aws:elasticbeanstalk:managedactions:platformupdate`**

Nome	Descrizione	Impostazione predefinita	Valori validi
UpdateLevel	Il livello più elevato di aggiornamento da applicare agli aggiornamenti gestiti della piattaforma. Le piattaforme sono versionate*`major`*. *`minor`*. *`patch`*. Ad esempio, 2.0.8 ha una versione principale 2, una versione secondaria 0 e una versione patch 8.	Nessuna	`patch` solo per gli aggiornamenti delle versioni patch. `minor` per gli aggiornamenti delle versioni secondarie e patch.

InstanceRefreshEnabled	Abilita la sostituzione settimanale delle istanze.
	
	Richiede che `ManagedActionsEnabled` sia impostato su `true`.
		`false`	`true`
			
			`false`

aws:elasticbeanstalk:monitoring

Configura l'ambiente per terminare le istanze EC2 che non superano i controlli dello stato.

Spazio dei nomi: `aws:elasticbeanstalk:monitoring`

Nome	Descrizione	Impostazione predefinita	Valori validi

Automatically Terminate Unhealthy Instances	Termina un'istanza se non supera i controlli dello stato.
	
	Note
	Questa opzione era supportata solo in ambienti legacy. Determinava l'integrità di un'istanza in base alla possibilità di raggiungerla e in base ad altri parametri d'istanza. Elastic Beanstalk non fornisce un modo per terminare automaticamente le istanze in base allo stato dell'applicazione.
		`true`	`true`
			
			`false`

aws:elasticbeanstalk:sns:topics

Configura notifiche per l'ambiente.

Spazio dei nomi: **`aws:elasticbeanstalk:sns:topics`**

Nome	Descrizione	Impostazione predefinita	Valori validi
Notification Endpoint	L'endpoint in cui intendi ricevere le notifiche di importanti eventi che riguardano la tua applicazione. **Note** Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console sostituisce questa opzione con un valore consigliato.	Nessuno	

Notification Protocol	Il protocollo utilizzato per inviare notifiche all'endpoint.	`email`	`http`
			
			`https`
			
			`email`
			
			`email-json`
			
			`sqs`

Notification Topic ARN	L'Amazon Resource Name (ARN) dell'argomento a cui hai effettuato la sottoscrizione.	Nessuna	
Notification Topic Name	Il nome dell'argomento a cui hai effettuato la sottoscrizione.	Nessuna	

aws:elasticbeanstalk:sqsd

Configura la coda Amazon SQS per un ambiente worker.

Spazio dei nomi: **`aws:elasticbeanstalk:sqsd`**

Nome	Descrizione	Impostazione predefinita	Valori validi
WorkerQueueURL	L'URL della coda da cui il daemon nel livello ambiente lavoratore legge i messaggi. **Note** Quando non specifichi un valore, la coda creata automaticamente da Elastic Beanstalk è una coda standard di Amazon SQS. Quando fornisci un valore, puoi fornire l'URL di una coda Amazon SQS FIFO o standard. Se si fornisce una coda FIFO, le attività periodiche non sono supportate.	generato automaticamente	Se non specifichi un valore, Elastic Beanstalk crea automaticamente una coda.
HttpPath	Il percorso relativo all'applicazione alla quale vengono inviati messaggi HTTP POST.	/	

MimeType	Il tipo MIME del messaggio inviato nella richiesta HTTP POST.	`application/json`	`application/json`
			
			`application/x-www-form-urlencoded`

Nome	Descrizione	Impostazione predefinita	Valori validi

			`application/xml`
			
			`text/plain`
			
			Il tipo MIME personalizzato.

HttpConnections	Il numero massimo di connessioni simultanee a qualsiasi applicazione all'interno di un'istanza Amazon EC2. **Note** Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console sostituisce questa opzione con un valore consigliato.	`50`	1 Da a `100`
ConnectTimeout	Il periodo di attesa, in secondi, per connessioni riuscite a un'applicazione.	`5`	1 Da a `60`

Nome	Descrizione	Impostazione predefinita	Valori validi
InactivityTimeout	Il periodo di attesa, in secondi, per una risposta su una connessione esistente a un'applicazione. Il messaggio viene rielaborato fino a che il daemon riceve una risposta 200 (OK) dall'applicazione nel livello ambiente lavoratore o `RetentionPeriod` scade.	`299`	`1` Da a `36000`
VisibilityTimeout	Il periodo di tempo, in secondi, durante il quale un messaggio in entrata dalla coda Amazon SQS viene bloccato per l'elaborazione. Trascorso il periodo di tempo impostato, il messaggio è di nuovo visibile nella coda per la lettura con qualsiasi altro daemon.	300	`0` Da a `43200`
ErrorVisibilityTimeout	Il periodo di tempo, in secondi, prima che Elastic Beanstalk restituisca un messaggio alla coda Amazon SQS dopo un tentativo di elaborazione non riuscito con errore esplicito.	2 secondi	Da `0` a `43200` secondi
Retention Period	Il periodo di tempo, in secondi, durante il quale un messaggio è valido e viene elaborato attivamente.	`345600`	`60` Da a `1209600`

Nome	Descrizione	Impostazione predefinita	Valori validi
MaxRetries	Il numero massimo di tentativi che Elastic Beanstalk effettua per inviare il messaggio all'applicazione Web che lo elaborerà prima di spostare il messaggio nella coda DLQ.	`10`	1 Da a `100`

aws:elasticbeanstalk:trafficsplitting

Configura le distribuzioni con suddivisione del traffico per l'ambiente in uso.

Questo spazio dei nomi si applica quando imposti l'opzione `DeploymentPolicy` dello spazio dei nomi aws:elasticbeanstalk:command su `TrafficSplitting`. Per ulteriori informazioni sulle policy di distribuzione, consulta the section called “Opzioni di implementazione”.

Spazio dei nomi: **`aws:elasticbeanstalk:trafficsplitting`**

Nome	Descrizione	Impostazione predefinita	Valori validi
NewVersionPercent	Percentuale iniziale di traffico client in entrata che Elastic Beanstalk passa alle istanze dell'ambiente che eseguono la nuova versione dell'applicazione che stai distribuendo.	`10`	1 Da a `100`
EvaluationTime	Tempo di attesa espresso in minuti per Elastic Beanstalk dopo una distribuzione iniziale integra prima di procedere allo spostamento di	`5`	3 Da a `600`

Nome	Descrizione	Impostazione predefinita	Valori validi
	tutto il traffico client in entrata alla nuova versione dell'applicazione che stai distribuendo.		

aws:elasticbeanstalk:xray

Esegui il AWS X-Ray demone per inoltrare le informazioni di traccia dall'applicazione integrata. X-Ray

`Spazio dei nomi: aws:elasticbeanstalk:xray`

Nome	Descrizione	Impostazione predefinita	Valori validi
`XRayEnabled`	Imposta per eseguire il demone sulle istanze del tuo ambiente. `true` X-Ray	`false`	`true` `false`

aws:elb:healthcheck

Configura i controlli dello stato per un sistema Classic Load Balancer.

`Spazio dei nomi: aws:elb:healthcheck`

Nome	Descrizione	Impostazione predefinita	Valori validi
HealthyThreshold	Il numero delle richieste consecutive riuscite prima che Elastic Load Balancing modifichi lo stato dell'istanza.	`3`	2 Da a `10`

Nome	Descrizione	Impostazione predefinita	Valori validi
Interval	L'intervallo in cui Elastic Load Balancing controlla lo stato delle istanze Amazon EC2 dell'applicazione.	`10`	`5` Da a `300`
Timeout	Il tempo, in secondi, nel quale Elastic Load Balancing attende una risposta prima di ritenere che l'istanza non risponde.	`5`	`2` Da a `60`
Unhealthy Threshold	Il numero delle richieste consecutive non riuscite prima che Elastic Load Balancing modifichi lo stato dell'istanza.	`5`	`2` Da a `10`
(Obsoleta) Target	La destinazione su un'istanza back-end alla quale vengono inviati i controlli dello stato. Utilizza piuttosto `Application Healthcheck URL` nel namespace `aws:elasticbeanstalk:application`.	`TCP:80`	Target nel formato: *`PROTOCOL PORT/PATH`*

aws:elb:loadbalancer

Configura il sistema Classic Load Balancer dell'ambiente.

Molte delle opzioni in questo spazio dei nomi non sono più supportate e sono state sostituite da opzioni specifiche del listener nello spazio dei nomi aws:elb:listener. Con queste opzioni che non sono più supportate è possibile configurare solo due listener (uno sicuro e uno non sicuro) sulle porte standard.

`Spazio dei nomi: aws:elb:loadbalancer`

Nome	Descrizione	Default	Valori validi
CrossZone	Configura il sistema di bilanciamento del carico per instradare il traffico in modo uniforme su tutte le istanze in tutte le zone	`false`	`true` `false`

Nome	Descrizione	Default	Valori validi
	di disponibilità anziché solo all'interno di ogni zona. **Note** Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.		
SecurityGroups	Assegna uno o più gruppi di sicurezza che hai creato per il sistema di bilanciamento del carico. Obbligatorio `DisableDefaultEC2SecurityGroup` se () è impostato su. aws:autoscaling:launchconfiguration `true` Gli ambienti con bilanciamento del carico che hanno disattivato il gruppo di sicurezza Elastic Beanstalk EC2 predefinito devono fornire a uno o più gruppi di sicurezza questa opzione. Per ulteriori informazioni, consulta Gestione dei gruppi di sicurezza EC2.	Nessuno	Uno o più ID di gruppo di sicurezza.

Nome	Descrizione	Default	Valori validi
ManagedSecurityGroup	Assegna un gruppo di sicurezza esistente al sistema di bilanciamento del carico dell'ambiente, anziché crearne uno. Per utilizzare questa impostazione, aggiornare l'impostazione `SecurityGroups` in questo spazio dei nomi di modo che includa l'ID del gruppo di sicurezza e rimuovi l'ID del gruppo di protezione creato automaticamente, se ne è stato creato uno. Per consentire il traffico tra il sistema di bilanciamento del carico e le istanze EC2 dell'ambiente, Elastic Beanstalk aggiunge una regola al gruppo di sicurezza delle istanze che autorizza il traffico in entrata dal gruppo di sicurezza gestito.	Nessuna	Un ID di gruppo di sicurezza.

(Obsoleta) LoadBalancerHTTPPort	La porta sulla quale il listener non protetto è in ascolto.	`80`	`OFF`
			
			`80`

(Obsoleta) LoadBalancerPortProtocol	Il protocollo da utilizzare per il listener non protetto.	`HTTP`	`HTTP`
			
			`TCP`

(Obsoleta) LoadBalancerHTTPSPort	La porta sulla quale il listener protetto è in ascolto.	`OFF`	`OFF`
			
			`443`
			
			`8443`

(Obsoleta) LoadBalancerSSLPortProtocol	Il protocollo da utilizzare per il listener protetto.	`HTTPS`	`HTTPS`
			
			`SSL`

(Obsoleta) SSLCertificateId	L'Amazon Resource Name (ARN) di un certificato SSL da associare al listener protetto.	Nessuna	

aws:elb:listener

Configura il listener di default (porta 80) su un sistema Classic Load Balancer.

`Spazio` dei nomi: aws:elb:listener

Nome	Descrizione	Default	Valori validi
ListenerProtocol	Il protocollo utilizzato dal listener.	`HTTP`	`HTTP` `TCP`
InstancePort	La porta che questo listener utilizza per comunicare con le istanze EC2.	80	1 Da a `65535`

InstanceProtocol	Il protocollo che questo listener utilizza per comunicare con le istanze EC2.
	
	Deve trovarsi allo stesso livello di protocollo Internet di `ListenerProtocol` . Deve avere anche lo stesso livello di sicurezza di qualsiasi altro listener che usa lo stesso oggetto `InstancePort` di questo listener.
	
	Se, ad esempio, `ListenerProtocol` è `HTTPS` (livello applicazione, con l'uso di una connessione sicura), puoi impostare `InstanceProtocol` su `HTTP` (sempre a livello applicazione, con l'uso di una connessione non sicura). Se, inoltre, imposti `InstancePort` su `80`, devi impostare `InstanceProtocol` su `HTTP` in tutti gli altri listener con il valore di `InstancePort` impostato su `80`.
		`HTTP` quando `ListenerProtocol` è `HTTP`
		
		`TCP` quando `ListenerProtocol` è `TCP`
			`HTTP` o `HTTPS` quando `ListenerProtocol` è `HTTP` o `HTTPS`
			
			`TCP` o `SSL` quando `ListenerProtocol` è `TCP` o `SSL`

PolicyNames	Un elenco di nomi di policy separati da virgole da applicare alla porta per questo listener. Ti consigliamo invece di utilizzare l'opzione del namespace. LoadBalancerPorts aws:elb:policies	Nessuno	

Nome	Descrizione	Default	Valori validi

ListenerEnabled	Specifica se il listener è abilitato. Se si specifica `false`, il listener non viene incluso nel servizio di bilanciamento del carico.	`true`	`true`
			
			`false`

aws:elb:listener:listener_port

Configura ulteriori listener su un sistema Classic Load Balancer.

Spazio dei nomi: aws:elb:listener: ***porta_listener***

Nome	Descrizione	Default	Valori validi
ListenerProtocol	Il protocollo utilizzato dal listener.	`HTTP`	`HTTP` `HTTPS` `TCP` `SSL`
InstancePort	La porta che questo listener utilizza per comunicare con le istanze EC2.	Lo stesso di. *`listener_port`*	1 Da a 65535

InstanceProtocol	Il protocollo che questo listener utilizza per comunicare con le istanze EC2.
	
	Deve trovarsi allo stesso livello di protocollo Internet di `ListenerProtocol` . Deve avere anche lo stesso livello di sicurezza di qualsiasi altro listener che usa lo stesso oggetto `InstancePort` di questo listener.
	
	Se, ad esempio, `ListenerProtocol` è `HTTPS` (livello applicazione, con l'uso di una connessione sicura), puoi impostare `InstanceProtocol` su `HTTP` (sempre a livello applicazione, con l'uso di una connessione non sicura). Se, inoltre, imposti
		`HTTP` quando `ListenerProtocol` è `HTTP` o `HTTPS`
		
		`TCP` quando `ListenerProtocol` è `TCP` o `SSL`
			`HTTP` o `HTTPS` quando `ListenerProtocol` è `HTTP` o `HTTPS`
			
			`TCP` o `SSL` quando `ListenerProtocol` è `TCP` o `SSL`

Nome	Descrizione	Default	Valori validi
	`InstancePort` su `80`, devi impostare `InstanceProtocol` su `HTTP` in tutti gli altri listener con il valore di `InstancePort` impostato su `80`.		
PolicyNames	Un elenco di nomi di policy separati da virgole da applicare alla porta per questo listener. Ti suggeriamo di utilizzare invece l' LoadBalancerPorts opzione del aws:elb:policies namespace.	Nessuno	
SSLCertificateId	L'Amazon Resource Name (ARN) di un certificato SSL da associare al listener.	Nessuna	
ListenerEnabled	Specifica se il listener è abilitato. Se si specifica `false`, il listener non viene incluso nel servizio di bilanciamento del carico.	`true` se è impostata una qualsiasi altra opzione; altrimenti `false`.	`true` `false`

aws:elb:policies

Modifica l'adesione di default e le policy globali di sistema di bilanciamento del carico per un sistema Classic Load Balancer.

Spazio dei nomi: aws:elb:policies

Nome	Descrizione	Impostazione predefinita	Valori validi

ConnectionDrainingEnabled	Specifica se il sistema di bilanciamento del carico mantiene le connessioni esistenti alle istanze che non sono più integre o la cui registrazione è stata annullata per completare le richieste in corso.
	
	Note
	Se utilizzi la console Elastic Beanstalk o la CLI EB per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La console e CLI EB sostituiscono questa opzione con un valore consigliato.
		`false`	`true`
			
			`false`

ConnectionDrainingTimeout	Il periodo di tempo massimo in secondi durante il quale il sistema di bilanciamento del carico mantiene le connessioni esistenti a un'istanza durante il connection draining prima di forzare la chiusura delle connessioni. **Note** Se utilizzi la console Elastic Beanstalk per creare un ambiente, non puoi impostare questa opzione in un file di configurazione. La	`20`	1 Da a `3600`

Nome	Descrizione	Impostazione predefinita	Valori validi
	console sostituisce questa opzione con un valore consigliato.		
ConnectionSettingIdleTimeout	Il periodo di tempo, espresso in secondi, durante il quale il sistema di bilanciamento del carico attende eventuali dati inviati o ricevuti tramite la connessione. Se non vengono inviati o ricevuti dati allo scadere di questo periodo di tempo, il sistema di bilanciamento del carico chiude la connessione.	`60`	`1` Da a `3600`
LoadBalancerPorts	Un elenco di porte di listener separate da virgole a cui viene applicata la policy di default (`AWSEB-ELB-StickinessPolicy`).	Nessuna	Puoi utilizzare `:all` per indicare tutte le porte di listener
Stickiness Cookie Expiration	Il periodo di validità, in secondi, di ogni cookie. Utilizza la policy di default (`AWSEB-ELB-StickinessPolicy`).	`0`	`0` Da a `1000000`
Stickiness Policy	Associa una sessione utente a una determinata istanza del server, in modo che tutte le richieste provenienti dall'utente durante la sessione siano inviate alla stessa istanza del server. Utilizza la policy di default (`AWSEB-ELB-StickinessPolicy`).	`false`	`true false`

aws:elb:policies:policy_name

Crea ulteriori policy per un sistema di bilanciamento del carico per Classic Load Balancer.

Spazio dei nomi: **`aws:elb:policies:`*`nome_policy`***

Nome	Descrizione	Impostazione predefinita	Valori validi
CookieName	Il nome del cookie generato dall'applicazione che controlla la durata delle sessioni di una policy `AppCookieStickinessPolicyType` . Questa politica può essere associata solo agli ascoltatori. HTTP/HTTPS	Nessuno	
InstancePorts	Un elenco di porte di istanza separate da virgole a cui questa policy viene applicata.	Nessuna	Un elenco di porte o `:all`
LoadBalancerPorts	Un elenco di porte di listener separate da virgole a cui questa policy viene applicata.	Nessuna	Un elenco di porte o `:all`
ProxyProtocol	Per una policy `ProxyProtocolPolicyType` , specifica se includere l'indirizzo IP e la porta della richiesta d'origine per i messaggi TCP. Questa politica può essere associata solo agli TCP/SSL ascoltatori.	Nessuno	`true false`
PublicKey	Il contenuto di una chiave pubblica per una policy `PublicKeyPolicyType` da utilizzare durante l'autenticazione di uno o più server di back-end. Questa policy non può essere applicata direttamente ai server di back-end o ai listener. Deve far parte di una policy `BackendServerAuthenticationPolicyType` .	Nessuno	

Nome	Descrizione	Impostazione predefinita	Valori validi
PublicKeyPolicyNames	Un elenco di nomi di policy separati da virgole (a partire dalle policy `PublicKeyPolicyType`) per una policy `BackendServerAuthenticationPolicyType` che controlla l'autenticazione su uno o più server di back-end o un server. Questa politica può essere associata solo ai server di backend che utilizzano. HTTPS/SSL	Nessuno	
SSLProtocols	Un elenco di protocolli SSL separati da virgole da attivare per una policy `SSLNegotiationPolicyType` che definisce le crittografie e i protocolli che sono accettati dal sistema di bilanciamento del carico. Questa politica può essere associata solo agli HTTPS/SSL ascoltatori.	Nessuno	
SSLReferencePolicy	Il nome di una politica di sicurezza predefinita che aderisce alle migliori pratiche di AWS sicurezza e che si desidera attivare per una `SSLNegotiationPolicyType` politica che definisce i codici e i protocolli accettati dal load balancer. Questa politica può essere associata solo ai listener. HTTPS/SSL	Nessuno	
Stickiness Cookie Expiration	Il periodo di validità, in secondi, di ogni cookie.	`0`	`0` Da a `1000000`

Nome	Descrizione	Impostazione predefinita	Valori validi
Stickiness Policy	Associa una sessione utente a una determinata istanza del server, in modo che tutte le richieste provenienti dall'utente durante la sessione siano inviate alla stessa istanza del server.	`false`	`true` `false`

aws:elbv2:listener:default

Configura il listener di default (porta 80) su un sistema Application Load Balancer o un sistema Network Load Balancer.

Questo spazio dei nomi non si applica a un ambiente che utilizza un sistema di bilanciamento del carico condiviso. I sistemi di bilanciamento del carico condivisi non dispongono di un listener predefinito.

`Spazio dei nomi: aws:elbv2:listener:default`

Nome	Descrizione	Impostazione predefinita	Valori validi
DefaultProcess	Il nome del processo a cui inoltrare il traffico quando non vi sono regole corrispondenti.	`default`	Un nome di processo.
ListenerEnabled	Imposta il valore `false` per disabilitare il listener. Puoi utilizzare questa opzione per disabilitare il listener di default sulla porta 80.	`true`	`true` `false`

Nome	Descrizione	Impostazione predefinita	Valori validi
Protocol	Il protocollo di traffico da elaborare.	Con Application Load Balancer: `HTTP` Con Network Load Balancer: `TCP`	Con Application Load Balancer: `HTTP`, `HTTPS` Con Network Load Balancer: `TCP`, `TLS`
Rules	Un elenco di regole da applicare al listener Questa opzione è applicabile solo agli ambienti con un sistema Application Load Balancer.	Nessuna	Un elenco separato da virgole di nomi di regole.
SSLCertificateArns	L'Amazon Resource Name (ARN) del certificato SSL da associare al listener. Questa opzione è applicabile solo agli ambienti con Application Load Balancer o Network Load Balancer.	Nessuno	L'ARN di un certificato archiviato in IAM o ACM.

Nome	Descrizione	Impostazione predefinita	Valori validi
SSLPolicy	Specifica una policy di sicurezza da applicare al listener. Questa opzione è applicabile solo agli ambienti con Application Load Balancer o Network Load Balancer.	Nessuno (valore di default ELB)	Il nome di una policy di sicurezza per sistema di bilanciamento del carico.

aws:elbv2:listener:listener_port

Configura ulteriori listener su un sistema Application Load Balancer o un sistema Network Load Balancer.

> **Note**
>
> Per un sistema Application Load Balancer condiviso, puoi specificare solo l'opzione `Rule`. Le altre opzioni non sono applicabili ai sistemi di bilanciamento del carico condivisi.

Spazio dei nomi: aws:elbv2:listener: ***`porta_listener`***

Nome	Descrizione	Impostazione predefinita	Valori validi
DefaultProcess	Il nome del processo a cui viene inoltrato il traffico quando non vi sono regole corrispondenti.	`default`	Un nome di processo.
ListenerEnabled	Imposta il valore `false` per disabilit	`true`	`true`

Nome	Descrizione	Impostazione predefinita	Valori validi
	are il listener. Puoi utilizzare questa opzione per disabilit are il listener di default sulla porta 80.		`false`
Protocol	Il protocollo di traffico da elaborare.	Con Application Load Balancer: `HTTP` Con Network Load Balancer: `TCP`	Con Application Load Balancer: `HTTP`, `HTTPS` Con Network Load Balancer: `TCP`, `TLS`
Rules	Un elenco di regole da applicare al listener. Questa opzione è applicabile solo agli ambienti con un sistema Application Load Balancer. Se l'ambiente utilizza un Application Load Balancer condiviso e non specifichi questa opzione per nessun listener, Elastic Beanstalk associa automaticamente la regola `default` a un listener della porta 80.	Nessuna	Un elenco separato da virgole di nomi di regole.

Nome	Descrizione	Impostazione predefinita	Valori validi
SSLCertificateArns	L'Amazon Resource Name (ARN) del certificato SSL da associare al listener. Questa opzione è applicabile solo agli ambienti con Application Load Balancer o Network Load Balancer.	Nessuno	L'ARN di un certificato archiviato in IAM o ACM.
SSLPolicy	Specifica una policy di sicurezza da applicare al listener. Questa opzione è applicabile solo agli ambienti con Application Load Balancer o Network Load Balancer.	Nessuno (valore di default ELB)	Il nome di una policy di sicurezza per sistema di bilanciamento del carico.

aws:elbv2:listenerrule:rule_name

Definisci le regole del listener per un sistema Application Load Balancer. Se una richiesta corrisponde ai nomi host o ai percorsi in una regola, il sistema di bilanciamento del carico la inoltra al processo specificato. Per usare una regola, aggiungila a un listener con l'opzione `Rules` nel namespace `aws:elbv2:listener:listener_port`.

> **Note**
>
> Questo namespace non è applicabile agli ambienti con un Network Load Balancer.

Spazio dei nomi: aws:elbv2:listenerrule: ***rule_name***

Nome	Descrizione	Impostazione predefinita	Valori validi
HostHeaders	Un elenco di nomi host con cui trovare la corrispondenza. Ad esempio, `my.example.com` .	Sistema di bilanciamento del carico dedicato: nessuno Sistema di bilanciamento del carico condiviso: CNAME dell'ambiente	Ogni nome può contenere fino a 128 caratteri. Un modello può includere lettere maiuscole e minuscole, numeri, trattini (–) e fino a tre caratteri jolly (`*` corrisponde a zero o più caratteri ; `?` corrisponde esattamente a un carattere). È possibile elencare più di un nome, ciascuno separato da una virgola. Un sistema Application Load Balancer supporta fino a cinque regole `PathPattern` e `HostHeader` combinate. Per ulteriori informazioni, consulta Condizioni host nella Guida per l'utente dei sistemi Application Load Balancer.

Nome	Descrizione	Impostazione predefinita	Valori validi
PathPatterns	I modelli di percorso da abbinare (ad esempio, `/img/*`). Questa opzione è applicabile solo agli ambienti con un Application Load Balancer.	Nessuna	Ogni modello può contenere fino a 128 caratteri. Un modello può includere lettere maiuscole e minuscole, numeri, trattini (–) e fino a tre caratteri jolly (`*` corrisponde a zero o più caratteri ; `?` corrisponde esattamente a un carattere). È possibile aggiungere più modelli di percorso separati da virgole. Un sistema Application Load Balancer supporta fino a cinque regole `PathPattern` e `HostHeader` combinate. Per ulteriori informazioni, consulta Condizioni del percorso nella Guida per l'utente dei sistemi Application Load Balancer.

Nome	Descrizione	Impostazione predefinita	Valori validi
Priority	La priorità di questa regola quando più regole corrispondono. Il numero più basso ha la priorità. Due regole non possono avere la stessa priorità. Con un sistema di bilanciamento del carico condiviso, Elastic Beanstalk tratta le priorità delle regole come relative tra gli ambienti di condivisione e le mappa alle priorità assolute durante la creazione.	`1`	`1` Da a `1000`
Process	Il nome del processo a cui inoltrare il traffico quando questa regola corrisponde alla richiesta.	`default`	Un nome di processo.

aws:elbv2:loadbalancer

Configura un sistema Application Load Balancer.

Per un sistema di bilanciamento del carico condiviso, sono valide solo le opzioni `SharedLoadBalancer` e `SecurityGroups`.

> **Note**
>
> Questo spazio dei nomi non è applicabile agli ambienti con un sistema Network Load Balancer.

Spazio dei nomi: **`aws:elbv2:loadbalancer`**

Nome	Descrizione	Impostazione predefinita	Valori validi
AccessLogsS3Bucket	Il bucket Amazon S3 in cui sono memorizzati i log di accesso. Il bucket deve essere nella stessa regione dell'ambiente e consentire l'accesso in scrittura al sistema di bilanciamento del carico.	Nessuna	Un nome di bucket.
AccessLogsS3Enabled	Abilita lo storage dei log di accesso.	`false`	`true` `false`
AccessLogsS3Prefix	Prefisso da anteporre ai nomi di log di accesso. Per impostazione predefinita, il sistema di bilanciamento del carico posiziona i log in una directory denominata AWSLogs nel bucket specificato. Specifica un prefisso per posizionare la directory AWSLogs in un'altra directory.	Nessuna	
IdleTimeout	Il periodo di attesa, in secondi, per il completamento di una richiesta prima della chiusura delle connessioni a client e istanza.	Nessuna	`1` Da a `3600`
IpAddressType	Specifica la configurazione del formato dell'indirizzo IP per il bilanciamento del carico dell'ambiente. Utilizza l'opzione dualstack per abilitare il protocollo IPv6, insieme al protocollo IPv4.	ipv4	ipv4, dualstack

Nome	Descrizione	Impostazione predefinita	Valori validi
	Solo Application Load Balancer e Network Load Balancer supportano l'opzione dualstack. Gli ambienti a istanza singola e gli ambienti che utilizzano Classic Load Balancer non la supportano. È necessario associare il VPC dell'ambiente e tutte le sottoreti VPC ai blocchi CIDR IPv6 per supportare l'opzione dualstack. Per ulteriori informazioni, consulta the section called “Configurazione del dual-stack”.		

Nome	Descrizione	Impostazione predefinita	Valori validi
ManagedSecurityGroup	Assegna un gruppo di sicurezza esistente al sistema di bilanciamento del carico dell'ambiente, anziché crearne uno. Per utilizzare questa impostazione, aggiorna l'impostazione `SecurityGroups` in questo namespace di modo che includa l'ID del tuo gruppo di sicurezza e rimuovi l'ID dell'eventuale gruppo di sicurezza creato automaticamente. Per consentire il traffico tra il sistema di bilanciamento del carico e le istanze EC2 dell'ambiente, Elastic Beanstalk aggiunge una regola al gruppo di sicurezza delle istanze che autorizza il traffico in entrata dal gruppo di sicurezza gestito.	Il gruppo di sicurezza che Elastic Beanstalk crea per il sistema di bilanciamento del carico.	Un ID di gruppo di sicurezza.

Nome	Descrizione	Impostazione predefinita	Valori validi
SecurityGroups	Un elenco di gruppi di sicurezza da collegare al sistema di bilanciamento del carico. Obbligatorio se `DisableDefaultEC2SecurityGroup` () aws:autoscaling:launchconfiguration è impostato su. `true` Gli ambienti con bilanciamento del carico che hanno disattivato il gruppo di sicurezza Elastic Beanstalk EC2 predefinito devono fornire a uno o più gruppi di sicurezza questa opzione. Per ulteriori informazioni, consulta Gestione dei gruppi di sicurezza EC2. Per un sistema di bilanciamento del carico condiviso, se non si specifica questo valore, Elastic Beanstalk verifica se un gruppo di sicurezza esistente che gestisce è già collegato al sistema di bilanciamento del carico. Se uno non è collegato al sistema di bilanciamento del carico, Elastic Beanstalk crea un gruppo di sicurezza e lo collega al sistema di bilanciamento del carico. Elastic Beanstalk elimina questo gruppo di sicurezza quando termina l'ultimo ambiente che condivide il sistema di bilanciamento del carico.	Il gruppo di sicurezza che Elastic Beanstalk crea per il sistema di bilanciamento del carico.	Comma-separated elenco di ID dei gruppi di sicurezza.

Nome	Descrizione	Impostazione predefinita	Valori validi
	I gruppi di sicurezza del sistema di bilanciamento del carico vengono utilizzati per impostare la regola in ingresso del gruppo di sicurezza dell'istanza Amazon EC2.		

Nome	Descrizione	Impostazione predefinita	Valori validi
SharedLoadBalancer	L'Amazon Resource Name (ARN) di un sistema di bilanciamento del carico condiviso. Questa opzione è rilevante solo per un sistema Application Load Balancer. È necessaria quando l'opzione `LoadBalancerIsShared` degli spazi dei nomi aws:elasticbeanstalk:environment è impostata su `true`. Non è possibile modificare l'ARN del sistema di bilanciamento del carico condiviso dopo la creazione dell'ambiente. Criteri per un valore valido: • Deve essere un load balancer valido e attivo nella AWS regione in cui si trova l'ambiente. • Deve trovarsi nello stesso Amazon Virtual Private Cloud (Amazon VPC) dell'ambiente. • Non può trattarsi di un sistema di bilanciamento del carico creato da Elastic Beanstalk come sistema di bilanciamento del carico dedicato per un altro ambiente. È possibile identificare questi sistemi di bilanciamento del carico dedicati utilizzando il prefisso `awseb-`.	Nessuna	ARN di un sistema di bilanciamento del carico valido che soddisfa tutti i criteri descritti di seguito.

Nome	Descrizione	Impostazione predefinita	Valori validi
	Esempio: `arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc`		

aws:rds:dbinstance

Configura un'istanza database Amazon RDS collegata.

`Spazio dei nomi: aws:rds:dbinstance`

Nome	Descrizione	Impostazione predefinita	Valori validi
DBAllocatedStorage	Le dimensioni di storage di database allocate, in gigabyte.	MySQL: `5` Oracle: `10` sqlserver-se: `200` sqlserver-ex: `30` sqlserver-web: `30`	MySQL: `5-1024` Oracle: `10-1024` sqlserver: non modificabile.

DBDeletionPolicy	Specifica se conservare, eliminare o creare uno snapshot dell'istanza database alla terminazione dell'ambiente.	`Delete`	`Delete`
			`Retain`
			`Snapshot`

Nome	Descrizione	Impostazione predefinita	Valori validi
	Questa opzione funziona in combinazione con `HasCoupledDatabase` , anche un'opzione di questo spazio dei nomi. **Warning** L'eliminazione di un'istanza database comporta una perdita di dati permanente.		

DBEngine	Il nome del motore di database da utilizzare per questa istanza.	`mysql`	`mysql`
			
			`oracle-se1`
			
			`sqlserver-ex`
			
			`sqlserver-web`
			
			`sqlserver-se`
			
			`postgres`

DBEngineVersion	Il numero di versione del motore di database.	`5.5`	

Nome	Descrizione	Impostazione predefinita	Valori validi
DBInstanceClass	Il tipo di istanza database.	`db.t2.micro` (`db.m1.large` per un ambiente non in esecuzione in Amazon VPC)	Per ulteriori informazioni, consulta Classe di istanza database nella Guida per l'utente di Amazon Relational Database Service.
DBPassword	Il nome della password utente master per l'istanza database.	Nessuna	
DBSnapshotIdentifier	L'identificatore dello snapshot DB da cui eseguire il ripristino.	Nessuna	
DBUser	Il nome dell'utente master per l'istanza database.	ebroot	

Nome	Descrizione	Impostazione predefinita	Valori validi

HasCoupledDatabase	Specifica se un'istanza database è accoppiata all'ambiente. Se è stato attivato `true`, Elastic Beanstalk crea una nuova istanza database accoppiata al tuo ambiente. Se è stato attivato `false`, Elastic Beanstalk inizia a disaccoppiare l'istanza database dall'ambiente.
	
	Questa opzione funziona in combinazione con `DBDeletionPolicy` , anche un'opzione di questo spazio dei nomi.
	
	Note
	Nota: se si torna nuovamente a questo valore `true` dopo aver disaccoppiato il database precedente, Elastic Beanstalk crea un nuovo database con le impostazioni precedenti delle opzioni del database. Tuttavia, per mantenere la sicurezza dell'ambiente, non mantiene le impostazioni esistenti di `DBUser` e `DBPassword` . È necessario specificare di nuovo `DBUser` e `DBPassword` .
		`false`	`true`
			
			`false`

Nome	Descrizione	Impostazione predefinita	Valori validi
MultiAZDatabase	Speciifica se è necessario creare una distribuzione di istanze di database. Multi-AZ Per ulteriori informazioni sulle Multi-AZ distribuzioni con Amazon Relational Database Service (RDS), consulta Regioni e zone di disponibilità nella Amazon Relational Database Service User Guide.	`false`	`true` `false`

Opzioni specifiche della piattaforma

Alcune piattaforme Elastic Beanstalk definiscono gli spazi dei nomi delle opzioni specifici della piattaforma. Questi spazi dei nomi e le relative opzioni sono elencati di seguito per ogni piattaforma.

> **Note**
>
> In precedenza, nelle versioni della piattaforma basate su AMI Amazon Linux (in precedenza, Amazon Linux 2), le seguenti due funzionalità e i rispettivi spazi dei nomi erano considerati caratteristiche specifiche della piattaforma e sono stati elencati qui per piattaforma:
>
> - Configurazione proxy per file statici – `aws:elasticbeanstalk:environment:proxy:staticfiles`
> - AWS X-Ray supporto — `aws:elasticbeanstalk:xray`
>
> Nelle versioni della piattaforma Amazon Linux 2, Elastic Beanstalk implementa queste caratteristiche in modo coerente su tutte le piattaforme di supporto. Lo spazio dei nomi correlato è ora elencato nella pagina the section called “Opzioni generali”. Abbiamo tenuto menzione di loro solo in questa pagina per le piattaforme che avevano spazi dei nomi diversi.

Piattaforme

- Opzioni della piattaforma Docker
- Opzioni della piattaforma Go
- Opzioni della piattaforma Java SE

- Java con le opzioni di piattaforma Tomcat
- Opzioni della piattaforma .NET Core su Linux
- Opzioni della piattaforma .NET
- Opzioni della piattaforma Node.js
- Opzioni della piattaforma PHP
- Opzioni della piattaforma Python
- Opzioni della piattaforma Ruby

Opzioni della piattaforma Docker

Le seguenti opzioni di configurazione specifiche per Docker si applicano a Docker e alle piattaforme Docker preconfigurate.

> **Note**
>
> Queste opzioni di configurazione non si applicano:
>
> - Piattaforma Docker (Amazon Linux 2) con Docker Compose
> - Piattaforma Docker Multicontainer (AMI Amazon Linux AL1): questa piattaforma è stata ritirata

Spazio dei nomi: **`aws:elasticbeanstalk:environment:proxy`**

Nome	Descrizione	Impostazione predefinita	Valori validi
ProxyServer	Specifica il server Web da usare come proxy.	`nginx`	`nginx` `none`— Solo Amazon Linux AM e Docker w/ DC

Opzioni della piattaforma Go

Opzioni della piattaforma AMI Amazon Linux (prima di Amazon Linux 2)

Spazio dei nomi: **`aws:elasticbeanstalk:container:golang:staticfiles`**

Puoi utilizzare il namespace seguente per configurare il server proxy per l'elaborazione dei file statici. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione. In questo modo viene ridotta la quantità di richieste eseguibili dall'applicazione.

Effettua il mapping di un percorso servito dal server proxy su una cartella nel codice sorgente contenente asset statici. Ciascuna opzione da te definita in questo namespace mappa un percorso differente.

Nome	Valore
Percorso in cui il server proxy servirà i file. Esempio: `/images` per servire i file su *`subdomain`* `.eleasticbeanstalk.com/images` .	Nome della cartella contenente i file. Esempio: `staticimages` per servire i file da una cartella denominata `staticimages` al primo livello del tuo bundle di origine.

Opzioni della piattaforma Java SE

Opzioni della piattaforma AMI Amazon Linux (prima di Amazon Linux 2)

Spazio dei nomi: **`aws:elasticbeanstalk:container:java:staticfiles`**

Puoi utilizzare il namespace seguente per configurare il server proxy per l'elaborazione dei file statici. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione. In questo modo viene ridotta la quantità di richieste eseguibili dall'applicazione.

Effettua il mapping di un percorso servito dal server proxy su una cartella nel codice sorgente contenente asset statici. Ciascuna opzione da te definita in questo namespace mappa un percorso differente.

Nome	Valore
Percorso in cui il server proxy servirà i file. Esempio: `/images` per servire i file su *`subdomain`* `.eleasticbeanstalk.com/images` .	Nome della cartella contenente i file. Esempio: `staticimages` per servire i file da una cartella denominata `staticimages` al primo livello del tuo bundle di origine.

Java con le opzioni di piattaforma Tomcat

Spazio dei nomi: **`aws:elasticbeanstalk:application:environment`**

Nome	Descrizione	Impostazione predefinita	Valori validi
JDBC_CONNECTION_STRING	La stringa di connessione a un database esterno.	n/d	n/a

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazio dei nomi: **`aws:elasticbeanstalk:container:tomcat:jvmoptions`**

Nome	Descrizione	Impostazione predefinita	Valori validi
JVM Options	Passa le opzioni della riga di comando a JVM all'avvio.	n/d	n/a
Xmx	Dimensioni heap massime di JVM.	`256m`	n/d
XX:MaxPermSize	Sezione dell'heap JVM utilizzata per memorizzare le definizioni di classi e i metadati associati.	`64m`	N/A

Nome	Descrizione	Impostazione predefinita	Valori validi
	Note Questa opzione si applica solo alle versioni Java precedenti a Java 8 e non è supportata sulle piattaforme Elastic Beanstalk Tomcat basate su Amazon Linux 2 e successive.		
Xms	Dimensioni heap iniziali di JVM.	`256m`	N/A
`optionName`	Specificare le opzioni JVM arbitrarie che andranno ad aggiungersi a quelle definite dalla piattaforma Tomcat.	n/d	n/a

Spazio dei nomi: **`aws:elasticbeanstalk:environment:proxy`**

Nome	Descrizione	Impostazione predefinita	Valori validi
GzipCompression	Imposta su `false` per disattivare la compressione di risposta. Valido solo nelle versioni della piattaforma AMI Amazon Linux (prima di Amazon Linux 2).	`true`	`true` `false`

ProxyServer	Imposta il proxy da usare nelle istanze dell'ambiente. Se imposti questa opzione su `apache`, Elastic Beanstalk utilizza Apache 2.4.	`nginx` (Amazon Linux 2)
		`apache` (AMI
			`apache`
			`apache/2.2` – Solo AMI Amazon Linux
			`nginx`

Nome	Descrizione	Impostazione predefinita	Valori validi
	Imposta l'opzione su `apache/2.2` se l'applicazione non è pronta per la migrazione da Apache 2.2 a causa di impostazioni di configurazione del proxy non compatibili. Questo valore è valido solo nelle versioni della piattaforma AMI Amazon Linux (prima di Amazon Linux 2). Imposta l'opzione su `nginx` per usare nginx. Questa è l'impostazione predefinita a partire dalle versioni della piattaforma Amazon Linux 2. Per ulteriori informazioni, consultare Configurazione del server proxy.	Amazon Linux)	

Opzioni della piattaforma .NET Core su Linux

Spazio dei nomi: **`aws:elasticbeanstalk:environment:proxy`**

Nome	Descrizione	Impostazione predefinita	Valori validi

ProxyServer	Specifica il server Web da usare come proxy.	`nginx`	`nginx`
			
			`none`

Opzioni della piattaforma .NET

Spazio dei nomi: **`aws:elasticbeanstalk:container:dotnet:apppool`**

Nome	Descrizione	Impostazione predefinita	Valori validi
Target Runtime	Scegli la versione di .NET Framework per la tua applicazione.	`4.0`	`2.0` `4.0`
Enable 32-bit Applications	Imposta su `True` per eseguire applicazioni a 32 bit.	`False`	`True` `False`

Opzioni della piattaforma Node.js

Spazio dei nomi: **`aws:elasticbeanstalk:environment:proxy`**

Nome	Descrizione	Impostazione predefinita	Valori validi
ProxyServer	Imposta il proxy da usare nelle istanze dell'ambiente.	`nginx`	`apache` `nginx`

Opzioni della piattaforma AMI Amazon Linux (prima di Amazon Linux 2)

Spazio dei nomi: **`aws:elasticbeanstalk:container:nodejs`**

Nome	Descrizione	Impostazione predefinita	Valori validi
NodeCommand	Comando utilizzato per avviare l'applicazione Node.js. Se viene specificata una stringa vuota, viene utilizzato `app.js`, quindi `server.js` e `npm start` in questo ordine.	""	n/d
NodeVersion	Versione di Node.js. Ad esempio, `4.4.6` Le versioni Node.js supportate variano tra le versioni della piattaforma Node.js. Consulta Node.js nel documento relativo alle piattaforme di AWS Elastic Beanstalk per ottenere un elenco delle versioni attualmente supportate. **Note** Quando il supporto della versione di Node.js in uso viene rimosso dalla piattaforma, è necessario modificare o rimuovere l'impostazione della versione prima di eseguire un aggiornamento della piattaforma. Questo può verificarsi quando viene rilevata una vulnerabilità di sicurezza per una o più versioni di Node.js Quando ciò accade, il tentativo di aggiornamento a una nuova	varia	varia

Nome	Descrizione	Impostazione predefinita	Valori validi
	versione della piattaforma che non supporta quella configurata NodeVersionfallisce. Per evitare di dover creare un nuovo ambiente, modificate l'opzione di NodeVersionconfigurazione impostando una versione di Node.js supportata sia dalla vecchia versione della piattaforma che da quella nuova, oppure rimuovete l'impostazione dell'opzione e quindi eseguite l'aggiornamento della piattaforma.		
GzipCompression	Specifica se è abilitata la compressione gzip. Se ProxyServer è impostata sunone, la compressione gzip è disabilitata.	`false`	`true` `false`

ProxyServer	Specifica il server Web da utilizzare per le connessioni proxy a Node.js. Se ProxyServer è impostato sunone, le mappature statiche dei file non hanno effetto e la compressione gzip è disabilitata.	`nginx`	`apache`
			`nginx`
			`none`

Spazio dei nomi: **`aws:elasticbeanstalk:container:nodejs:staticfiles`**

Puoi utilizzare il namespace seguente per configurare il server proxy per l'elaborazione dei file statici. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione. In questo modo viene ridotta la quantità di richieste eseguibili dall'applicazione.

Effettua il mapping di un percorso servito dal server proxy su una cartella nel codice sorgente contenente asset statici. Ciascuna opzione da te definita in questo namespace mappa un percorso differente.

> **Note**
>
> Le impostazioni di file statici non si applicano se `aws:elasticbeanstalk:container:nodejs::ProxyFiles` è impostato su `none`.

Nome	Valore
Percorso in cui il server proxy servirà i file. Esempio: `/images` per servire i file su *`subdomain`* `.eleasticbeanstalk.com/images` .	Nome della cartella contenente i file. Esempio: `staticimages` per servire i file da una cartella denominata `staticimages` al primo livello del tuo bundle di origine.

Opzioni della piattaforma PHP

Spazio dei nomi: **`aws:elasticbeanstalk:container:php:phpini`**

Nome	Descrizione	Impostazione predefinita	Valori validi
document_root	Specifica la directory figlio del tuo progetto trattata come la radice Web rivolta al pubblico.	`/`	Una stringa vuota viene trattata come `/`oppure specifica una stringa che inizia con `/`
memory_limit	Quantità di memoria allocata per l'ambiente PHP.	`256M`	n/d
zlib.output_compression	Specifica se PHP deve utilizzare la compressione per l'output.	`Off`	`On`

Nome	Descrizione	Impostazione predefinita	Valori validi

			`Off`
			`true`
			`false`

allow_url_fopen	Specifica se alle funzioni file di PHP è consentito recuperare i dati da sedi remote, come siti web o server FTP.	`On`	`On`
			`Off`
			`true`
			`false`

display_errors	Specifica se i messaggi di errore devono essere parte dell'output.	`Off`	`On` `Off`
max_execution_time	Imposta il tempo massimo, in secondi, durante il quale a uno script è consentita l'esecuzione prima di essere terminato dall'ambiente.	`60`	`0` a `9223372036854775807` (PHP_INT_MAX)
composer_options	Imposta le opzioni personalizzate da utilizzare durante l'installazione delle dipendenze utilizzando Composer tramite il comando. composer.phar install Per ulteriori informazioni, consulta install sul sito getcomposer.org.	N/A	n/a

Spazio dei nomi: **`aws:elasticbeanstalk:environment:proxy`**

Nome	Descrizione	Impostazione predefinita	Valori validi
ProxyServer	Imposta il proxy da usare nelle istanze dell'ambiente.	`nginx`	`apache` `nginx`

> **Note**
>
> Per ulteriori informazioni sulla piattaforma PHP, consulta Utilizzo della piattaforma PHP di Elastic Beanstalk.

Opzioni della piattaforma Python

Spazio dei nomi: **`aws:elasticbeanstalk:application:environment`**

Nome	Descrizione	Impostazione predefinita	Valori validi
DJANGO_SETTINGS_MODULE	Specifica il file delle impostazioni da utilizzare.	n/d	n/a

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazio dei nomi: **`aws:elasticbeanstalk:container:python`**

Nome	Descrizione	Impostazione predefinita	Valori validi
WSGIPath	Il file che contiene l'applicazione WSGI. Questo file deve disporre di una `application` chiamabile.	Sulle versioni	n/d

Nome	Descrizione	Impostazione predefinita	Valori validi
		della piattaforma Python di Amazon Linux 2: `application` Sulle versioni della piattaforma Python di AMI Amazon Linux: `application.py`	
NumProcesses	Il numero di processi daemon che devono essere avviati per il gruppo di processi quando si eseguono le applicazioni WSGI.	1	n/d
NumThreads	Il numero di thread creati per gestire le richieste in ciascun processo daemon all'interno del gruppo di processi quando si eseguono le applicazioni WSGI.	15	n/d

Spazio dei nomi: **`aws:elasticbeanstalk:environment:proxy`**

Nome	Descrizione	Impostazione predefinita	Valori validi
ProxyServer	Imposta il proxy da usare nelle istanze dell'ambiente.	`nginx`	`apache` `nginx`

Opzioni della piattaforma AMI Amazon Linux (prima di Amazon Linux 2)

Spazio dei nomi: **`aws:elasticbeanstalk:container:python:staticfiles`**

Puoi utilizzare il namespace seguente per configurare il server proxy per l'elaborazione dei file statici. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione. In questo modo viene ridotta la quantità di richieste eseguibili dall'applicazione.

Effettua il mapping di un percorso servito dal server proxy su una cartella nel codice sorgente contenente asset statici. Ciascuna opzione da te definita in questo namespace mappa un percorso differente.

Per impostazione predefinita, il server proxy in un ambiente Python serve qualsiasi file in una cartella denominata `static` nel percorso `/static`.

Spazio dei nomi: **`aws:elasticbeanstalk:container:python:staticfiles`**

Nome	Valore
Percorso in cui il server proxy servirà i file.	Nome della cartella contenente i file.
Esempio: `/images` per servire i file su *`subdomain`* `.eleasticbeanstalk.com/images` .	Esempio: `staticimages` per servire i file da una cartella denominata `staticimages` al primo livello del tuo bundle di origine.

Opzioni della piattaforma Ruby

Spazio dei nomi: **`aws:elasticbeanstalk:application:environment`**

Nome	Descrizione	Impostazione predefinita	Valori validi
RAILS_SKIP_MIGRATIONS	Specifica se eseguire `` `rake db:migrate` `` per conto delle applicazioni degli utenti, oppure se deve essere ignorato. Questo è valido solo per le applicazioni Rails 3.	`false`	`true` `false`
RAILS_SKIP_ASSET_COMPILATION	Specifica se il container deve eseguire `` `rake assets:precompile` `` per conto delle applicazioni degli utenti, oppure se deve essere ignorato. Questo è anche valido solo per le applicazioni Rails 3.	`false`	`true` `false`
BUNDLE_WITHOUT	Un elenco di gruppi separati da due punti (`:`) da ignorare quando si installano dipendenze da Gemfile.	`test:development`	n/d
RACK_ENV	Specifica quale fase di un ambiente può essere eseguita un'applicazione. Tra gli ambienti comuni figurano lo sviluppo, la produzione e il test.	`production`	n/d

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Opzioni personalizzate

Utilizza il namespace `aws:elasticbeanstalk:customoption` per definire opzioni e valori che possono essere letti nei blocchi `Resources` in altri file di configurazione. Usa opzioni personalizzate per raccogliere le impostazioni specificate dall'utente in un singolo file di configurazione.

Ad esempio, è possibile avere un file di configurazione complesso per definire una risorsa che può essere configurata dall'utente all'avvio dell'ambiente. Se utilizzi `Fn::GetOptionSetting` per

recuperare il valore di un'opzione personalizzata, è possibile inserire la definizione di tale opzione in un file di configurazione differente, dove può essere più facilmente individuata e modificata dall'utente.

Inoltre, poiché si tratta di opzioni di configurazione, le opzioni personalizzate possono essere configurate a livello di API per sostituire i valori definiti in un file di configurazione. Consulta Priorità per ulteriori informazioni.

Le opzioni personalizzate sono definite come qualsiasi altra opzione:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    option name: option value
```

Ad esempio, il seguente file di configurazione crea un'opzione denominata `ELBAlarmEmail` e imposta il valore su `someone@example.com`:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    ELBAlarmEmail: someone@example.com
```

In altre sezioni, un file di configurazione definisce un argomento SNS che legge l'opzione con `Fn::GetOptionSetting` per compilare il valore dell'attributo `Endpoint`:

```
Resources:
  MySNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: ELBAlarmEmail
              DefaultValue: nobody@example.com
          Protocol: email
```

Puoi trovare ulteriori frammenti di esempio usando `Fn::GetOptionSetting` in Aggiunta e personalizzazione delle risorse dell'ambiente Elastic Beanstalk.

Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`)

È possibile aggiungere file di AWS Elastic Beanstalk configurazione (`.ebextensions`) al codice sorgente dell'applicazione Web per configurare l'ambiente e personalizzare le AWS risorse in esso contenute. I file di configurazione sono documenti in formato YAML o JSON con un'estensione di `.config` file che vengono inseriti in una cartella denominata `.ebextensions` e distribuiti nel bundle dei sorgenti dell'applicazione.

Example.ebextensions/ .config network-load-balancer

Questo esempio apporta una semplice modifica alla configurazione. Modifica un'opzione di configurazione per impostare il tipo del sistema di bilanciamento del carico dell'ambiente su Network Load Balancer.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

Ti consigliamo di usare il formato YAML per i file di configurazione, perché è più leggibile rispetto a JSON. YAML supporta commenti, comandi multi-riga, diverse alternative per l'utilizzo di virgolette e molto altro. Puoi comunque effettuare qualsiasi modifica nei file di configurazione Elastic Beanstalk in modo identico utilizzando YAML o JSON.

> **Suggerimento**
>
> Quando sviluppi o esegui il test di nuovi i file di configurazione, avvia un ambiente pulito che esegue l'applicazione predefinita e distribuisci in questo ambiente. I file di configurazione con formattazione insufficiente determinano un errore irreversibile di avvio dell'ambiente.

La sezione `option_settings` di un file di configurazione definisce i valori per le opzioni di configurazione. Le opzioni di configurazione consentono di configurare l'ambiente Elastic Beanstalk AWS , le risorse in esso contenute e il software che esegue l'applicazione. I file di configurazione sono solo uno dei diversi modi per impostare le opzioni di configurazione.

La `Resources`sezione consente di personalizzare ulteriormente le risorse nell'ambiente dell'applicazione e definire AWS risorse aggiuntive oltre alle funzionalità fornite dalle opzioni di

configurazione. Puoi aggiungere e configurare tutte le risorse supportate da AWS CloudFormation, utilizzate da Elastic Beanstalk per creare ambienti.

Le altre sezioni di un file di configurazione (`packages`,,`sources`,`files`,`users`, `groups` `commandscontainer_commands`, `eservices`) consentono di configurare le EC2 istanze che vengono avviate nel proprio ambiente. Ogni volta che un server viene lanciato nel tuo ambiente, Elastic Beanstalk esegue le operazioni definite in queste sezioni per preparare il sistema operativo e il sistema di storage per la tua applicazione.

Per esempi di .ebextensions uso comune, vedere il repository dei file di configurazione di Elastic Beanstalk.

Requisiti

- Posizione: Elastic Beanstalk `.ebextensions` elaborerà tutte le cartelle presenti nella distribuzione. Tuttavia, ti consigliamo di inserire tutti i file di configurazione in un'unica cartella, denominata`.ebextensions`, nella radice del pacchetto sorgente. Le cartelle che iniziano con un punto possono essere nascoste dai browser di file, pertanto accertati che la cartella venga aggiunta al momento della creazione del bundle di origine. Per ulteriori informazioni, consulta Crea un bundle di sorgenti dell'applicazione Elastic Beanstalk.
- Denominazione: i file di configurazione devono avere l'estensione file `.config`.
- Formattazione: i file di configurazione devono essere conformi alle specifiche YAML o JSON.

  Con il formato YAML utilizza sempre gli spazi per applicare il rientro alle chiavi a diversi livelli di nidificazione. Per ulteriori informazioni su YAML, vedi YAML Ain't Markup Language (YAML™) Version 1.1.
- Unicità: utilizza ciascuna chiave solo una volta in ogni file di configurazione.

> **Attenzione**
>
> Se una chiave (ad esempio, `option_settings`) viene utilizzata due volte nello stesso file di configurazione, una delle sezioni verrà cancellata. Combina le sezioni duplicate in un'unica sezione o inseriscile in file di configurazione separati.

Il processo di distribuzione varia leggermente in funzione del client utilizzato per gestire gli ambienti. Per ulteriori informazioni, consulta le sezioni seguenti:

- Console Elastic Beanstalk

- CLI EB
- AWS CLI

Argomenti

- Impostazioni opzioni
- Personalizzazione di software su server Linux
- Personalizzazione di software su server Windows
- Aggiunta e personalizzazione delle risorse dell'ambiente Elastic Beanstalk

Impostazioni opzioni

Puoi utilizzare la chiave `option_settings` per modificare la configurazione Elastic Beanstalk e definire le variabili che possono essere recuperate dalla tua applicazione tramite variabili di ambiente. Alcuni spazi dei nomi consentono di estendere il numero di parametri e di specificare i relativi nomi. Per un elenco di namespace e di opzioni di configurazione, vedi Opzioni di configurazione.

Le impostazioni delle opzioni possono anche essere applicate direttamente a un ambiente durante la creazione o l'aggiornamento di un ambiente. Le impostazioni applicate direttamente all'ambiente sovrascrivono le impostazioni per le stesse opzioni nei file di configurazione. Se rimuovi le impostazioni dalla configurazione di un ambiente, le impostazioni nei file di configurazione saranno effettive. Per informazioni dettagliate, vedi Priorità.

Sintassi

La sintassi standard per le impostazioni delle opzioni è rappresentata da una gamma di oggetti, ognuno con `namespace`, `option_name` e una chiave `value`.

```
option_settings:
  - namespace:  namespace
    option_name:  option name
    value:  option value
  - namespace:  namespace
    option_name:  option name
    value:  option value
```

La chiave `namespace` è facoltativa. Se non specifichi uno spazio dei nomi, l'impostazione predefinita utilizzata è `aws:elasticbeanstalk:application:environment`:

```
option_settings:
  - option_name:  option name
    value:  option value
  - option_name:  option name
    value:  option value
```

Elastic Beanstalk supporta anche una sintassi abbreviata per le impostazioni di opzioni che consente di specificare le opzioni come coppie chiave-valore sotto lo spazio dei nomi:

```
option_settings:
  namespace:
    option name: option value
    option name: option value
```

Esempi

I seguenti esempi impostano un'opzione specifica per la piattaforma Tomcat nello spazio dei nomi `aws:elasticbeanstalk:container:tomcat:jvmoptions` e una proprietà dell'ambiente denominata `MYPARAMETER`.

In formato YAML standard:

Example.ebextensions/options.config

```
option_settings:
  - namespace:  aws:elasticbeanstalk:container:tomcat:jvmoptions
    option_name:  Xmx
    value:  256m
  - option_name: MYPARAMETER
    value: parametervalue
```

In formato abbreviato:

Example.ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xmx: 256m
  aws:elasticbeanstalk:application:environment:
    MYPARAMETER: parametervalue
```

In JSON:

Example.ebextensions/options.config

```
{
  "option_settings": [
    {
      "namespace": "aws:elasticbeanstalk:container:tomcat:jvmoptions",
      "option_name": "Xmx",
      "value": "256m"
    },
    {
      "option_name": "MYPARAMETER",
      "value": "parametervalue"
    }
  ]
}
```

Personalizzazione di software su server Linux

Questa sezione descrive il tipo di informazioni che puoi includere in un file di configurazione per personalizzare il software sulle istanze che EC2 eseguono Linux. Per informazioni generali sulla personalizzazione e la configurazione degli ambienti Elastic Beanstalk, consulta Configurazione degli ambienti Elastic Beanstalk. Per informazioni sulla personalizzazione del software sulle EC2 istanze che eseguono Windows, consulta. Personalizzazione di software su server Windows

È possibile che tu voglia personalizzare e configurare il software da cui dipende la tua applicazione. È possibile aggiungere comandi da eseguire durante il provisioning delle istanze, definire utenti e gruppi Linux e scaricare o creare direttamente file nelle istanze dell'ambiente. Questi file potrebbero essere dipendenze richieste dall'applicazione (ad esempio, pacchetti aggiuntivi dal repository yum) oppure file di configurazione, ad esempio file di configurazione come la sostituzione di un file di configurazione del proxy per sostituire impostazioni specifiche impostate in modo predefinito da Elastic Beanstalk).

> **Note**
>
> - Sulle piattaforme Amazon Linux 2, invece di fornire file e comandi nei file di configurazione con estensione .ebextensions, si consiglia di utilizzare Buildfile. Procfile e gli hook di piattaforma quando possibile per configurare ed eseguire codice personalizzato

sulle istanze dell'ambiente durante il provisioning delle istanze. Per i dettagli su questi meccanismi, consulta the section called “Estensione delle piattaforme Linux”.

- YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

I file di configurazione supportano le seguenti chiavi utilizzabili sul server Linux in cui è in esecuzione l'applicazione.

Chiavi

- Pacchetti
- Gruppi
- Utenti
- Origini
- File
- Comandi
- Servizi
- Comandi del container
- Esempio: utilizzo di CloudWatch parametri Amazon personalizzati

Le chiavi sono elaborate nell'ordine in cui sono elencate qui.

Controlla gli eventi del tuo ambiente durante lo sviluppo e il test dei file di configurazione. Elastic Beanstalk ignora un file di configurazione contenente errori di convalida, ad esempio una chiave non valida, e non elabora le altre chiavi contenute nello stesso file. In questi casi, Elastic Beanstalk aggiunge un avviso nel log degli eventi.

Pacchetti

Puoi utilizzare la chiave `packages` per scaricare e installare pacchetti di componenti e applicazioni.

Sintassi

```
packages:
  name of package manager:
    package name: version
```

```
    ...
  name of package manager:
    package name: version
    ...
  ...
```

È possibile specificare più pacchetti in ciascuna chiave del programma di gestione dei pacchetti.

Formati di pacchetto supportati

Elastic Beanstalk supporta attualmente i seguenti programmi di gestione dei pacchetti: yum, rubygems, python e rpm. I pacchetti sono elaborati nel seguente ordine: rpm, yum, quindi rubygems e python. Non vi è alcun ordine tra rubygems e Python. All'interno di ogni programma di gestione dei pacchetti, l'ordine di installazione non è garantito. Utilizza un programma di gestione dei pacchetti supportato dal sistema operativo in uso.

> **Note**
>
> Elastic Beanstalk supporta due programmi di gestione dei pacchetti sottostanti per Python, ovvero pip ed easy_install. Tuttavia, nella sintassi del file di configurazione, devi specificare `python` come nome del programma di gestione dei pacchetti. Quando si utilizza un file di configurazione per specificare un pacchetto Python manager, Elastic Beanstalk usa Python 2.7. Se la tua applicazione si basa su un'altra versione di Python, è possibile specificare i pacchetti da installare in un file `requirements.txt`. Per ulteriori informazioni, consulta Specificazione delle dipendenze utilizzando un file dei requisiti su Elastic Beanstalk.

Specificazione delle versioni

In ogni programma di gestione dei pacchetti, ogni pacchetto è specificato come nome di pacchetto ed elenco di versioni. La versione può essere una stringa, un elenco di versioni oppure una stringa o un elenco vuoto. Una stringa o un elenco vuoto indica che deve essere utilizzata la versione più recente. Per il programma rpm, la versione è specificata come percorso a un file su un disco o a un URL. I percorsi relativi non sono supportati.

Se specifichi una versione di un pacchetto, Elastic Beanstalk tenta di installare tale versione anche se una versione più recente del pacchetto è già installata sull'istanza. Se una versione più recente è già installata, la distribuzione non riesce. Solo alcuni programmi di gestione dei pacchetti supportano più versioni. Per ulteriori informazioni, consulta la documentazione del programma di gestione dei pacchetti utilizzato. Se non specifichi una versione e una versione del pacchetto è già installata,

Elastic Beanstalk non installa una nuova versione, in quanto presuppone che intendi continuare a utilizzare la versione esistente.

Frammento di esempio

Il frammento seguente specifica un URL di versione per rpm, richiede la versione più recente di yum e la versione 0.10.2 di chef da rubygems.

```
packages:
  yum:
    libmemcached: []
    ruby-devel: []
    gcc: []
  rpm:
    epel: http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
  rubygems:
    chef: '0.10.2'
```

Gruppi

È possibile utilizzare la `groups` chiave per creare Linux/UNIX gruppi e assegnare gruppi IDs. Per creare un gruppo, aggiungi una nuova coppia chiave-valore che esegue la mappatura di un nuovo nome di gruppo a un ID di gruppo facoltativo. La chiave "groups" può contenere uno o più nomi di gruppo. La tabella seguente elenca le chiavi disponibili.

Sintassi

```
groups:
  name of group: {}
  name of group:
    gid: "group id"
```

Opzioni

`gid`

Un numero di ID di gruppo.

Se viene specificato un ID di gruppo e il nome di gruppo esiste già, la creazione del gruppo non riuscirà. Se un altro gruppo è associato a l'ID di gruppo specificato, il sistema operativo può rifiutare la creazione del gruppo.

Frammento di esempio

Il seguente frammento specifica un gruppo denominato groupOne senza assegnare un ID di gruppo e un gruppo denominato groupTwo, il cui ID di gruppo è 45.

```
groups:
  groupOne: {}
  groupTwo:
    gid: "45"
```

Utenti

È possibile utilizzare la `users` chiave per creare Linux/UNIX utenti sull' EC2 istanza.

Sintassi

```
users:
  name of user:
    groups:
      - name of group
    uid: "id of the user"
    homeDir: "user's home directory"
```

Opzioni

`uid`

Un ID utente. Il processo di creazione non riesce se esiste un nome utente con un ID utente differente. Se l'ID utente è già assegnato a un utente esistente, il sistema operativo può respingere la richiesta di creazione.

`groups`

Un elenco di nomi di gruppo. L'utente viene aggiunto a ogni gruppo nell'elenco.

`homeDir`

La home directory dell'utente.

Gli utenti vengono creati come utenti di sistema non interattivo con una shell `/sbin/nologin`. Questa è un'impostazione predefinita e non può essere modificata.

Frammento di esempio

```
users:
  myuser:
    groups:
      - group1
      - group2
    uid: "50"
    homeDir: "/tmp"
```

Origini

È possibile utilizzare la `sources` chiave per scaricare un file di archivio da un URL pubblico e decomprimerlo in una directory di destinazione sull' EC2 istanza.

Sintassi

```
sources:
  target directory: location of archive file
```

Formati supportati

I formati supportati sono tar, tar+gzip, tar+bz2 e zip. Puoi fare riferimento a posizioni esterne, come Amazon Simple Storage Service (Amazon S3) (ad esempio, `https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject`), a condizione che l'URL sia pubblicamente accessibile.

Frammento di esempio

L'esempio seguente scarica un file .zip pubblico da un bucket Amazon S3 e lo decomprime in `/etc/myapp`:

```
sources:
  /etc/myapp: https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject
```

> **Note**
>
> Lo stesso percorso di destinazione non può essere riutilizzato da più estrazioni. L'estrazione di un'altra origine per lo stesso percorso di destinazione sostituirà i contenuti invece di accodarli.

File

È possibile utilizzare la `files` chiave per creare file sull' EC2 istanza. Il contenuto può essere inline nel file di configurazione o essere estratto da un URL. I file sono scritti su disco in ordine lessicografico.

Puoi utilizzare la chiave `files` per scaricare file privati da Amazon S3 fornendo un profilo dell'istanza per l'autorizzazione.

Se il percorso del file specificato esiste già nell'istanza, il file esistente viene mantenuto con l'estensione `.bak` aggiunta al suo nome.

Sintassi

```
files:
  "target file location on disk":
     mode: "six-digit octal value"
     owner: name of owning user for file
     group: name of owning group for file
     source: URL
     authentication: authentication name:

  "target file location on disk":
     mode: "six-digit octal value"
     owner: name of owning user for file
     group: name of owning group for file
     content: |
      # this is my
      # file content
     encoding: encoding format
     authentication: authentication name:
```

Opzioni

`content`

Contenuto della stringa da aggiungere al file. Specifica `content` o `source`, ma non entrambi.

`source`

L'URL di un file da scaricare. Specifica `content` o `source`, ma non entrambi.

`encoding`

Il formato di codifica della stringa specificata con l'opzione `content`.

Valori validi: `plain` | `base64`

`group`

Il gruppo Linux proprietario del file.

`owner`

L'utente Linux proprietario del file.

`mode`

Valore ottale a sei cifre che rappresenta la modalità per questo file. Questa chiave non è supportata per i sistemi Windows. Usa le prime tre cifre per i collegamenti simbolici e le ultime tre per impostare le autorizzazioni. Per creare un collegamento simbolico, specifica `120xxx`, dove `xxx` definisce le autorizzazioni del file di destinazione. Per specificare le autorizzazioni per un file, usa le ultime tre cifre, ad esempio `000644`.

`authentication`

Il nome di un metodo di autenticazione CloudFormation da utilizzare. Puoi aggiungere metodi di autenticazione ai metadati del gruppo Auto Scaling con la chiave Resources. Un esempio è fornito di seguito.

Frammento di esempio

```
files:
  "/home/ec2-user/myfile" :
    mode: "000755"
    owner: root
    group: root
    source: http://foo.bar/myfile

  "/home/ec2-user/myfile2" :
    mode: "000755"
    owner: root
    group: root
    content: |
      this is my
      file content
```

Esempio che utilizza un collegamento simbolico. Viene creato un collegamento `/tmp/myfile2.txt` che punta al file esistente `/tmp/myfile1.txt`.

```
files:
  "/tmp/myfile2.txt" :
    mode: "120400"
    content: "/tmp/myfile1.txt"
```

L'esempio seguente utilizza la chiave Resources per aggiungere un metodo di autenticazione denominato S3Auth e utilizzato per scaricare un file privato da un bucket Amazon S3:

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["amzn-s3-demo-bucket2"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"

files:
  "/tmp/data.json" :
    mode: "000755"
    owner: root
    group: root
    authentication: "S3Auth"
    source: https://elasticbeanstalk-us-west-2-123456789012.s3-us-west-2.amazonaws.com/
data.json
```

Comandi

È possibile utilizzare la `commands` chiave per eseguire comandi sull' EC2 istanza. I comandi vengono eseguiti prima che l'applicazione e il server Web vengano configurati e che il file di versione dell'applicazione venga estratto.

I comandi specificati vengono eseguiti come utente root ed elaborati in ordine alfabetico in base al nome. Per impostazione predefinita, i comandi vengono eseguiti nella directory principale. Per eseguire comandi da un'altra directory, utilizza l'opzione `cwd`.

Per risolvere i problemi emersi con i comandi, è possibile trovare il relativo output nei log delle istanze.

Sintassi

```
commands:
  command name:
    command: command to run
    cwd: working directory
    env:
      variable name: variable value
    test: conditions for command
    ignoreErrors: true
```

Opzioni

`command`

Un array (insieme di sequenze di blocchi con sintassi YAML) o una stringa che specifica il comando da eseguire. Alcune note importanti:

- Se utilizzi una stringa, non devi racchiuderla tra virgolette. Se utilizzi le virgolette, applica un carattere di escape alle occorrenze dello stesso tipo di virgolette.
- Se utilizzi una matrice, non è necessario utilizzare un carattere di escape negli spazi o racchiudere i parametri di comando tra virgolette. Ogni elemento della matrice è un singolo argomento del comando. Non utilizzare un array per specificare più comandi.

I seguenti esempi sono tutti equivalenti:

```
commands:
  command1:
    command: git commit -m "This is a comment."
  command2:
    command: "git commit -m \"This is a comment.\""
  command3:
    command: 'git commit -m "This is a comment."'
  command4:
    command:
      - git
      - commit
      - -m
      - This is a comment.
```

Per specificare più comandi, utilizza un blocco scalare letterale, come mostrato nel seguente esempio.

```
commands:
  command block:
    command: |
      git commit -m "This is a comment."
      git push
```

`env`

(Facoltativa) Imposta variabili di ambiente per il comando. Questa proprietà non aggiunge l'ambiente esistente ma lo sovrascrive.

`cwd`

(Facoltativa) La directory di lavoro. Se questa opzione non è specificata, i comandi vengono eseguiti dalla directory principale (/).

`test`

(Facoltativo) Un comando che deve restituire il valore `true` (codice di uscita 0) per consentire a Elastic Beanstalk di elaborare il comando, ad esempio uno script shell, contenuto nella chiave `command`.

`ignoreErrors`

(Facoltativa) Un valore booleano che determina se altri comandi devono essere eseguiti nel caso in cui il comando nella chiave `command` non riesca (restituisce un valore diverso da zero). Imposta questo valore su `true` per continuare a eseguire i comandi anche se il comando non riesce. Impostalo su `false` per interrompere l'esecuzione dei comandi se il comando non riesce. Il valore predefinito è `false`.

Frammento di esempio

L'esempio di frammento seguente esegue uno script python.

```
commands:
  python_install:
    command: myscript.py
    cwd: /home/ec2-user
    env:
```

```
      myvarname: myvarvalue
    test: "[ -x /usr/bin/python ]"
```

Servizi

Puoi utilizzare la chiave `services` per definire i servizi che devono essere avviati o arrestati all'avvio dell'istanza. La chiave `services` consente inoltre di specificare le dipendenze in origini, pacchetti e file, in modo che, se richiesto a causa del file installati, Elastic Beanstalk esegue il riavvio del servizio.

Sintassi

```
services:
  sysvinit:
    name of service:
      enabled: "true"
      ensureRunning: "true"
      files:
        - "file name"
      sources:
        - "directory"
      packages:
        name of package manager:
          "package name[: version]"
      commands:
        - "name of command"
```

Opzioni

`ensureRunning`

Imposta su `true` per continuare a eseguire il servizio dopo la chiusura di Elastic Beanstalk.

Imposta su `false` per continuare a eseguire il servizio dopo la chiusura di Elastic Beanstalk.

Ometti la chiave per non apportare modifiche allo stato del servizio.

`enabled`

Imposta su `true` per eseguire automaticamente il servizio all'avvio.

Imposta su `false` per non eseguire automaticamente il servizio all'avvio.

Ometti la chiave per non apportare modifiche a questa proprietà.

`files`

Un elenco di file. Se Elastic Beanstalk ne modifica uno direttamente tramite il blocco di file, il servizio viene riavviato.

`sources`

Un elenco di directory. Se Elastic Beanstalk espande un archivio in una di queste directory, il servizio viene riavviato.

`packages`

La mappatura del programma di gestione dei pacchetti a un elenco di nomi di pacchetto. Se Elastic Beanstalk installa o aggiorna uno di questi pacchetti, il servizio viene riavviato.

`commands`

Un elenco di nomi di comando. Se Elastic Beanstalk esegue il comando specificato, il servizio viene riavviato.

Frammento di esempio

Di seguito viene riportato una esempio di frammento:

```
services:
  sysvinit:
    myservice:
      enabled: true
      ensureRunning: true
```

Comandi del container

Puoi utilizzare la chiave `container_commands` per eseguire comandi che modificano il codice sorgente dell'applicazione. I comandi di contenitore vengono eseguiti dopo la configurazione dell'applicazione e del server Web e l'estrazione dell'archivio delle versioni dell'applicazione, ma prima della distribuzione della versione dell'applicazione. I comandi non di contenitore e altre operazioni di personalizzazione vengono eseguiti prima dell'estrazione del codice sorgente dell'applicazione.

I comandi specificati vengono eseguiti come utente root ed elaborati in ordine alfabetico in base al nome. I comandi di contenitore vengono eseguiti a partire dalla directory di gestione temporanea, dove viene estratto il codice sorgente prima di essere distribuito al server di applicazioni. Qualsiasi

modifica apportata al codice sorgente nella directory di gestione temporanea con un comando di contenitore verrà inclusa al momento della distribuzione del codice sorgente nella posizione finale.

> **Note**
>
> L'output dei comandi del container viene registrato nel log delle istanze `cfn-init-cmd.log`. Per ulteriori informazioni sul recupero e la visualizzazione dei log delle istanze, consulta Visualizzazione dei log dalle istanze Amazon. EC2

Puoi utilizzare `leader_only` per eseguire il comando solo su una singola istanza o puoi configurare un oggetto `test` per eseguire il comando esclusivamente quando un comando di test restituisce `true`. I comandi di contenitore "leader-only" vengono eseguiti solo durante la creazione dell'ambiente e le distribuzioni, mentre altri comandi e operazioni di personalizzazione di server sono eseguiti a ogni provisioning o aggiornamento di un'istanza. I comandi di contenitore "leader-only" non vengono eseguiti a causa di modifiche alla configurazione di avvio, ad esempio nel caso di modifica dell'ID AMI o del tipo di istanza.

Sintassi

```
container_commands:
  name of container_command:
    command: "command to run"
    leader_only: true
  name of container_command:
    command: "command to run"
```

Opzioni

`command`

Una stringa o una matrice di stringhe da eseguire.

`env`

(Facoltativa) Imposta le variabili di ambiente prima di eseguire il comando, ignorando qualsiasi valore esistente.

`cwd`

(Facoltativa) La directory di lavoro. Per impostazione predefinita, questa è la directory di gestione temporanea dell'applicazione decompressa.

`leader_only`

(Facoltativa) Esegui il comando solo su una singola istanza scelta da Elastic Beanstalk. I comandi di contenitore "leader-only" vengono eseguiti prima di altri comandi di contenitore. Un comando può essere "leader-only" o avere un oggetto `test`, ma non entrambe le cose (`leader_only` è prioritario).

`test`

(Facoltativa) Esegui un comando di test che deve restituire `true` per eseguire il comando di contenitore. Un comando può essere "leader-only" o avere un oggetto `test`, ma non entrambe le cose (`leader_only` è prioritario).

`ignoreErrors`

(Facoltativa) Esegui correttamente le distribuzioni se il comando di container restituisce un valore diverso da 0 (riuscito). Imposta su `true` per attivarla.

Frammento di esempio

Di seguito viene riportato una esempio di frammento.

```
container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  99customize:
    command: "scripts/customize.sh"
```

Esempio: utilizzo di CloudWatch parametri Amazon personalizzati

Questo argomento fornisce un esempio di configurazione che integra i parametri di Elastic Beanstalk CloudWatch con l'agente Amazon per piattaforme basate su Amazon Linux 2 e versioni successive. L'esempio di configurazione utilizza file e comandi in un file di configurazione. `.ebextensions`

Amazon CloudWatch è un servizio web che ti consente di monitorare, gestire e pubblicare diverse metriche, nonché di configurare azioni di allarme in base ai dati delle metriche. Puoi definire metriche

personalizzate per uso personale ed Elastic Beanstalk le invierà ad Amazon. CloudWatch Una volta che Amazon CloudWatch contiene le tue metriche personalizzate, puoi visualizzarle nella CloudWatch console Amazon.

L' CloudWatch agente Amazon

L' CloudWatch agente Amazon consente la raccolta di CloudWatch parametri e registri sia dalle EC2 istanze Amazon che dai server locali su tutti i sistemi operativi. L'agente supporta i parametri raccolti a livello di sistema. Supporta anche la raccolta di registri e parametri personalizzati dalle applicazioni o dai servizi. Per ulteriori informazioni sull' CloudWatch agente Amazon, consulta Collecting metrics and logs with the CloudWatch agent nella Amazon CloudWatch User Guide.

> **Note**
>
> Elastic Beanstalk Enhanced Health Reporting offre un supporto nativo per la pubblicazione di un'ampia gamma di metriche di istanze e ambienti su. CloudWatch Per informazioni dettagliate, vedi Pubblicazione di parametri Amazon CloudWatch personalizzati per un ambiente.

Argomenti

- File di configurazione .Ebextensions
- Permissions
- Visualizzazione delle metriche nella console CloudWatch

File di configurazione .Ebextensions

Questo esempio utilizza file e comandi in un file di configurazione .ebextensions per configurare ed eseguire CloudWatch l'agente Amazon sulla piattaforma Amazon Linux 2. L'agente è preconfezionato con Amazon Linux 2. Se si utilizza un sistema operativo diverso, potrebbero essere necessari ulteriori passaggi per l'installazione dell'agente. Per ulteriori informazioni, consulta Installazione dell' CloudWatch agente nella Amazon CloudWatch User Guide.

Per utilizzare questo esempio, salvalo in un file denominato `cloudwatch.config` in una directory denominata `.ebextensions` al livello principale della directory del progetto, quindi distribuisci l'applicazione tramite la console Elastic Beanstalk (includi la directory .ebextensions nel bundle di origine) o l'interfaccia a riga di comando EB.

Per ulteriori informazioni sui file di configurazione, consulta Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

In questo file sono presenti due sezioni:

- `files`: questa sezione aggiunge il file di configurazione dell'agente. Indica quali parametri e registri l'agente deve inviare ad Amazon. CloudWatch In questo esempio, viene inviato solo il parametro mem_used_percent. Per un elenco completo delle metriche a livello di sistema supportate dall' CloudWatch agente Amazon, consulta Metriche raccolte dall' CloudWatch agente nella Amazon CloudWatch User Guide.
- `container_commands`: questa sezione contiene il comando che avvia l'agente, che passa il file di configurazione come parametro. Per maggiori dettagli su `container_commands`, consultare Comandi del container.

.ebextensions/cloudwatch.config

```
files:
  "/opt/aws/amazon-cloudwatch-agent/bin/config.json":
    mode: "000600"
    owner: root
    group: root
    content: |
      {
        "agent": {
          "metrics_collection_interval": 60,
          "run_as_user": "root"
        },
        "metrics": {
          "namespace": "System/Linux",
          "append_dimensions": {
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
          },
          "metrics_collected": {
            "mem": {
              "measurement": [
                "mem_used_percent"
              ]
            }
          }
        }
      }
```

```
container_commands:
  start_cloudwatch_agent:
    command: /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a
 append-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
```

Permissions

Le istanze del tuo ambiente necessitano delle autorizzazioni IAM appropriate per pubblicare CloudWatch metriche Amazon personalizzate utilizzando l'agente Amazon. CloudWatch È possibile concedere le autorizzazioni alle istanze dell'ambiente aggiungendole al profilo dell'istanza dell'ambiente. Puoi aggiungere le autorizzazioni al profilo dell'istanza prima o dopo la distribuzione dell'applicazione.

Per concedere le autorizzazioni alla pubblicazione di metriche CloudWatch

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Scegliere il ruolo del profilo dell'istanza dell'ambiente. Per impostazione predefinita, al momento della creazione di un ambiente tramite la console Elastic Beanstalk o l'interfaccia a riga di comando EB, il ruolo è `aws-elasticbeanstalk-ec2-role`.
4. Scegli la scheda Autorizzazioni.
5. In Permissions Policies (Policy di autorizzazione), nella sezione Permissions (Autorizzazioni), scegliere Attach policies (Allega policy).
6. In Allega autorizzazioni, scegli la politica gestita AWS . CloudWatchAgentServerPolicy Quindi fare clic su Attach policy (Collega policy).

Per ulteriori informazioni sulla gestione delle policy, consulta l'argomento relativo all'utilizzo delle policy nella Guida per l'utente di IAM.

Visualizzazione delle metriche nella console CloudWatch

Dopo aver distribuito il file CloudWatch di configurazione nel tuo ambiente, controlla la CloudWatchconsole Amazon per visualizzare i tuoi parametri. Le metriche personalizzate si troveranno nel namespace. CWAgent

Per ulteriori informazioni, consulta Visualizzazione delle metriche disponibili nella Amazon CloudWatch User Guide.

Personalizzazione di software su server Windows

È possibile che tu voglia personalizzare e configurare il software da cui dipende la tua applicazione. Questi file potrebbero essere dipendenze richieste dall'applicazione, ad esempio servizi o pacchetti aggiuntivi che devono essere eseguiti. Per informazioni generali sulla personalizzazione e sulla configurazione degli ambienti Elastic Beanstalk, consulta Configurazione degli ambienti Elastic Beanstalk.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

I file di configurazione supportano le seguenti chiavi utilizzabili con il server Windows in cui è in esecuzione l'applicazione.

Chiavi

- Pacchetti
- Origini
- File
- Comandi
- Servizi
- Comandi del container

Le chiavi sono elaborate nell'ordine in cui sono elencate qui.

> **Note**
>
> Le versioni della piattaforma .NET (senza versione) meno recenti non sono in grado di elaborare i file di configurazione nell'ordine corretto. Ulteriori informazioni sono disponibili in Migrazione tra le versioni principali della piattaforma server Windows di Elastic Beanstalk.

Controlla gli eventi del tuo ambiente durante lo sviluppo e il test dei file di configurazione. Elastic Beanstalk ignora un file di configurazione contenente errori di convalida, ad esempio una chiave

non valida, e non elabora le altre chiavi contenute nello stesso file. In questi casi, Elastic Beanstalk aggiunge un avviso nel log degli eventi.

Pacchetti

Utilizza la chiave `packages` per scaricare e installare pacchetti di componenti e applicazioni.

Negli ambienti Windows, Elastic Beanstalk supporta il download e l'installazione di pacchetti MSI. Gli ambienti Linux supportano programmi di gestione dei pacchetti aggiuntivi. Per maggiori dettagli, consultare Pacchetti nella pagina Personalizzazione di software su server Linux.

Puoi fare riferimento a qualsiasi posizione esterna, come un oggetto Amazon Simple Storage Service (Amazon S3), a condizione che l'URL sia pubblicamente accessibile.

Se si specificano diversi `msi:` pacchetti, il loro ordine di installazione non è garantito.

Sintassi

Specifica un nome di tua scelta come nome del pacchetto e un URL per un percorso di file MSI come valore. È possibile specificare più pacchetti nella chiave `msi:`.

```
packages:
  msi:
    package name: package url
    ...
```

Esempi

L'esempio seguente specifica un URL per scaricare mysql da `https://dev.mysql.com/`.

```
packages:
  msi:
    mysql: https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-
net-8.0.11.msi
```

L'esempio seguente specifica un oggetto Amazon S3 come posizione del file MSI.

```
packages:
  msi:
    mymsi: https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject.msi
```

Origini

Utilizza la chiave `sources` per scaricare un file di archivio da un URL pubblico e decomprimerlo in una directory di destinazione sull'istanza EC2.

Sintassi

```
sources:
  target directory: location of archive file
```

Formati supportati

In ambienti Windows, Elastic Beanstalk supporta il formato ZIP. Gli ambienti Linux supportano formati aggiuntivi. Per maggiori dettagli, consultare Origini nella pagina Personalizzazione di software su server Linux.

Puoi fare riferimento a qualsiasi posizione esterna, come un oggetto Amazon Simple Storage Service (Amazon S3), a condizione che l'URL sia pubblicamente accessibile.

Esempio

L'esempio seguente scarica un file.zip pubblico da Amazon S3 e lo decomprime. `c:/myproject/myapp`

```
sources:
  "c:/myproject/myapp": https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject.zip
```

File

Utilizza la chiave `files` per creare file sull'istanza EC2. Il contenuto può essere inline nel file di configurazione o essere estratto da un URL. I file sono scritti su disco in ordine lessicografico. Per scaricare file privati da Amazon S3, fornisci un profilo dell'istanza per l'autorizzazione.

Sintassi

```
files:
  "target file location on disk":
    source: URL
    authentication: authentication name:
```

```
  "target file location on disk":
    content: |
      this is my content
    encoding: encoding format
```

Opzioni

`content`

(Facoltativa) Una stringa.

`source`

(Facoltativa) L'URL da cui il file è caricato. Questa opzione non può essere specificata con la chiave contenuto.

`encoding`

(Facoltativa) Il formato di codifica. Questa opzione è utilizzata solo per un valore di chiave contenuto fornito. Il valore predefinito è `plain`.

Valori validi: `plain` | `base64`

`authentication`

(Facoltativo) Il nome di un metodo di autenticazione CloudFormation da utilizzare. Puoi aggiungere metodi di autenticazione ai metadati del gruppo Auto Scaling con la chiave Resources.

Esempi

L'esempio seguente mostra due modi per fornire i contenuti di un file: da un URL o inline nel file di configurazione.

```
files:
  "c:\\targetdirectory\\targetfile.txt":
    source: http://foo.bar/myfile

  "c:/targetdirectory/targetfile.txt":
    content: |
      # this is my file
```

```
      # with content
```

> **Note**
>
> Se utilizzi una barra rovesciata (\) nel percorso di file, devi anteporvi un'altra barra rovesciata (il carattere escape) come mostrato nell'esempio precedente.

L'esempio seguente utilizza la chiave Resources per aggiungere un metodo di autenticazione denominato S3Auth e lo utilizza per scaricare un file privato da Amazon S3:

```
files:
  "c:\\targetdirectory\\targetfile.zip":
    source: https://elasticbeanstalk-us-east-2-123456789012.s3.amazonaws.com/prefix/
myfile.zip
    authentication: S3Auth

Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          s: ["amzn-s3-demo-bucket"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"
```

Comandi

Utilizza la chiave `commands` per eseguire comandi sull'istanza EC2. I comandi vengono elaborati in ordine alfabetico per nome e vengono eseguiti prima della configurazione dell'applicazione e del server Web e prima che il file della versione dell'applicazione venga estratto.

I comandi specificati sono eseguiti come utente amministratore.

Per risolvere i problemi emersi con i comandi, è possibile trovare il relativo output nei log delle istanze.

Sintassi

```
commands:
  command name:
    command: command to run
```

Opzioni

`command`

Una matrice o una stringa che specifica il comando da eseguire. Se utilizzi una matrice, non è necessario utilizzare un carattere di escape negli spazi o racchiudere i parametri di comando tra virgolette.

`cwd`

(Facoltativa) La directory di lavoro. Per impostazione predefinita, Elastic Beanstalk tenta di trovare la posizione della directory del tuo progetto. Se non la trova, utilizza `c:\Windows\System32` come percorso di default.

`env`

(Facoltativa) Imposta variabili di ambiente per il comando. Questa proprietà non aggiunge l'ambiente esistente ma lo sovrascrive.

`ignoreErrors`

(Facoltativa) Un valore booleano che determina se altri comandi devono essere eseguiti nel caso in cui il comando nella chiave `command` non riesca (restituisce un valore diverso da zero). Imposta questo valore su `true` per continuare a eseguire i comandi anche se il comando non riesce. Impostalo su `false` per interrompere l'esecuzione dei comandi se il comando non riesce. Il valore predefinito è `false`.

`test`

(Facoltativo) Un comando che deve restituire il valore `true` (codice di uscita 0) per consentire a Elastic Beanstalk di elaborare il comando contenuto nella chiave `command`.

`waitAfterCompletion`

(Facoltativa) L'intervallo in secondi tra il completamento del comando e l'esecuzione del comando successivo. Il valore predefinito è **`60`** secondi. Puoi anche specificare. **`forever`**

> **Important**
>
> I riavvii del sistema durante la distribuzione non sono supportati. Se il sistema si riavvia a seguito di un comando, l'inizializzazione dell'istanza avrà esito negativo, causando il fallimento della distribuzione.
> Come soluzione alternativa, è possibile utilizzarla `.ebextensions configuration`per pianificare i riavvii dopo il completamento della distribuzione.

Esempio

L'esempio seguente salva l'output del comando `set` nel file specificato. Se c'è un comando successivo, viene eseguito da Elastic Beanstalk subito dopo il completamento di questo comando. Se tale comando richiede un riavvio, Elastic Beanstalk riavvia l'istanza subito dopo il completamento del comando.

```
commands:
  test:
    command: set > c:\\myapp\\set.txt
    waitAfterCompletion: 0
```

Servizi

Utilizza la chiave `services` per definire i servizi che devono essere avviati o arrestati all'avvio dell'istanza. La chiave `services` consente inoltre di specificare le dipendenze in origini, pacchetti e file, di modo che se è necessario un riavvio a seguito dell'installazione dei file, Elastic Beanstalk esegue il riavvio del servizio.

Sintassi

```
services:
  windows:
    name of service:
      files:
        - "file name"
      sources:
        - "directory"
      packages:
        name of package manager:
            "package name[: version]"
```

```
      commands:
        - "name of command"
```

Opzioni

`ensureRunning`

(Facoltativo) Imposta su `true` per continuare a eseguire il servizio dopo la chiusura di Elastic Beanstalk.

Imposta su `false` per continuare a eseguire il servizio dopo la chiusura di Elastic Beanstalk.

Ometti la chiave per non apportare modifiche allo stato del servizio.

`enabled`

(Facoltativa) Imposta su `true` per eseguire automaticamente il servizio all'avvio.

Imposta su `false` per non eseguire automaticamente il servizio all'avvio.

Ometti la chiave per non apportare modifiche a questa proprietà.

`files`

Un elenco di file. Se Elastic Beanstalk ne modifica uno direttamente tramite il blocco di file, il servizio viene riavviato.

`sources`

Un elenco di directory. Se Elastic Beanstalk espande un archivio in una di queste directory, il servizio viene riavviato.

`packages`

La mappatura del programma di gestione dei pacchetti a un elenco di nomi di pacchetto. Se Elastic Beanstalk installa o aggiorna uno di questi pacchetti, il servizio viene riavviato.

`commands`

Un elenco di nomi di comando. Se Elastic Beanstalk esegue il comando specificato, il servizio viene riavviato.

Esempio

```
services:
```

```
  windows:
    myservice:
      enabled: true
      ensureRunning: true
```

Comandi del container

Utilizza la chiave `container_commands` per eseguire comandi che modificano il codice sorgente dell'applicazione. I comandi di contenitore vengono eseguiti dopo la configurazione dell'applicazione e del server Web e l'estrazione dell'archivio delle versioni dell'applicazione, ma prima della distribuzione della versione dell'applicazione. I comandi non di contenitore e altre operazioni di personalizzazione vengono eseguiti prima dell'estrazione del codice sorgente dell'applicazione.

I comandi di contenitore vengono eseguiti a partire dalla directory di gestione temporanea, dove viene estratto il codice sorgente prima di essere distribuito al server di applicazioni. Qualsiasi modifica apportata al codice sorgente nella directory di gestione temporanea con un comando di contenitore verrà inclusa al momento della distribuzione del codice sorgente nella posizione finale.

Per risolvere i problemi emersi con i comandi del container, è possibile trovare il relativo output nei log delle istanze.

Utilizza l'opzione `leader_only` per eseguire il comando solo su una singola istanza oppure configura un oggetto `test` per eseguire il comando esclusivamente quando un comando di test restituisce `true`. I comandi di contenitore "leader-only" vengono eseguiti solo durante la creazione dell'ambiente e le distribuzioni, mentre altri comandi e operazioni di personalizzazione di server sono eseguiti a ogni provisioning o aggiornamento di un'istanza. I comandi di contenitore "leader-only" non vengono eseguiti a causa di modifiche alla configurazione di avvio, ad esempio nel caso di modifica dell'ID AMI o del tipo di istanza.

Sintassi

```
container_commands:
  name of container_command:
    command: command to run
```

Opzioni

`command`

Una stringa o una matrice di stringhe da eseguire.

`env`

(Facoltativa) Imposta le variabili di ambiente prima di eseguire il comando, ignorando qualsiasi valore esistente.

`cwd`

(Facoltativa) La directory di lavoro. Per impostazione predefinita, questa è la directory di gestione temporanea dell'applicazione decompressa.

`leader_only`

(Facoltativa) Esegui il comando solo su una singola istanza scelta da Elastic Beanstalk. I comandi di contenitore "leader-only" vengono eseguiti prima di altri comandi di contenitore. Un comando può essere "leader-only" o avere un oggetto `test`, ma non entrambe le cose (`leader_only` è prioritario).

`test`

(Facoltativa) Esegui un comando di test che deve restituire `true` per eseguire il comando di contenitore. Un comando può essere "leader-only" o avere un oggetto `test`, ma non entrambe le cose (`leader_only` è prioritario).

`ignoreErrors`

(Facoltativa) Esegui correttamente le distribuzioni se il comando di container restituisce un valore diverso da 0 (riuscito). Imposta su `true` per attivarla.

`waitAfterCompletion`

(Facoltativa) L'intervallo in secondi tra il completamento del comando e l'esecuzione del comando successivo. Il valore predefinito è **`60`** secondi. È inoltre possibile specificare. **`forever`**

> **Important**
>
> I riavvii del sistema durante la distribuzione non sono supportati. Se il sistema si riavvia a seguito di un comando, l'inizializzazione dell'istanza avrà esito negativo, causando il fallimento della distribuzione.
> Come soluzione alternativa, è possibile utilizzarla `.ebextensions configuration`per pianificare i riavvii dopo il completamento della distribuzione.

Esempio

L'esempio seguente salva l'output del comando `set` nel file specificato. Elastic Beanstalk esegue il comando su un'istanza e riavvia l'istanza subito dopo il completamento del comando.

```
container_commands:
  foo:
    command: set > c:\\myapp\\set.txt
    leader_only: true
    waitAfterCompletion: 0
```

Aggiunta e personalizzazione delle risorse dell'ambiente Elastic Beanstalk

Potresti voler personalizzare le risorse dell'ambiente che fanno parte del tuo ambiente Elastic Beanstalk. Ad esempio, potresti voler aggiungere una coda Amazon SQS e un allarme sulla profondità della coda oppure potresti voler aggiungere un cluster Amazon. ElastiCache Includendo un file di configurazione nel bundle di origine, puoi personalizzare agevolmente il tuo ambiente in contemporanea con la distribuzione della versione dell'applicazione.

Puoi usare la `Resources` chiave in un file di configurazione per creare e personalizzare AWS le risorse nel tuo ambiente. Le risorse definite nei file di configurazione vengono aggiunte al CloudFormation modello utilizzato per avviare l'ambiente. Sono supportati tutti i tipi di CloudFormation risorse.

> **Note**
>
> Ogni volta che aggiungi una risorsa non gestita da Elastic Beanstalk, assicurati di aggiungere una policy utente con le autorizzazioni AWS Identity and Access Management appropriate per i tuoi utenti (IAM). Le policy utente gestite fornite da Elastic Beanstalk coprono solo le autorizzazioni per le risorse gestite da Elastic Beanstalk.

Ad esempio, il file di configurazione riportato di seguito consente di aggiungere un hook del ciclo di vita di Auto Scaling al gruppo Auto Scaling predefinito creato da Elastic Beanstalk:

`~/my-app/.ebextensions/as-hook.config`

```
Resources:
  hookrole:
    Type: AWS::IAM::Role
```

```
    Properties:
      AssumeRolePolicyDocument: {
               "Version": "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "autoscaling.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            }
      Policies: [ {
               "PolicyName": "SNS",
               "PolicyDocument": {
                      "Version": "2012-10-17",
                      "Statement": [{
                          "Effect": "Allow",
                          "Resource": "*",
                          "Action": [
                              "sqs:SendMessage",
                              "sqs:GetQueueUrl",
                              "sns:Publish"
                          ]
                        }
                      ]
                  }
               } ]
  hooktopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint: "my-email@example.com"
          Protocol: email
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
      LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
      NotificationTargetARN: { "Ref" : "hooktopic" }
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

In questo esempio vengono definite tre risorse: `hookrole`, `hooktopic` e `lifecyclehook`. Le prime due risorse sono un ruolo IAM, che concede ad Amazon EC2 Auto Scaling l'autorizzazione a

pubblicare messaggi su Amazon SNS, e un argomento SNS, che inoltra i messaggi del gruppo Auto Scaling a un indirizzo e-mail. Elastic Beanstalk crea queste risorse con le proprietà e i tipi specificati.

L'ultima risorsa, `lifecyclehook`, è l'hook del ciclo di vita stesso:

```
lifecyclehook:
  Type: AWS::AutoScaling::LifecycleHook
  Properties:
    AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
    LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
    NotificationTargetARN: { "Ref" : "hooktopic" }
    RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

La definizione dell'hook del ciclo di vita utilizza due funzioni per compilare i valori per le proprietà dell'hook. `{ "Ref" : "AWSEBAutoScalingGroup" }` recupera il nome del gruppo Auto Scaling creato da Elastic Beanstalk per l'ambiente. `AWSEBAutoScalingGroup` è uno dei nomi di risorsa standard fornito da Elastic Beanstalk.

Per `AWS::IAM::Role`, la risorsa `Ref` restituisce soltanto il nome del ruolo, non l'ARN. Per ottenere l'ARN del parametro `RoleARN`, puoi utilizzare invece un'altra funzione intrinseca, `Fn::GetAtt`, in grado di ottenere attributi da una risorsa. `RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }` ottiene l'attributo `Arn` dalla risorsa `hookrole`.

`{ "Ref" : "hooktopic" }` ottiene l'ARN dell'argomento Amazon SNS creato in precedenza nel file di configurazione. Il valore restituito da `Ref` varia in base al tipo di risorsa e può essere trovato nell'argomento della Guida per l' CloudFormation utente relativo al tipo di risorsa. AWS::SNS::Topic

Modifica delle risorse create da Elastic Beanstalk per l'ambiente

Le risorse create da Elastic Beanstalk per l'ambiente hanno dei nomi. È possibile utilizzare questi nomi per ottenere informazioni sulle risorse con una funzione o modificare le proprietà sulle risorse per personalizzarne il comportamento. Questo argomento descrive le AWS risorse utilizzate da Elastic Beanstalk nei diversi tipi di ambienti.

> **Note**
>
> L'argomento precedente Risorse personalizzate fornisce alcuni casi d'uso ed esempi per la personalizzazione delle risorse dell'ambiente. È inoltre possibile trovare ulteriori esempi di file di configurazione nell'argomento successivo Esempi di risorse personalizzate.

Gli ambienti di server Web hanno le risorse seguenti.

Ambienti del server Web

- `AWSEBAutoScalingGroup`(AWS::AutoScaling::AutoScalingGruppo): il gruppo Auto Scaling collegato all'ambiente.
- Una delle due risorse seguenti.
  - `AWSEBAutoScalingLaunchConfiguration`(AWS::AutoScaling::LaunchConfiguration) — La configurazione di avvio collegata al gruppo Auto Scaling dell'ambiente.
  - `AWSEBEC2LaunchTemplate`(AWS::EC2::LaunchTemplate) — Il modello di lancio di Amazon EC2 utilizzato dal gruppo Auto Scaling del tuo ambiente.

> **Note**
>
> Se l'ambiente utilizza funzionalità che richiedono modelli di avvio di Amazon EC2 e la policy utente non dispone delle autorizzazioni richieste, la creazione o l'aggiornamento dell'ambiente potrebbe non riuscire. Utilizza la policy utente gestita da AdministratorAccess- AWSElastic Beanstalk o aggiungi le autorizzazioni richieste alla tua politica personalizzata.

- `AWSEBEnvironmentName`(AWS::ElasticBeanstalk::Environment) — Il tuo ambiente.
- `AWSEBSecurityGroup`(AWS::EC2::SecurityGroup) — Il gruppo di sicurezza collegato al gruppo Auto Scaling.
- `AWSEBRDSDatabase`(AWS: :RDS:: DBInstance) — L'istanza database di Amazon RDS collegata al tuo ambiente (se applicabile).

In un ambiente con carico bilanciato, è possibile accedere a risorse aggiuntive correlate al load balancer. I Classic Load Balancer hanno una risorsa per il load balancer e una per il gruppo di sicurezza collegato. Application e Network Load Balancer dispongono di risorse aggiuntive per il listener di default, per le regole del listener e per il gruppo di destinazione del load balancer.

Ambienti con bilanciamento del carico

- `AWSEBLoadBalancer`(AWS::ElasticLoadBalancing::LoadBalancer) — Il sistema di bilanciamento del carico classico del tuo ambiente.
- `AWSEBV2LoadBalancer`(AWS::ElasticLoadBalancingV2::LoadBalancer) — L'applicazione o il sistema di bilanciamento del carico di rete dell'ambiente.

- `AWSEBLoadBalancerSecurityGroup`(AWS::EC2::SecurityGroup) — Solo in un Amazon Virtual Private Cloud (Amazon VPC) personalizzato, il nome del gruppo di sicurezza creato da Elastic Beanstalk per il bilanciamento del carico. In un VPC predefinito o EC2 classico, Elastic Load Balancing assegna un gruppo di sicurezza predefinito al load balancer.
- `AWSEBV2LoadBalancerListener`(AWS::ElasticLoadBalancingV2::Listener) — Un listener che consente al sistema di bilanciamento del carico di verificare le richieste di connessione e di inoltrarle a uno o più gruppi target.
- `AWSEBV2LoadBalancerListenerRule`(AWS::ElasticLoadBalancingV2::ListenerRule) — Definisce su quali richieste interviene un listener Elastic Load Balancing e l'azione che esegue.
- `AWSEBV2LoadBalancerTargetGroup`(AWS::ElasticLoadBalancingV2::TargetGroup) — Un gruppo target Elastic Load Balancing che indirizza le richieste verso una o più destinazioni registrate, come le istanze Amazon EC2.

Gli ambienti worker (operatore) dispongono di risorse per la coda SQS che memorizza nel buffer le richieste in entrata e una tabella Amazon DynamoDB che le istanze utilizzano per la scelta del leader.

Ambienti lavoratore

- `AWSEBWorkerQueue`(AWS::SQS::Queue) — La coda Amazon SQS da cui il demone estrae le richieste che devono essere elaborate.
- `AWSEBWorkerDeadLetterQueue`(AWS::SQS::Queue) — La coda Amazon SQS che memorizza i messaggi che non possono essere recapitati o che altrimenti non sono stati elaborati correttamente dal demone.
- `AWSEBWorkerCronLeaderRegistry`(AWS::DynamoDB::Table) — La tabella Amazon DynamoDB che è il registro interno utilizzato dal demone per le attività periodiche.

Altre chiavi CloudFormation del modello

Abbiamo già introdotto le chiavi dei file di configurazione CloudFormation come `Resourcesfiles`, `epackages`. Elastic Beanstalk aggiunge il contenuto dei file di configurazione CloudFormation al modello che supporta l'ambiente, in modo da poter CloudFormation utilizzare altre sezioni per eseguire attività avanzate nei file di configurazione.

Chiavi

- Parameters
- Output

- Mappature

Parameters

I parametri sono un'alternativa alle opzioni personalizzate di Elastic Beanstalk che puoi utilizzare per definire i valori che utilizzi in altre posizioni nei file di configurazione. Come le opzioni personalizzate, puoi utilizzare i parametri per raccogliere i valori configurabili dall'utente in un unico posto. A differenza delle opzioni personalizzate, non è possibile utilizzare l'API di Elastic Beanstalk per impostare i valori dei parametri e il numero di parametri che è possibile definire in un modello è limitato da. CloudFormation

Uno dei motivi per cui potresti voler utilizzare i parametri è fare in modo che i tuoi file di configurazione fungano anche da modelli. CloudFormation Se si utilizzano parametri anziché opzioni personalizzate, è possibile utilizzare il file di configurazione per creare la stessa risorsa in CloudFormation un proprio stack. Ad esempio, puoi avere un file di configurazione che aggiunge un file system Amazon EFS al tuo ambiente per il testing, quindi puoi utilizzare lo stesso file per creare un file system indipendente, non collegato al ciclo di vita del tuo ambiente, per l'uso in produzione.

Il seguente esempio mostra l'uso di parametri per la raccolta di valori configurabili dall'utente all'inizio di un file di configurazione.

Example L oadbalancer-accesslogs-existingbucket .config — Parametri

```
Parameters:
  bucket:
    Type: String
    Description: "Name of the Amazon S3 bucket in which to store load balancer logs"
    Default: "amzn-s3-demo-bucket"
  bucketprefix:
    Type: String
    Description: "Optional prefix. Can't start or end with a /, or contain the word
 AWSLogs"
    Default: ""
```

Output

Puoi utilizzare un blocco `Outputs` per esportare informazioni sulle risorse create in CloudFormation. È quindi possibile utilizzare la `Fn::ImportValue` funzione per inserire il valore in un CloudFormation modello esterno a Elastic Beanstalk.

L'esempio seguente crea un argomento Amazon SNS e ne esporta l'ARN in con il nome. CloudFormation `NotificationTopicArn`

Example sns-topic.config

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

Outputs:
  NotificationTopicArn:
    Description: Notification topic ARN
    Value: { "Ref" : "NotificationTopic" }
    Export:
      Name: NotificationTopicArn
```

In un file di configurazione per un ambiente diverso o in un CloudFormation modello esterno a Elastic Beanstalk, puoi `Fn::ImportValue` usare la funzione per ottenere l'ARN esportato. Questo esempio assegna il valore esportato a una proprietà dell'ambiente denominata `TOPIC_ARN`.

Example env.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    TOPIC_ARN: '`{ "Fn::ImportValue" : "NotificationTopicArn" }`'
```

Mappature

Puoi usare una mappatura per archiviare coppie chiave-valore organizzate per spazio dei nomi. Una mappatura può aiutarti a organizzare i valori che utilizzi in tutte le configurazioni o a modificare il valore di un parametro in base a un altro valore. Ad esempio, la seguente configurazione imposta il valore di un parametro ID dell'account in base alla regione corrente.

Example L .config — Mappature oadbalancer-accesslogs-newbucket

```
Mappings:
  Region2ELBAccountId:
    us-east-1:
      AccountId: "111122223333"
    us-west-2:
```

```
      AccountId: "444455556666"
    us-west-1:
      AccountId: "123456789012"
    eu-west-1:
      AccountId: "777788889999"
...
            Principal:
              AWS:
                ? "Fn::FindInMap"
                :
                  - Region2ELBAccountId
                  -
                    Ref: "AWS::Region"
                  - AccountId
```

Funzioni

È possibile utilizzare le funzioni nei file di configurazione per inserire i valori per le proprietà delle risorse con informazioni provenienti da altre risorse o dalle impostazioni delle opzioni di configurazione di Elastic Beanstalk. Elastic CloudFormation Beanstalk supporta `Ref` le funzioni `Fn::GetAtt` (`Fn::Join`,,) e una funzione specifica di Elastic Beanstalk,. `Fn::GetOptionSetting`

Funzioni

- Ref
- Fn:: GetAtt
- Fn::Join
- Fn:: GetOptionSetting

Ref

`Ref`Da utilizzare per recuperare la rappresentazione di stringa predefinita di una risorsa. AWS Il valore restituito da `Ref`dipende dal tipo di risorsa e talvolta dipende anche da altri fattori. Ad esempio, un gruppo di sicurezza (AWS::EC2::SecurityGroup) restituisce il nome o l'ID del gruppo di sicurezza, a seconda che il gruppo di sicurezza si trovi in un VPC Amazon Virtual Private Cloud (Amazon VPC) predefinito, EC2 classico o personalizzato.

```
{ "Ref" : "resource name" }
```

> **Note**
>
> Per dettagli su ciascun tipo di risorsa, inclusi i valori restituiti di `Ref`, consultare Informazioni di riferimento dei tipi di risorse AWS nella Guida per l'utente di CloudFormation .

Dall'Hook del ciclo di vita Auto Scaling di esempio:

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
```

Puoi anche usarlo `Ref` per recuperare il valore di un CloudFormation parametro definito altrove nello stesso file o in un file di configurazione diverso.

Fn:: GetAtt

Si usa `Fn::GetAtt` per recuperare il valore di un attributo su una AWS risorsa.

```
{ "Fn::GetAtt" : [ "resource name", "attribute name"] }
```

Dall'Hook del ciclo di vita Auto Scaling di esempio:

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

Vedi Fn:: GetAtt per ulteriori informazioni.

Fn::Join

Utilizza `Fn::Join` per abbinare le stringhe a un delimitatore. Le stringhe possono essere hardcoded oppure utilizzare l'output da `Fn::GetAtt` o `Ref`.

```
{ "Fn::Join" : [ "delimiter", [ "string1", "string2" ] ] }
```

Per ulteriori informazioni, consulta Fn::Join.

Fn:: GetOptionSetting

Usa `Fn::GetOptionSetting` per recuperare il valore di un'impostazione di un'opzione di configurazione applicata all'ambiente.

```
"Fn::GetOptionSetting":
  Namespace: "namespace"
  OptionName: "option name"
  DefaultValue: "default value"
```

Dall'esempio archiviazione delle chiavi private:

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-west-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"
```

Esempi di risorse personalizzate

Di seguito viene riportato un elenco di file di configurazione di esempio che puoi utilizzare per personalizzare il tuo ambiente Elastic Beanstalk.

- DynamoDB e CloudWatch SNS
- Elastic Load Balancing e CloudWatch
- ElastiCache
- RDS e CloudWatch
- SQS, SNS e CloudWatch

I sottoargomenti di questa pagina forniscono alcuni esempi estesi per l'aggiunta e la configurazione di risorse personalizzate in un ambiente Elastic Beanstalk.

Esempi

- Esempio: ElastiCache
- Esempio: SQS e CloudWatch SNS
- Esempio: DynamoDB e CloudWatch SNS

Esempio: ElastiCache

I seguenti esempi aggiungono un ElastiCache cluster Amazon alle piattaforme EC2-Classic ed EC2-VPC (entrambe predefinite e personalizzate Amazon Virtual Private Cloud (Amazon VPC)). Per ulteriori informazioni su queste piattaforme e su come determinare quali sono supportate da EC2 per la tua regione e il tuo account, consulta. AWS https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html Quindi, consulta la sezione di questo argomento che si applica alla tua piattaforma.

- Piattaforme EC2-Classic
- EC2-VPC (predefinito)
- EC2-VPC (personalizzato)

Piattaforme EC2-Classic

Questo esempio aggiunge un ElastiCache cluster Amazon a un ambiente con istanze lanciate nella piattaforma EC2-Classic. Tutte le proprietà elencate in questo esempio sono le proprietà minime richieste che devono essere impostate per ogni tipo di risorsa. Puoi scaricare l'esempio alla pagina example. ElastiCache

> **Note**
>
> Questo esempio crea AWS risorse che potrebbero essere addebitate all'utente. Per ulteriori informazioni sui AWS prezzi, consultahttps://aws.amazon.com/pricing/. Alcuni servizi fanno parte del piano di utilizzo AWS gratuito. Se sei un nuovo cliente, puoi provare questi servizi gratuitamente. Per ulteriori informazioni, consultare https://aws.amazon.com/free/.

Per utilizzare questo esempio, esegui le operazioni indicate di seguito:

1. Crea una directory `.ebextensions` nella directory di primo livello del bundle di origine.

2. Crea due file di configurazione con estensione `.config`, posizionandoli nella directory `.ebextensions`. Un file di configurazione definisce le risorse, l'altro file di configurazione definisce le opzioni.
3. Distribuzione dell'applicazione in Elastic Beanstalk.

   YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Crea un file di configurazione (ad esempio, `elasticache.config`) che definisca le risorse. In questo esempio, creiamo il ElastiCache cluster specificando il nome della risorsa del ElastiCache cluster (`MyElastiCache`), dichiarandone il tipo e quindi configurando le proprietà del cluster. L'esempio fa riferimento al nome della risorsa del gruppo di ElastiCache sicurezza che viene creata e definita in questo file di configurazione. Successivamente, creiamo un gruppo ElastiCache di sicurezza. Definiamo il nome di questa risorsa, dichiariamo il tipo e aggiungiamo una descrizione per il gruppo di sicurezza. Infine, abbiamo impostato le regole di ingresso per il gruppo di ElastiCache sicurezza per consentire l'accesso solo dalle istanze all'interno del gruppo di ElastiCache sicurezza (`MyCacheSecurityGroup`) e del gruppo di sicurezza Elastic Beanstalk (). `AWSEBSecurityGroup` Il nome del parametro, `AWSEBSecurityGroup`, è un nome di risorsa fisso fornito da Elastic Beanstalk. È necessario aggiungere regole di ingresso `AWSEBSecurityGroup` al gruppo di ElastiCache sicurezza per consentire all'applicazione Elastic Beanstalk di connettersi alle istanze del cluster. ElastiCache

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.

Resources:
  MyElastiCache:
    Type: AWS::ElastiCache::CacheCluster
    Properties:
      CacheNodeType:
         Fn::GetOptionSetting:
             OptionName : CacheNodeType
             DefaultValue: cache.m1.small
      NumCacheNodes:
           Fn::GetOptionSetting:
             OptionName : NumCacheNodes
             DefaultValue: 1
      Engine:
```

```
            Fn::GetOptionSetting:
              OptionName : Engine
              DefaultValue: memcached
      CacheSecurityGroupNames:
        - Ref: MyCacheSecurityGroup
  MyCacheSecurityGroup:
    Type: AWS::ElastiCache::SecurityGroup
    Properties:
      Description: "Lock cache down to webserver access only"
  MyCacheSecurityGroupIngress:
    Type: AWS::ElastiCache::SecurityGroupIngress
    Properties:
      CacheSecurityGroupName:
        Ref: MyCacheSecurityGroup
      EC2SecurityGroupName:
        Ref: AWSEBSecurityGroup
```

Per ulteriori informazioni sulle risorse utilizzate in questo file di configurazione di esempio, consulta i seguenti riferimenti:

- AWS::ElastiCache::CacheCluster
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache: SecurityGroupIngress

Crea un file di configurazione separato denominato `options.config` e definisci le impostazioni delle opzioni personalizzate.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     CacheNodeType : cache.m1.small
     NumCacheNodes : 1
     Engine : memcached
```

Queste righe indicano a Elastic Beanstalk di ottenere i valori CacheNodeTypeper NumCacheNodes le proprietà, ed Engine CacheNodeTypedai NumCacheNodes valori, ed Engine in un file di configurazione (options.config nel nostro esempio) che contiene una sezione option_settings con una sezione aws:elasticbeanstalk:customoption che contiene una coppia nome-valore che contiene il valore effettivo da utilizzare. Nell'esempio precedente, questo significa che cache.m1.small, 1 e memcached verrebbero utilizzati per i valori. Per ulteriori informazioni su `Fn::GetOptionSetting`, consultare Funzioni.

EC2-VPC (predefinito)

Questo esempio aggiunge un ElastiCache cluster Amazon a un ambiente con istanze lanciate nella piattaforma EC2-VPC. Nello specifico, le informazioni contenute in questa sezione si applicano a uno scenario in cui EC2 avvia le istanze nel VPC predefinito. Tutte le proprietà elencate in questo esempio sono le proprietà minime richieste che devono essere impostate per ogni tipo di risorsa. Per ulteriori informazioni sull'impostazione predefinita VPCs, consulta VPC e sottoreti predefiniti.

> **Note**
>
> Questo esempio crea AWS risorse che potrebbero essere addebitate all'utente. Per ulteriori informazioni sui AWS prezzi, consultahttps://aws.amazon.com/pricing/. Alcuni servizi fanno parte del piano di utilizzo AWS gratuito. Se sei un nuovo cliente, puoi provare questi servizi gratuitamente. Per ulteriori informazioni, consultare https://aws.amazon.com/free/.

Per utilizzare questo esempio, esegui le operazioni indicate di seguito:

1. Crea una directory `.ebextensions` nella directory di primo livello del bundle di origine.
2. Crea due file di configurazione con estensione `.config`, posizionandoli nella directory `.ebextensions`. Un file di configurazione definisce le risorse, l'altro file di configurazione definisce le opzioni.
3. Distribuzione dell'applicazione in Elastic Beanstalk.

   YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Ora dai un nome al file di configurazione delle risorse `elasticache.config`. Per creare il ElastiCache cluster, questo esempio specifica il nome della risorsa ElastiCache cluster (`MyElastiCache`), ne dichiara il tipo e quindi configura le proprietà del cluster. L'esempio fa riferimento all'ID della risorsa del gruppo di sicurezza creata e definita in questo file di configurazione.

Successivamente, creiamo un gruppo di sicurezza EC2. Definiamo il nome per questa risorsa, dichiariamo il tipo, aggiungiamo una descrizione e impostiamo le regole in ingresso per il gruppo di sicurezza per consentire l'accesso solo dalle istanze all'interno del gruppo di sicurezza Elastic Beanstalk (`AWSEBSecurityGroup`). (Il nome del parametro,`AWSEBSecurityGroup`, è un nome di risorsa fisso fornito da Elastic Beanstalk. È necessario aggiungere regole di ingresso

`AWSEBSecurityGroup` al gruppo di ElastiCache sicurezza per consentire all'applicazione Elastic Beanstalk di connettersi alle istanze del cluster.) ElastiCache

Le regole in ingresso per il gruppo di sicurezza EC2 definiscono inoltre il protocollo IP e i numeri di porta su cui i nodi di cache possono accettare le connessioni. Per Redis, il numero predefinito della porta è `6379`.

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.

Resources:
  MyCacheSecurityGroup:
    Type: "AWS::EC2::SecurityGroup"
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
      SecurityGroupIngress :
        - IpProtocol : "tcp"
          FromPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          ToPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          SourceSecurityGroupName:
            Ref: "AWSEBSecurityGroup"
  MyElastiCache:
    Type: "AWS::ElastiCache::CacheCluster"
    Properties:
      CacheNodeType:
        Fn::GetOptionSetting:
          OptionName : "CacheNodeType"
          DefaultValue : "cache.t2.micro"
      NumCacheNodes:
        Fn::GetOptionSetting:
          OptionName : "NumCacheNodes"
          DefaultValue : "1"
      Engine:
        Fn::GetOptionSetting:
          OptionName : "Engine"
          DefaultValue : "redis"
      VpcSecurityGroupIds:
```

```
        -
          Fn::GetAtt:
            - MyCacheSecurityGroup
            - GroupId

Outputs:
  ElastiCache:
    Description : "ID of ElastiCache Cache Cluster with Redis Engine"
    Value :
      Ref : "MyElastiCache"
```

Per ulteriori informazioni sulle risorse utilizzate in questo file di configurazione di esempio, consulta i seguenti riferimenti:

- AWS::ElastiCache::CacheCluster
- AWS::EC2::SecurityGroup

Successivamente, dai un nome al file di configurazione delle opzioni `options.config` e definisci le impostazioni delle opzioni personalizzate.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
    NumCacheNodes : 1
    Engine : redis
    CachePort : 6379
```

Queste righe dicono a Elastic Beanstalk di ottenere i valori per le proprietà `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheNodeType`, `NumCacheNodes`, `Engine` e dai valori `CachePort` in un file config (`options.config` nel nostro esempio). Questo file include una sezione `aws:elasticbeanstalk:customoption` (in `option_settings`) che contiene coppie nome-valore con i valori effettivi da utilizzare. Nell'esempio precedente, `cache.t2.micro`, `1`, `redis` e `6379` verranno utilizzati per i valori. Per ulteriori informazioni su `Fn::GetOptionSetting`, consultare Funzioni.

EC2-VPC (personalizzato)

Se crei un VPC personalizzato sulla piattaforma EC2-VPC e lo specifichi come VPC in cui EC2 lancia le istanze, il processo di aggiunta di un ElastiCache cluster Amazon al tuo ambiente è diverso da quello di un VPC predefinito. La differenza principale è che è necessario creare un gruppo di sottoreti

per il cluster. ElastiCache Tutte le proprietà elencate in questo esempio sono le proprietà minime richieste che devono essere impostate per ogni tipo di risorsa.

> **Note**
>
> Questo esempio crea AWS risorse che potrebbero essere addebitate all'utente. Per ulteriori informazioni sui AWS prezzi, consultahttps://aws.amazon.com/pricing/. Alcuni servizi fanno parte del piano di utilizzo AWS gratuito. Se sei un nuovo cliente, puoi provare questi servizi gratuitamente. Per ulteriori informazioni, consultare https://aws.amazon.com/free/.

Per utilizzare questo esempio, esegui le operazioni indicate di seguito:

1. Crea una directory `.ebextensions` nella directory di primo livello del bundle di origine.
2. Crea due file di configurazione con estensione `.config`, posizionandoli nella directory `.ebextensions`. Un file di configurazione definisce le risorse, l'altro file di configurazione definisce le opzioni.
3. Distribuzione dell'applicazione in Elastic Beanstalk.

   YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Ora dai un nome al file di configurazione delle risorse `elasticache.config`. Per creare il ElastiCache cluster, questo esempio specifica il nome della risorsa ElastiCache cluster (`MyElastiCache`), ne dichiara il tipo e quindi configura le proprietà del cluster. Le proprietà dell'esempio fanno riferimento al nome del gruppo di sottoreti per il ElastiCache cluster e all'ID della risorsa del gruppo di sicurezza che creiamo e definiamo in questo file di configurazione.

Successivamente, creiamo un gruppo di sicurezza EC2. Definiamo il nome per questa risorsa, dichiariamo il tipo, aggiungiamo una descrizione, l'ID VPC, e impostiamo le regole in ingresso per il gruppo di sicurezza per consentire l'accesso solo dalle istanze all'interno del gruppo di sicurezza Elastic Beanstalk (`AWSEBSecurityGroup`). (Il nome del parametro,`AWSEBSecurityGroup`, è un nome di risorsa fisso fornito da Elastic Beanstalk. È necessario aggiungere regole di ingresso `AWSEBSecurityGroup` al gruppo di ElastiCache sicurezza per consentire all'applicazione Elastic Beanstalk di connettersi alle istanze del cluster.) ElastiCache

Le regole in ingresso per il gruppo di sicurezza EC2 definiscono inoltre il protocollo IP e i numeri di porta su cui i nodi di cache possono accettare le connessioni. Per Redis, il numero predefinito della porta è 6379. Infine, questo esempio crea un gruppo di sottoreti per il cluster. ElastiCache Definiamo il nome di questa risorsa, dichiariamo il tipo e aggiungiamo una descrizione e l'ID della sottorete nel gruppo di sottoreti.

> **Note**
>
> Si consiglia di utilizzare sottoreti private per il cluster. ElastiCache Per ulteriori informazioni su un VPC con una sottorete privata, consulta https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html.

```
#This sample requires you to create a separate configuration file that defines the
 custom option settings for CacheCluster properties.

Resources:
  MyElastiCache:
    Type: "AWS::ElastiCache::CacheCluster"
    Properties:
      CacheNodeType:
        Fn::GetOptionSetting:
          OptionName : "CacheNodeType"
          DefaultValue : "cache.t2.micro"
      NumCacheNodes:
        Fn::GetOptionSetting:
          OptionName : "NumCacheNodes"
          DefaultValue : "1"
      Engine:
        Fn::GetOptionSetting:
          OptionName : "Engine"
          DefaultValue : "redis"
      CacheSubnetGroupName:
        Ref: "MyCacheSubnets"
      VpcSecurityGroupIds:
        - Ref: "MyCacheSecurityGroup"
  MyCacheSecurityGroup:
    Type: "AWS::EC2::SecurityGroup"
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
      VpcId:
```

```
        Fn::GetOptionSetting:
          OptionName : "VpcId"
      SecurityGroupIngress :
        - IpProtocol : "tcp"
          FromPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          ToPort :
            Fn::GetOptionSetting:
              OptionName : "CachePort"
              DefaultValue: "6379"
          SourceSecurityGroupId:
            Ref: "AWSEBSecurityGroup"
  MyCacheSubnets:
    Type: "AWS::ElastiCache::SubnetGroup"
    Properties:
      Description: "Subnets for ElastiCache"
      SubnetIds:
        Fn::GetOptionSetting:
          OptionName : "CacheSubnets"
Outputs:
  ElastiCache:
    Description : "ID of ElastiCache Cache Cluster with Redis Engine"
    Value :
      Ref : "MyElastiCache"
```

Per ulteriori informazioni sulle risorse utilizzate in questo file di configurazione di esempio, consulta i seguenti riferimenti:

- AWS::ElastiCache::CacheCluster
- AWS::EC2::SecurityGroup
- AWS::ElastiCache::SubnetGroup

Successivamente, dai un nome al file di configurazione delle opzioni `options.config` e definisci le impostazioni delle opzioni personalizzate.

> **Note**
>
> In questo esempio, sostituisci i valori di esempio `CacheSubnets` e `VpcId` con il tuo VPC e le tue sottoreti.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
    NumCacheNodes : 1
    Engine : redis
    CachePort : 6379
    CacheSubnets:
      - subnet-1a1a1a1a
      - subnet-2b2b2b2b
      - subnet-3c3c3c3c
    VpcId: vpc-4d4d4d4d
```

Queste righe dicono a Elastic Beanstalk di ottenere i valori per le proprietà `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets`, `VpcId`, `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` e dai valori `VpcId` in un file config (`options.config` nel nostro esempio). Questo file include una sezione `aws:elasticbeanstalk:customoption` (in `option_settings`) che contiene coppie nome-valore con valori di esempio. Nell'esempio precedente, `cache.t2.micro`, `1`, `redis`, `6379`, `subnet-1a1a1a1a`, `subnet-2b2b2b2b`, `subnet-3c3c3c3c` e `vpc-4d4d4d4d` verranno utilizzati per i valori. Per ulteriori informazioni su `Fn::GetOptionSetting`, consultare Funzioni.

Esempio: SQS e CloudWatch SNS

In questo esempio viene aggiunta una coda Amazon SQS e un allarme sulla profondità della coda all'ambiente. Le proprietà elencate in questo esempio sono le proprietà minime richieste che devono essere impostate per ciascuna di queste risorse. Puoi scaricare l'esempio da SQS, SNS e. CloudWatch

> **Note**
>
> Questo esempio crea AWS risorse che potrebbero essere addebitate all'utente. Per ulteriori informazioni sui AWS prezzi, consultahttps://aws.amazon.com/pricing/. Alcuni servizi fanno

parte del piano di utilizzo AWS gratuito. Se sei un nuovo cliente, puoi provare questi servizi gratuitamente. Per ulteriori informazioni, consultare https://aws.amazon.com/free/.

Per utilizzare questo esempio, esegui le operazioni indicate di seguito:

1. Crea una directory `.ebextensions` nella directory di primo livello del bundle di origine.
2. Crea due file di configurazione con estensione `.config`, posizionandoli nella directory `.ebextensions`. Un file di configurazione definisce le risorse, l'altro file di configurazione definisce le opzioni.
3. Distribuzione dell'applicazione in Elastic Beanstalk.

   YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Crea un file di configurazione (ad esempio, sqs.config) che definisca le risorse. In questo esempio creiamo un coda SQS e definiamo la proprietà `VisbilityTimeout` nella risorsa `MySQSQueue`. Successivamente creiamo un `Topic` SNS e specifichiamo che le e-mail vengano inviate a `someone@example.com` quando viene generato l'allarme. Infine, creiamo un CloudWatch allarme se la coda supera i 10 messaggi. Nella proprietà `Dimensions` specifichiamo il nome della dimensione e il valore che rappresenta la misurazione della dimensione. Utilizziamo `Fn::GetAtt` per restituire il valore di `QueueName` da `MySQSQueue`.

```
#This sample requires you to create a separate configuration file to define the custom
 options for the SNS topic and SQS queue.
Resources:
  MySQSQueue:
    Type: AWS::SQS::Queue
    Properties:
      VisibilityTimeout:
        Fn::GetOptionSetting:
          OptionName: VisibilityTimeout
          DefaultValue: 30
  AlarmTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
```

```
            Fn::GetOptionSetting:
              OptionName: AlarmEmail
              DefaultValue: "nobody@amazon.com"
          Protocol: email
  QueueDepthAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: "Alarm if queue depth grows beyond 10 messages"
      Namespace: "AWS/SQS"
      MetricName: ApproximateNumberOfMessagesVisible
      Dimensions:
        - Name: QueueName
          Value : { "Fn::GetAtt" : [ "MySQSQueue", "QueueName"] }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 1
      Threshold: 10
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: AlarmTopic
      InsufficientDataActions:
        - Ref: AlarmTopic

Outputs :
  QueueURL:
    Description : "URL of newly created SQS Queue"
    Value : { Ref : "MySQSQueue" }
  QueueARN :
    Description : "ARN of newly created SQS Queue"
    Value : { "Fn::GetAtt" : [ "MySQSQueue", "Arn"]}
  QueueName :
    Description : "Name newly created SQS Queue"
    Value : { "Fn::GetAtt" : [ "MySQSQueue", "QueueName"]}
```

Per ulteriori informazioni sulle risorse utilizzate in questo file di configurazione di esempio, consulta i seguenti riferimenti:

- AWS::SQS::Queue
- AWS::SNS::Topic
- AWS::CloudWatch::Alarm

Crea un file di configurazione separato denominato `options.config` e definisci le impostazioni delle opzioni personalizzate.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     VisibilityTimeout : 30
     AlarmEmail : "nobody@example.com"
```

Queste righe indicano a Elastic Beanstalk di ottenere i valori VisibilityTimeout per le proprietà e Subscription Endpoint VisibilityTimeout dai valori e Subscription Endpoint in un file di configurazione (options.config nel nostro esempio) che contiene una sezione option_settings con una sezione aws:elasticbeanstalk:customoption che contiene una coppia nome-valore che contiene il valore effettivo da utilizzare. Nell'esempio precedente questo significa che per i valori verrebbero utilizzati 30 e "nobody@amazon.com". Per ulteriori informazioni su `Fn::GetOptionSetting`, consultare the section called “Funzioni”.

Esempio: DynamoDB e CloudWatch SNS

Questo file di configurazione imposta la tabella DynamoDB come gestore di sessione per un'applicazione basata su PHP utilizzando l'SDK for PHP 2. AWS Per utilizzare questo esempio, è necessario disporre di un profilo dell'istanza IAM, che viene aggiunto alle istanze nell'ambiente e utilizzato per accedere alla tabella DynamoDB.

Puoi scaricare l'esempio che utilizzeremo in questa fase facendo clic su DynamoDB session Support example. L'esempio contiene i file seguenti:

- L'applicazione d'esempio, `index.php`.
- Un file di configurazione`dynamodb.config`, per creare e configurare una tabella DynamoDB e AWS altre risorse e installare il software sulle istanze EC2 che ospitano l'applicazione in un ambiente Elastic Beanstalk
- Un file di configurazione, `options.config`, che sovrascrive le impostazioni di default in `dynamodb.config` con impostazioni specifiche per questa particolare installazione.

`index.php`

```
<?php

// Include the SDK using the Composer autoloader
require '../vendor/autoload.php';
```

```
use Aws\DynamoDb\DynamoDbClient;

// Grab the session table name and region from the configuration file
list($tableName, $region) = file(__DIR__ . '/../sessiontable');
$tableName = rtrim($tableName);
$region = rtrim($region);

// Create a DynamoDB client and register the table as the session handler
$dynamodb = DynamoDbClient::factory(array('region' => $region));
$handler = $dynamodb->registerSessionHandler(array('table_name' => $tableName,
 'hash_key' => 'username'));

// Grab the instance ID so we can display the EC2 instance that services the request
$instanceId = file_get_contents("http://169.254.169.254/latest/meta-data/instance-id");
?>
<h1>Elastic Beanstalk PHP Sessions Sample</h1>
<p>This sample application shows the integration of the Elastic Beanstalk PHP
container and the session support for DynamoDB from the AWS SDK for PHP 2.
Using DynamoDB session support, the application can be scaled out across
multiple web servers. For more details, see the
<a href="https://aws.amazon.com/php/">PHP Developer Center</a>.</p>

<form id="SimpleForm" name="SimpleForm" method="post" action="index.php">
<?php
echo 'Request serviced from instance ' . $instanceId . '<br/>';
echo '<br/>';

if (isset($_POST['continue'])) {
  session_start();
  $_SESSION['visits'] = $_SESSION['visits'] + 1;
  echo 'Welcome back ' . $_SESSION['username'] . '<br/>';
  echo 'This is visit number ' . $_SESSION['visits'] . '<br/>';
  session_write_close();
  echo '<br/>';
  echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
  echo '<input type="Submit" value="Delete Session" name="killsession"
 id="killsession"/>';
} elseif (isset($_POST['killsession'])) {
  session_start();
  echo 'Goodbye ' . $_SESSION['username'] . '<br/>';
  session_destroy();
  echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
  echo '<br/>';
```

```
  echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
} elseif (isset($_POST['newsession'])) {
  session_start();
  $_SESSION['username'] = $_POST['username'];
  $_SESSION['visits'] = 1;
  echo 'Welcome to a new session ' . $_SESSION['username'] . '<br/>';
  session_write_close();
  echo '<br/>';
  echo '<input type="Submit" value="Refresh" name="continue" id="continue"/>';
  echo '<input type="Submit" value="Delete Session" name="killsession"
 id="killsession"/>';
} else {
  echo 'To get started, enter a username.<br/>';
  echo '<br/>';
  echo 'Username: <input type="text" name="username" id="username" size="30"/><br/>';
  echo '<input type="Submit" value="New Session" name="newsession" id="newsession"/>';
}
?>
</form>
```

.ebextensions/dynamodb.config

```
Resources:
  SessionTable:
    Type: AWS::DynamoDB::Table
    Properties:
      KeySchema:
        HashKeyElement:
          AttributeName:
            Fn::GetOptionSetting:
              OptionName : SessionHashKeyName
              DefaultValue: "username"
          AttributeType:
            Fn::GetOptionSetting:
              OptionName : SessionHashKeyType
              DefaultValue: "S"
      ProvisionedThroughput:
        ReadCapacityUnits:
          Fn::GetOptionSetting:
            OptionName : SessionReadCapacityUnits
            DefaultValue: 1
        WriteCapacityUnits:
          Fn::GetOptionSetting:
```

```
            OptionName : SessionWriteCapacityUnits
            DefaultValue: 1

  SessionWriteCapacityUnitsLimit:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, "
 write capacity limit on the session table." ]]}
      Namespace: "AWS/DynamoDB"
      MetricName: ConsumedWriteCapacityUnits
      Dimensions:
        - Name: TableName
          Value: { "Ref" : "SessionTable" }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 12
      Threshold:
          Fn::GetOptionSetting:
            OptionName : SessionWriteCapacityUnitsAlarmThreshold
            DefaultValue: 240
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: SessionAlarmTopic
      InsufficientDataActions:
        - Ref: SessionAlarmTopic

  SessionReadCapacityUnitsLimit:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, " read
 capacity limit on the session table." ]]}
      Namespace: "AWS/DynamoDB"
      MetricName: ConsumedReadCapacityUnits
      Dimensions:
        - Name: TableName
          Value: { "Ref" : "SessionTable" }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 12
      Threshold:
          Fn::GetOptionSetting:
            OptionName : SessionReadCapacityUnitsAlarmThreshold
            DefaultValue: 240
      ComparisonOperator: GreaterThanThreshold
```

```
      AlarmActions:
        - Ref: SessionAlarmTopic
      InsufficientDataActions:
        - Ref: SessionAlarmTopic

  SessionThrottledRequestsAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmDescription: { "Fn::Join" : ["", [{ "Ref" : "AWSEBEnvironmentName" }, ":
 requests are being throttled." ]]}
      Namespace: AWS/DynamoDB
      MetricName: ThrottledRequests
      Dimensions:
        - Name: TableName
          Value: { "Ref" : "SessionTable" }
      Statistic: Sum
      Period: 300
      EvaluationPeriods: 1
      Threshold:
        Fn::GetOptionSetting:
          OptionName: SessionThrottledRequestsThreshold
          DefaultValue: 1
      ComparisonOperator: GreaterThanThreshold
      AlarmActions:
        - Ref: SessionAlarmTopic
      InsufficientDataActions:
        - Ref: SessionAlarmTopic

  SessionAlarmTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: SessionAlarmEmail
              DefaultValue: "nobody@amazon.com"
          Protocol: email

files:
  "/var/app/sessiontable":
    mode: "000444"
    content: |
      `{"Ref" : "SessionTable"}`
      `{"Ref" : "AWS::Region"}`
```

```
  "/var/app/composer.json":
    mode: "000744"
    content:
      {
        "require": {
           "aws/aws-sdk-php": "*"
        }
      }

container_commands:
 "1-install-composer":
   command: "cd /var/app; curl -s http://getcomposer.org/installer | php"
 "2-install-dependencies":
   command: "cd /var/app; php composer.phar install"
 "3-cleanup-composer":
   command: "rm -Rf /var/app/composer.*"
```

Nel file di configurazione di esempio, creiamo dapprima la tabella DynamoDB e configuriamo la struttura della chiave primaria per la tabella e le unità di capacità per allocare risorse sufficienti e fornire il throughput richiesto. Successivamente, creiamo allarmi per e. CloudWatch `WriteCapacity` `ReadCapacity` e quindi un argomento SNS che invia e-mail a "nobody@amazon.com" se le soglie di allarme vengono superate.

Dopo aver creato e configurato AWS le risorse per il nostro ambiente, dobbiamo personalizzare le istanze EC2. Utilizziamo la `files` chiave per passare i dettagli della tabella DynamoDB alle istanze EC2 nel nostro ambiente e per aggiungere un «require» nel file per `composer.json` l'SDK for PHP 2. AWS Eseguiamo, infine, i comandi contenitore per installare Composer, le dipendenze necessarie e quindi rimuovere il programma di installazione.

`.ebextensions/options.config`

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     SessionHashKeyName                      : username
     SessionHashKeyType                      : S
     SessionReadCapacityUnits                : 1
     SessionReadCapacityUnitsAlarmThreshold  : 240
     SessionWriteCapacityUnits               : 1
     SessionWriteCapacityUnitsAlarmThreshold : 240
     SessionThrottledRequestsThreshold       : 1
     SessionAlarmEmail                       : me@example.com
```

Sostituisci il SessionAlarmEmail valore con l'e-mail a cui desideri inviare le notifiche di allarme. Il file `options.config` contiene i valori utilizzati per alcune delle variabili definite in `dynamodb.config`. Ad esempio, `dynamodb.config` contiene le seguenti righe:

```
Subscription:
  - Endpoint:
      Fn::GetOptionSetting:
        OptionName: SessionAlarmEmail
        DefaultValue: "nobody@amazon.com"
```

Queste righe indicano a Elastic Beanstalk di ottenere il valore per la proprietà Endpoint SessionAlarmEmaildal valore in un `options.config` file di configurazione (nella nostra applicazione di esempio) che contiene una sezione option_settings con una sezione aws:elasticbeanstalk:customoption che contiene una coppia nome-valore che contiene il valore effettivo da utilizzare. Nell'esempio `nobody@amazon.com` precedente, ciò significa che verrebbe assegnato il valore. SessionAlarmEmail

Per ulteriori informazioni sulle CloudFormation risorse utilizzate in questo esempio, consulta i seguenti riferimenti:

- AWS::DynamoDB::Table
- AWS::CloudWatch::Alarm
- AWS::SNS::Topic

Utilizzo delle configurazioni salvate di Elastic Beanstalk

Puoi salvare la configurazione dell'ambiente come un oggetto in Amazon Simple Storage Service (Amazon S3) e puoi applicarla ad altri ambienti durante la creazione dell'ambiente o a un ambiente in esecuzione. Le configurazioni salvate sono modelli in formato YAML che definiscono le impostazioni versione della piattaforma, livello, opzioni di configurazione di un ambiente e i tag.

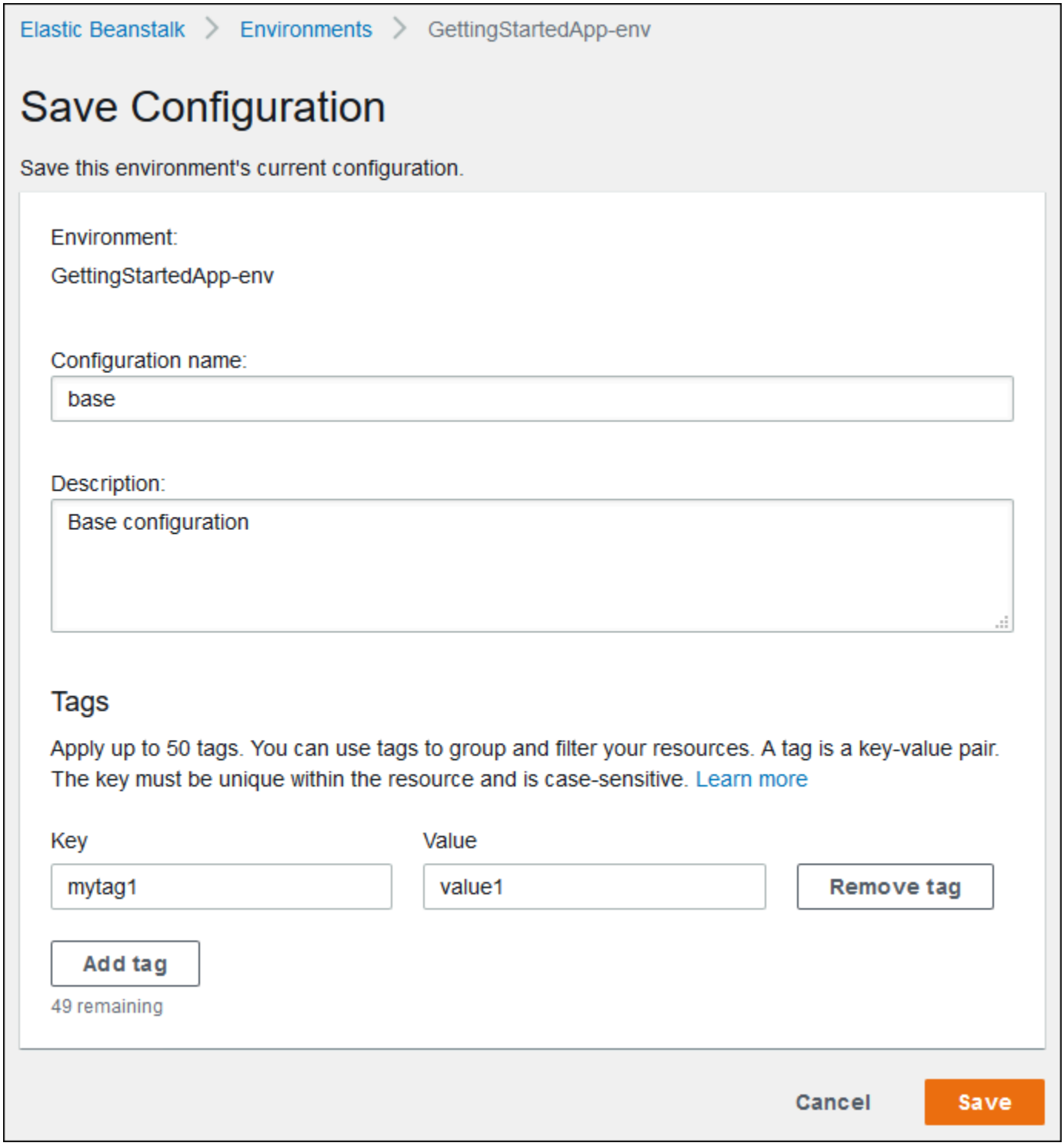
Puoi applicare tag a una configurazione salvata al momento della sua creazione e modificare i tag di configurazioni salvate esistenti. I tag applicati a una configurazione salvata non sono correlati ai tag specificati in una configurazione salvata utilizzando la chiave `Tags:`. Quest'ultima viene applicata a un ambiente quando la configurazione salvata viene applicata all'ambiente. Per informazioni dettagliate, vedi Tagging delle configurazioni salvate.

> **Note**
>
> È possibile creare e applicare le configurazioni salvate agli ambienti Elastic Beanstalk utilizzando diversi metodi. Questi includono la console Elastic Beanstalk, l'interfaccia a riga di comando EB e AWS CLI.
> Vedi i seguenti argomenti per esempi di metodi alternativi per la creazione e l'applicazione di configurazioni salvate:
>
> - Impostazione delle opzioni di configurazione prima della creazione di un ambiente
> - Impostazione delle opzioni di configurazione durante la creazione di un ambiente
> - Impostazione delle opzioni di configurazione dopo la creazione di un ambiente

Crea una configurazione salvata dallo stato attuale dell'ambiente nella console di gestione Elastic Beanstalk.

Per salvare la configurazione di un ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegli Actions (Operazioni), quindi Save configuration (Salva configurazione).
4. Utilizzare il modulo sullo schermo per assegnare un nome alla configurazione salvata. Facoltativamente, fornire una breve descrizione e aggiungere chiavi e valori dei tag.
5. Scegli Save (Salva).



La configurazione salvata include le impostazioni applicate all'ambiente con la console o con qualsiasi altro client che utilizza l'API Elastic Beanstalk. Puoi quindi applicare la configurazione salvata all'ambiente in un secondo momento per ripristinare lo stato precedente, oppure applicarla a un nuovo ambiente durante la creazione dell'ambiente.

È possibile scaricare una configurazione utilizzando il the section called “eb config” comando EB CLI, come illustrato nell'esempio seguente. *NAME*è il nome della configurazione salvata.

```
eb config get NAME
```

Per applicare una configurazione salvata durante la creazione dell'ambiente (console Elastic Beanstalk)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Saved configurations (Configurazioni salvate).
4. Selezionare la configurazione salvata che si desidera applicare, quindi scegliere Launch environment (Avvia ambiente).
5. Procedi seguendo la procedura guidata per creare l'ambiente.

Le configurazioni salvate non includono le impostazioni applicate con i file di configurazione nel codice sorgente della tua applicazione. Se la stessa impostazione viene applicata in un file di configurazione e in una configurazione salvata, l'impostazione nella configurazione salvata prevale. Analogamente, le opzioni specificate nella console Elastic Beanstalk sovrascrivono le opzioni presenti nelle configurazioni salvate. Per ulteriori informazioni, consulta Priorità.

Le configurazioni salvate sono memorizzate nel bucket S3 di Elastic Beanstalk in una cartella che ha lo stesso nome dell'applicazione. Ad esempio, le configurazioni per un'applicazione denominata `my-app` nella regione us-west-2 per l'account numero 123456789012 sono disponibili in `s3:// elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/`.

Visualizza i contenuti di una configurazione salvata aprendola in un editor di testo. La seguente configurazione di esempio mostra la configurazione di un ambiente di server Web avviata con la console di gestione Elastic Beanstalk.

```
EnvironmentConfigurationMetadata:
  Description: Saved configuration from a multicontainer Docker environment created
 with the Elastic Beanstalk Management Console
  DateCreated: '1520633151000'
  DateModified: '1520633151000'
```

```
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
 Amazon Linux/2.5.0
OptionSettings:
  aws:elasticbeanstalk:command:
    BatchSize: '30'
    BatchSizeType: Percentage
  aws:elasticbeanstalk:sns:topics:
    Notification Endpoint: me@example.com
  aws:elb:policies:
    ConnectionDrainingEnabled: true
    ConnectionDrainingTimeout: '20'
  aws:elb:loadbalancer:
    CrossZone: true
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: /
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    InstanceType: t2.micro
    EC2KeyName: workstation-uswest2
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Health
    RollingUpdateEnabled: true
EnvironmentTier:
  Type: Standard
  Name: WebServer
AWSConfigurationTemplateVersion: 1.1.0.0
Tags:
  Cost Center: WebApp Dev
```

È possibile modificare i contenuti di una configurazione salvata e salvarla nella stessa posizione in Amazon S3. Qualsiasi configurazione salvata formattata correttamente memorizzata nel percorso corretto può essere applicata a un ambiente utilizzando la console di gestione Elastic Beanstalk.

Sono supportate le seguenti chiavi.

- AWSConfigurationTemplateVersion(obbligatorio) — La versione del modello di configurazione (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- Platform: Amazon Resource Name (ARN) della versione della piattaforma dell'ambiente. Puoi specificare una piattaforma in base all'ARN oppure il nome dello stack di soluzioni.

```
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
 Amazon Linux/2.5.0
```

- SolutionStack— Il nome completo dello stack di soluzioni utilizzato per creare l'ambiente.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings— Impostazioni delle opzioni di configurazione da applicare all'ambiente. Ad esempio, la seguente voce imposta il tipo di istanza su t2.micro.

```
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: t2.micro
```

- Tags: fino a 47 tag da applicare alle risorse create all'interno dell'ambiente.

```
Tags:
  Cost Center: WebApp Dev
```

- EnvironmentTier— Il tipo di ambiente da creare. Per un ambiente server Web, è possibile escludere questa sezione (il server Web è l'impostazione predefinita). Per un ambiente lavoratore, utilizza le seguenti indicazioni.

```
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
```

> **Note**
>
> È possibile creare e applicare le configurazioni salvate agli ambienti Elastic Beanstalk utilizzando diversi metodi. Questi includono la console Elastic Beanstalk, l'interfaccia a riga di comando EB e AWS CLI.

Vedi i seguenti argomenti per esempi di metodi alternativi per la creazione e l'applicazione di configurazioni salvate:

- Impostazione delle opzioni di configurazione prima della creazione di un ambiente
- Impostazione delle opzioni di configurazione durante la creazione di un ambiente
- Impostazione delle opzioni di configurazione dopo la creazione di un ambiente

Tagging delle configurazioni salvate

Puoi applicare tag alle configurazioni AWS Elastic Beanstalk salvate. I tag sono coppie chiave-valore associate alle risorse. AWS Per informazioni sul tagging delle risorse, sui casi d'uso, sui vincoli delle chiavi e dei valori di tag e sui tipi di risorse supportati di Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Puoi specificare i tag durante la creazione di una configurazione salvata. In una configurazione salvata esistente, puoi aggiungere o rimuovere i tag e aggiornare i valori di tag esistenti. Puoi aggiungere fino a 50 tag per ogni configurazione salvata.

Aggiunta di tag durante la creazione della configurazione salvata

Quando utilizzi la console Elastic Beanstalk per salvare una configurazione, puoi specificare chiavi e valori dei tag nella pagina Save Configuration (Salva configurazione).

Se utilizzi la CLI EB per salvare un'applicazione, utilizza l'opzione `--tags` con eb config per aggiungere i tag.

```
~/workspace/my-app$ eb config --tags mytag1=value1,mytag2=value2
```

Con AWS CLI o con altri client basati su API, aggiungi tag utilizzando il `--tags` parametro sul comando. create-configuration-template

```
$ aws elasticbeanstalk create-configuration-template \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --template-name my-template --solution-stack-
name solution-stack
```

Gestione dei tag di una configurazione salvata esistente

Puoi aggiungere, aggiornare ed eliminare i tag in una configurazione salvata di Elastic Beanstalk esistente.

Per gestire i tag di una configurazione salvata utilizzando la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegliere Applicazioni, quindi selezionare il nome dell'applicazione dall'elenco.
3. Nel riquadro di spostamento individuare il nome dell'applicazione e scegliere Saved configurations (Configurazioni salvate).
4. Selezionare la configurazione salvata che si desidera gestire.
5. Scegliere Actions (Operazioni), quindi Manage tags (Gestisci tag).
6. Utilizzare il modulo su schermo per aggiungere, aggiornare o eliminare tag.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Se utilizzi la CLI EB per aggiornare la configurazione salvata, utilizza eb tags per aggiungere, aggiornare, eliminare o elencare i tag.

Ad esempio, il comando seguente consente di elencare i tag in una configurazione salvata.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Il comando seguente aggiorna il tag `mytag1` ed elimina il tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Per un elenco completo delle opzioni e per altri esempi, consulta `eb tags`.

Con il AWS CLI o altri client basati su API, usa il list-tags-for-resource comando per elencare i tag di una configurazione salvata.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-
template"
```

Utilizza il comando update-tags-for-resource per aggiungere, aggiornare o eliminare i tag in una configurazione salvata.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-
id:configurationtemplate/my-app/my-template"
```

Specifica i tag da aggiungere e i tag da aggiornare nel parametro `--tags-to-add` di update-tags-for-resource. È stato aggiunto un tag non esistente e il valore di un tag esistente è stato aggiornato.

> **Note**
>
> Per utilizzare alcuni AWS CLI comandi e CLI di EB con una configurazione salvata di Elastic Beanstalk, è necessario l'ARN della configurazione salvata. Per costruire l'ARN, utilizza innanzitutto il comando seguente per recuperare il nome della configurazione salvata.
>
> ```
> $ aws elasticbeanstalk describe-applications --application-names my-app
> ```
>
> Cerca la chiave `ConfigurationTemplates` nell'output di comando. Questo elemento mostra il nome della configurazione salvata. Utilizza questo nome dove *`my-template`* viene specificato nei comandi citati in questa pagina.

Manifest di ambiente (**`env.yaml`**)

Puoi includere un manifest di ambiente in formato YAML nella radice del bundle di origine della tua applicazione per configurare il nome dell'ambiente, lo stack di soluzioni e collegamenti tra ambienti da utilizzare durante la creazione dell'ambiente.

Questo formato di file include supporto per gruppi di ambiente. Per utilizzare i gruppi, specifica il nome dell'ambiente nel manifest con un simbolo + alla fine. Quando crei o aggiorni l'ambiente, specifica il nome del gruppo con `--group-name` (AWS CLI) o `--env-group-suffix` (CLI EB).

Per ulteriori informazioni sui gruppi, consulta Creazione e aggiornamento di gruppi di ambienti Elastic Beanstalk.

L'esempio di manifest seguente definisce un ambiente server Web con un collegamento a un componente di ambiente lavoratore da cui dipende. Il manifest utilizza i gruppi per consentire la creazione di più ambienti con lo stesso bundle di origine:

~/myapp/frontend/env.yaml

```
AWSConfigurationTemplateVersion: 1.1.0.0
SolutionStack: 64bit Amazon Linux 2015.09 v2.0.6 running Multi-container Docker 1.7.1
 (Generic)
OptionSettings:
  aws:elasticbeanstalk:command:
    BatchSize: '30'
    BatchSizeType: Percentage
  aws:elasticbeanstalk:sns:topics:
    Notification Endpoint: me@example.com
  aws:elb:policies:
    ConnectionDrainingEnabled: true
    ConnectionDrainingTimeout: '20'
  aws:elb:loadbalancer:
    CrossZone: true
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: /
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    InstanceType: t2.micro
    EC2KeyName: workstation-uswest2
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Health
    RollingUpdateEnabled: true
Tags:
  Cost Center: WebApp Dev
CName: front-A08G28LG+
EnvironmentName: front+
EnvironmentLinks:
  "WORKERQUEUE" : "worker+"
```

Sono supportate le seguenti chiavi.

- AWSConfigurationTemplateVersion(obbligatorio) — La versione del modello di configurazione (1.1.0.0).

```
AWSConfigurationTemplateVersion: 1.1.0.0
```

- Platform: Amazon Resource Name (ARN) della versione della piattaforma dell'ambiente. Puoi specificare una piattaforma in base all'ARN oppure il nome dello stack di soluzioni.

```
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
 Amazon Linux/2.5.0
```

- SolutionStack— Il nome completo dello stack di soluzioni utilizzato per creare l'ambiente.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings— Impostazioni delle opzioni di configurazione da applicare all'ambiente. Ad esempio, la seguente voce imposta il tipo di istanza su t2.micro.

```
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: t2.micro
```

- Tags: fino a 47 tag da applicare alle risorse create all'interno dell'ambiente.

```
Tags:
  Cost Center: WebApp Dev
```

- EnvironmentTier— Il tipo di ambiente da creare. Per un ambiente server Web, è possibile escludere questa sezione (il server Web è l'impostazione predefinita). Per un ambiente lavoratore, utilizza le seguenti indicazioni.

```
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
```

- CName— Il CNAME per l'ambiente. Includi un carattere + alla fine del nome per abilitare i gruppi.

```
CName: front-A08G28LG+
```

- EnvironmentName— Il nome dell'ambiente da creare. Includi un carattere + alla fine del nome per abilitare i gruppi.

  ```
  EnvironmentName: front+
  ```

  Con i gruppi abilitati, devi specificare un nome del gruppo quando crei gli ambienti. Elastic Beanstalk aggiunge il nome del gruppo al nome dell'ambiente con un trattino. Ad esempio, con il nome dell'ambiente `front+` e il nome del gruppo `dev`, Elastic Beanstalk creerà l'ambiente con il nome `front-dev`.
- EnvironmentLinks— Una mappa dei nomi delle variabili e dei nomi di ambiente delle dipendenze. L'esempio seguente rende l'ambiente `worker+` una dipendenza e indica a Elastic Beanstalk di salvare le informazioni di collegamento su una variabile denominata `WORKERQUEUE`.

  ```
  EnvironmentLinks:
    "WORKERQUEUE" : "worker+"
  ```

  Il valore della variabile di collegamento varia a seconda del tipo di ambiente collegato. Per un ambiente server Web, il link è il CNAME dell'ambiente. Per un ambiente worker, il collegamento è il nome della coda Amazon Simple Queue Service (Amazon SQS) dell'ambiente.

Le CNameEnvironmentLinkschiavi EnvironmentNamee possono essere utilizzate per creare gruppi di ambienti e collegamenti ad altri ambienti. Queste funzionalità sono attualmente supportate quando si utilizza l'EB CLI AWS CLI o un SDK.

Utilizzo di un'immagine Amazon Machine Image (AMI) personalizzata nell'ambiente Elastic Beanstalk

Questa sezione spiega quando prendere in considerazione l'utilizzo di un'AMI personalizzata e fornisce le procedure per configurare e gestire l'AMI personalizzata nel proprio ambiente. Quando crei un AWS Elastic Beanstalk ambiente, puoi specificare un'Amazon Machine Image (AMI) da utilizzare al posto dell'AMI Elastic Beanstalk standard inclusa nella versione della tua piattaforma. Un'AMI personalizzata può migliorare i tempi di provisioning quando le istanze vengono lanciate nell'ambiente se è necessario installare molto software non incluso nello standard. AMIs

L'uso dei file di configurazione è efficace per personalizzare l'ambiente in modo rapido e coerente. Tuttavia, l'applicazione delle configurazioni può iniziare a richiedere molto tempo durante la creazione

e gli aggiornamenti dell'ambiente. Se la configurazione del server occupa una posizione preminente nei file di configurazione, puoi ridurre tale tempo effettuando un'AMI personalizzata che include già il software e la configurazione di cui hai bisogno.

Un'AMI personalizzata, inoltre, consente di apportare modifiche ai componenti di basso livello, ad esempio il kernel Linux, difficili da implementare o che richiedono tempi lunghi di applicazione nei file di configurazione. Per creare un'AMI personalizzata, avvia un'AMI della piattaforma Elastic Beanstalk in Amazon EC2, personalizza il software e la configurazione in base alle tue esigenze, quindi arresta l'istanza e salva l'AMI.

Creazione di un'AMI personalizzata

È possibile utilizzare EC2 Image Builder per creare e AMIs gestire soluzioni personalizzate in alternativa a queste procedure. Per ulteriori informazioni, consultare la Guida per l'utente di Image Builder.

Per identificare la AMI Elastic Beanstalk di base

1. In una finestra di comando, eseguire un comando come il seguente. Per ulteriori informazioni, consulta describe-platform-version nella documentazione di riferimento dei comandi della AWS CLI .

   Specificate la AWS regione in cui desiderate utilizzare l'AMI personalizzata e sostituite l'ARN e il numero di versione della piattaforma con la piattaforma Elastic Beanstalk su cui è basata l'applicazione.

   Example- Mac OS/Linux

```
$ aws elasticbeanstalk describe-platform-version --region us-east-2 \
      --platform-arn "arn:aws:elasticbeanstalk:us-east-2::platform/Node.js 20
 running on 64bit Amazon Linux 2023/6.1.7" \
      --query PlatformDescription.CustomAmiList
[
    {
        "VirtualizationType": "pv",
        "ImageId": ""
    },
    {
        "VirtualizationType": "hvm",
        "ImageId": "ami-020ae06fdda6a0f66"
```

```
    }
]
```

Example- Sistema operativo Windows

```
C:\> aws elasticbeanstalk describe-platform-version --region us-east-2 --platform-
arn"arn:aws:elasticbeanstalk:us-east-2::platform/
IIS 10.0 running on 64bit Windows Server 2022/2.15.3" --query
 PlatformDescription.CustomAmiList
[
    {
        "VirtualizationType": "pv",
        "ImageId": ""
    },
    {
        "VirtualizationType": "hvm",
        "ImageId": "ami-020ae06fdda6a0f66"
    }
]
```

2. Annotare il `ImageId` valore simile `ami-020ae06fdda6a0f66` nel risultato.

Il valore è l'AMI Elastic Beanstalk di serie per la versione della piattaforma, l'architettura AWS dell'istanza EC2 e la regione pertinenti per l'applicazione. Se devi creare AMIs per più piattaforme, architetture o AWS regioni, ripeti questo processo per identificare l'AMI di base corretta per ogni combinazione.

> **Note**
>
> Non creare un'AMI da un'istanza avviata in un ambiente Elastic Beanstalk. Durante il provisioning Elastic Beanstalk apporta modifiche alle istanze che possono causare problemi nell'AMI salvata. Il salvataggio di un'immagine da un'istanza in un ambiente Elastic Beanstalk consoliderà inoltre nell'immagine la versione dell'applicazione distribuita sull'istanza.

Per Linux, è anche possibile creare un'AMI personalizzata da un'AMI della community che non è stata pubblicata da Elastic Beanstalk. Puoi usare l'AMI più recente di Amazon Linux come punto di partenza. Quando si avvia un ambiente con un'AMI Linux che non è ancora gestita da Elastic Beanstalk, Elastic Beanstalk tenta di installare i software della piattaforma (lingua, framework, server

proxy, ecc.) e i componenti aggiuntivi per supportare le caratteristiche avanzate, come il reporting sullo stato migliorato.

> **Note**
>
> La personalizzazione AMIs basata su Windows Server richiede l'AMI Elastic Beanstalk di serie da cui `describe-platform-version` viene restituita, come illustrato in precedenza nel passaggio 1.

Sebbene Elastic Beanstalk possa utilizzare un'AMI non gestita da Elastic Beanstalk, l'aumento del tempo di provisioning risultante dall'installazione di componenti mancanti da parte di Elastic Beanstalk può ridurre o eliminare i vantaggi della creazione di un'AMI personalizzata. Altre distribuzioni Linux possono funzionare con alcune risoluzioni dei problemi, ma non sono ufficialmente supportate. Se l'applicazione richiede una distribuzione Linux specifica, un'alternativa è creare un'immagine Docker ed eseguirla sulla piattaforma Docker o sulla piattaforma multicontainer Docker di Elastic Beanstalk.

Per creare un'AMI personalizzata

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Scegliere Launch Instance (Avvia istanza).
3. Se hai identificato un'AMI Elastic Beanstalk di base (utilizzando `describe-platform-version`) o un'AMI Amazon Linux, inserisci il suo ID AMI nella casella di ricerca. Quindi, premere Enter (Invio).

   È inoltre possibile eseguire una ricerca nell'elenco per un'altra AMI della community che meglio si adatta alle tue esigenze.

   > **Note**
   >
   > Consigliamo di scegliere un'AMI che utilizza la virtualizzazione HVM. Questi AMIs mostrano il tipo di virtualizzazione: hvm nella loro descrizione.
   > Per ulteriori informazioni, consulta i tipi di virtualizzazione nella Guida per l'utente di Amazon EC2.

4. Scegli Select (Seleziona) per selezionare l'AMI.

5. Seleziona un tipo di istanza, quindi scegli Next: Configure Instance Details (Successivo: configura dettagli dell'istanza).
6. (Per le piattaforme Amazon Linux AMI (AL1) dismesse) Salta questo passaggio se il tuo ambiente funziona su una piattaforma basata su Linux supportata o su una piattaforma Windows.

   Espandi la sezione Dettagli avanzati e incolla il testo seguente nel campo Dati utente.

   ```
   #cloud-config
     repo_releasever: repository version number
     repo_upgrade: none
   ```

   Il numero di versione dell'archivio è l'anno e il mese della versione nel nome dell'AMI. Ad esempio, AMIs in base alla versione di marzo 2015 di Amazon Linux, disponi di un numero `2015.03` di versione del repository. Per un'immagine Elastic Beanstalk, questo valore corrisponde alla data indicata nel nome dello stack di soluzioni per la versione della piattaforma basata su AMI Amazon Linux (precedente ad Amazon Linux 2).

   > **Note**
   >
   > L'`repo_releasever`impostazione configura la lock-on-launch funzionalità per un'AMI Amazon Linux. Questo fa sì che l'AMI utilizzi una versione fissa e specifica del repository all'avvio. Questa funzionalità non è supportata su Amazon Linux 2. Non specificarla se l'ambiente utilizza un ramo di piattaforma Amazon Linux 2 corrente. L'impostazione è obbligatoria se usi una AMI personalizzata con Elastic Beanstalk solo sui rami della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2). L'impostazione `repo_upgrade` disattiva l'installazione automatica degli aggiornamenti di sicurezza. È richiesto l'uso di un'AMI personalizzata con Elastic Beanstalk.

7. Prosegui con la procedura guidata fino ad avviare l'istanza EC2. Quando richiesto, seleziona una coppia di chiavi a cui hai accesso, in modo da connetterti all'istanza per le fasi successive.
8. Connettiti all'istanza con SSH o RDP.
9. Esegui eventuali personalizzazioni desiderate.
10. (Piattaforme Windows) Eseguire il servizio EC2 Config Sysprep. Per informazioni su EC2 Config, vedere Configurazione di un'istanza di Windows utilizzando il servizio EC2 Config. Verifica che Sysprep sia configurato per generare una password casuale che può essere recuperata dalla Console di gestione AWS.

11. Nella console Amazon EC2 arresta l'istanza EC2. Nel menu Instance Actions (Operazioni istanza) scegli quindi Create Image (EBS AMI) (Crea immagine – AMI EBS).
12. Per evitare di incorrere in AWS costi aggiuntivi, interrompi l'istanza EC2.

Per utilizzare l'AMI personalizzata in un ambiente Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Per AMI ID (ID AMI), inserisci il tuo ID AMI personalizzato.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Quando crei un nuovo ambiente con l'AMI personalizzata, è consigliabile utilizzare la stessa versione della piattaforma che hai utilizzato come base per creare l'AMI.

Gestione di un ambiente con un'AMI personalizzata

Aggiornamenti della piattaforma

Quando si utilizza un'AMI personalizzata, Elastic Beanstalk continuerà a utilizzare la stessa AMI personalizzata in un ambiente quando la versione della piattaforma viene aggiornata, indipendentemente dal fatto che l'aggiornamento venga applicato manualmente o tramite aggiornamenti gestiti della piattaforma. L'ambiente non verrà ripristinato per utilizzare l'AMI stock della nuova versione della piattaforma.

Ti consigliamo di creare una nuova AMI personalizzata basata sull'AMI di serie della nuova versione della piattaforma. In questo modo verranno applicate le patch disponibili nella nuova versione della piattaforma e si ridurranno al minimo gli errori di distribuzione dovuti a versioni incompatibili di pacchetti o librerie.

Per ulteriori informazioni sulla creazione di una nuova AMI personalizzata, consulta la sezione Creazione di un'AMI personalizzata precedente di questo argomento.

Rimozione di un'AMI personalizzata

Se desideri rimuovere un'AMI personalizzata da un ambiente e reimpostarla per utilizzare l'AMI di serie per la versione della piattaforma dell'ambiente, usa il seguente comando CLI.

```
aws elasticbeanstalk update-environment \
  --application-name my-application \
  --environment-name my-environment \
  --region us-east-1 \
  --options-to-remove Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId
```

> **Note**
>
> Per evitare interruzioni del servizio, testate l'applicazione con un'AMI di serie prima di applicare questa modifica al vostro ambiente di produzione.

Pulizia di un'AMI personalizzata

Dopo aver usato un'AMI personalizzata e quando questa non è più necessaria per avviare ambienti Elastic Beanstalk, puoi scegliere di eliminarla per ridurre al minimo i costi di storage. L'eliminazione di un'AMI personalizzata comporta l'annullamento della registrazione da Amazon EC2 e l'eliminazione di altre risorse associate. Per informazioni dettagliate, consulta Annullamento della registrazione dell'AMI Linux o Annullamento della registrazione dell'AMI Windows.

Preservare l'accesso a una Amazon Machine Image (AMI) per una piattaforma ritirata

Elastic Beanstalk imposta lo stato di una ramificazione della piattaforma come ritirato quando il sistema operativo o il componente principale utilizzato dalla ramificazione raggiunge lo stato End of Life. L'AMI Elastic Beanstalk di base per il ramo della piattaforma può anche essere resa privata per impedire l'uso di questa AMI. out-of-date Gli ambienti AMIs che utilizzano quelle che sono state rese private non saranno più in grado di avviare istanze.

Se non riesci a migrare l'applicazione in un ambiente supportato prima che venga ritirata, l'ambiente potrebbe trovarsi in questa situazione. Potrebbe sorgere la necessità di aggiornare un ambiente per una ramificazione della piattaforma Beanstalk, in cui la sua AMI Elastic Beanstalk di base è stata resa privata. È disponibile un approccio alternativo. Puoi aggiornare un ambiente esistente in base a una copia dell'AMI Elastic Beanstalk di base utilizzata dall'ambiente.

Questo argomento offre alcuni passaggi e uno script autonomo per aggiornare un ambiente esistente basato su una copia dell'AMI Elastic Beanstalk di base utilizzata dall'ambiente. Una volta che sarai in grado di migrare l'applicazione su una piattaforma supportata, potrai continuare a utilizzare le procedure standard per la manutenzione dell'applicazione e degli ambienti supportati.

Procedura manuale

Per aggiornare un ambiente basato su una copia AMI dell'AMI Elastic Beanstalk di base

1. Determina quale AMI sta utilizzando il tuo ambiente. Questo comando restituisce l'AMI usato dall'ambiente Elastic Beanstalk fornito nei parametri. Il valore restituito viene utilizzato come source-ami-idnel passaggio successivo.

   In una finestra di comando, eseguire un comando come il seguente. Per ulteriori informazioni, consulta describe-configuration-settings nella documentazione di riferimento dei comandi della AWS CLI .

   Specificare la AWS regione che memorizza l'AMI di origine che si desidera copiare. Sostituisci il nome dell'applicazione e il nome dell'ambiente con quelli basati sull'AMI di origine. Inserisci il testo per il parametro di interrogazione come mostrato.

   Example

   ```
   >aws elasticbeanstalk describe-configuration-settings \
     --application-name my-application \
     --environment-name my-environment \
     --region us-east-2 \
     --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] |
    [0].Value"
   ```

2. Copia l'AMI nell'account. Questo comando restituisce la nuova AMI risultante dalla copia di source-ami-idquella restituita nel passaggio precedente.

   > **Note**
   >
   > Assicurati di prendere nota del nuovo ID AMI emesso da questo comando. Dovrai inserirlo nel passaggio successivo, sostituendolo copied-ami-idnel comando di esempio.

In una finestra di comando, eseguire un comando come il seguente. Per ulteriori informazioni, consulta copy-image nella Guida di riferimento dei comandi AWS CLI .

Specificare la AWS regione dell'AMI di origine che si desidera copiare (--source-region) e la regione in cui si desidera utilizzare la nuova AMI personalizzata (--region). Sostituisci source-ami-id con l'AMI dell'immagine che stai copiando. source-ami-idÈ stato restituito dal comando nel passaggio precedente. Sostituiscilo new-ami-namecon un nome per descrivere la nuova AMI nella regione di destinazione. Lo script che segue questa procedura genera il nuovo nome AMI aggiungendo la stringa "Copia di" all'inizio del nome di. source-ami-id

```
>aws ec2 copy-image \
    --region us-east-2 \
    --source-image-id source-ami-id \
    --source-region us-east-2 \
    --name new-ami-name
```

3. Aggiornare un ambiente per utilizzare l'AMI copiata. Dopo l'esecuzione, il comando restituisce lo stato dell'ambiente.

   In una finestra di comando, eseguire un comando come il seguente. Per ulteriori informazioni, consulta update-environment nella Guida di riferimento dei comandi AWS CLI .

   Specificate la AWS regione dell'ambiente e dell'applicazione da aggiornare. Sostituisci il nome dell'applicazione e il nome dell'ambiente con quelli che devi associare copied-ami-idal passaggio precedente. Per il parametro --option-setttings, sostituiscilo con `copied-ami-id` l'ID AMI che hai annotato dall'output del comando precedente.

```
>aws elasticbeanstalk update-environment \
  --application-name my-application \
  --environment-name my-environment \
  --region us-east-2 \
  --option-settings
 "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=copied-
ami-id"
```

> **Note**
>
> Per ridurre al minimo i costi di archiviazione, si consiglia di ripulire l'AMI personalizzata quando non è più necessaria per avviare gli ambienti Elastic Beanstalk. Per ulteriori informazioni, consulta Pulizia di un'AMI personalizzata.

Script standalone

Lo script seguente fornisce gli stessi risultati dei passaggi manuali precedenti. Scarica lo script selezionando questo link: copy_ami_and_update_env.zip.

Fonte dello script: copy_ami_and_update_env.sh

```
#!/bin/bash

set -ue

USAGE="This script is used to copy an AMI used by your Elastic Beanstalk environment
 into your account to use in your environment.\n\n"
USAGE+="Usage:\n\n"
USAGE+="./$(basename $0) [OPTIONS]\n"
USAGE+="OPTIONS:\n"
USAGE+="\t--application-name <application-name>\tThe name of your Elastic Beanstalk
 application.\n"
USAGE+="\t--environment-name <environment-name>\tThe name of your Elastic Beanstalk
 environment.\n"
USAGE+="\t--region <region> \t\t\tThe AWS region your Elastic Beanstalk environment is
 deployed to.\n"
USAGE+="\n\n"
USAGE+="Script Usage Example(s):\n"
USAGE+="./$(basename $0) --application-name my-application --environment-name my-
environment --region us-east-1\n"

if [ $# -eq 0 ]; then
  echo -e $USAGE
  exit
fi

while [[ $# -gt 0 ]]; do
  case $1 in
    --application-name)       APPLICATION_NAME="$2"; shift ;;
```

```
    --environment-name)       ENVIRONMENT_NAME="$2"; shift ;;
    --region)                 REGION="$2"; shift ;;
    *)                        echo "Unknown option $1" ; echo -e $USAGE ; exit ;;
  esac
  shift
done

aws_cli_version="$(aws --version)"
if [ $? -ne 0 ]; then
  echo "aws CLI not found. Please install it: https://docs.aws.amazon.com/cli/latest/
userguide/getting-started-install.html. Exiting."
  exit 1
fi
echo "Using aws CLI version: ${aws_cli_version}"

account=$(aws sts get-caller-identity --query "Account" --output text)
echo "Using account ${account}"

environment_ami_id=$(aws elasticbeanstalk describe-configuration-settings \
  --application-name "$APPLICATION_NAME" \
  --environment-name "$ENVIRONMENT_NAME" \
  --region "$REGION" \
  --query "ConfigurationSettings[0].OptionSettings[?OptionName=='ImageId'] | [0].Value"
 \
  --output text)
echo "Image associated with environment ${ENVIRONMENT_NAME} is ${environment_ami_id}"

owned_image=$(aws ec2 describe-images \
  --owners self \
  --image-ids "$environment_ami_id" \
  --region "$REGION" \
  --query "Images[0]" \
  --output text)
if [ "$owned_image" != "None" ]; then
  echo "${environment_ami_id} is already owned by account ${account}. Exiting."
  exit
fi

source_image_name=$(aws ec2 describe-images \
  --image-ids "$environment_ami_id" \
  --region "$REGION" \
  --query "Images[0].Name" \
  --output text)
if [ "$source_image_name" = "None" ]; then
```

```
  echo "Cannot find ${environment_ami_id}. Please contact AWS support if you need
 additional help: https://aws.amazon.com/support."
  exit 1
fi

copied_image_name="Copy of ${source_image_name}"
copied_ami_id=$(aws ec2 describe-images \
  --owners self \
  --filters Name=name,Values="${copied_image_name}" \
  --region "$REGION" \
  --query "Images[0].ImageId" \
  --output text)
if [ "$copied_ami_id" != "None" ]; then
  echo "Detected that ${environment_ami_id} has already been copied by account
 ${account}. Skipping image copy."
else
  echo "Copying ${environment_ami_id} to account ${account} with name
 ${copied_image_name}"
  copied_ami_id=$(aws ec2 copy-image \
    --source-image-id "$environment_ami_id" \
    --source-region "$REGION" \
    --name "$copied_image_name" \
    --region "$REGION" \
    --query "ImageId" \
    --output text)
  echo "New AMI ID is ${copied_ami_id}"

  echo "Waiting for ${copied_ami_id} to become available"
  aws ec2 wait image-available \
    --image-ids "$copied_ami_id" \
    --region "$REGION"
  echo "${copied_ami_id} is now available"
fi

echo "Updating environment ${ENVIRONMENT_NAME} to use ${copied_ami_id}"
environment_status=$(aws elasticbeanstalk update-environment \
  --application-name "$APPLICATION_NAME" \
  --environment-name "$ENVIRONMENT_NAME" \
  --option-settings
 "Namespace=aws:autoscaling:launchconfiguration,OptionName=ImageId,Value=
${copied_ami_id}" \
  --region "$REGION" \
  --query "Status" \
  --output text)
```

```
echo "Environment ${ENVIRONMENT_NAME} is now ${environment_status}"

echo "Waiting for environment ${ENVIRONMENT_NAME} update to complete"
aws elasticbeanstalk wait environment-updated \
  --application-name "$APPLICATION_NAME" \
  --environment-names "$ENVIRONMENT_NAME" \
  --region "$REGION"
echo "Environment ${ENVIRONMENT_NAME} update complete"
```

> **Note**
>
> È necessario averlo AWS CLI installato per eseguire lo script. Per le istruzioni di installazione, consulta Installa o aggiorna l'ultima versione della AWS CLI nella Guida per l'utente di AWS Command Line Interface .
>
> Dopo aver installato AWS CLI, è inoltre necessario configurarlo per utilizzare l' AWS account proprietario dell'ambiente. Per ulteriori informazioni, consulta Configura la AWS CLI nella Guida per l'utente di AWS Command Line Interface . L'account deve inoltre disporre delle autorizzazioni per creare un'AMI e aggiornare l'ambiente Elastic Beanstalk.

Questi passaggi descrivono il processo seguito dallo script.

1. Stampa l'account in uso.
2. Determina quale AMI viene utilizzata dall'ambiente (AMI di origine).
3. Verifica se l'AMI di origine è già di proprietà dell'account. Se sì, esci.
4. Determina il nome dell'AMI di origine in modo che possa essere utilizzato nel nuovo nome AMI. Questo serve anche a confermare l'accesso all'AMI di origine.
5. Controlla se l'AMI di origine è già stato copiato nell'account. Questa operazione viene eseguita mediante la ricerca AMIs con il nome dell'AMI copiata di proprietà dell'account. Se il nome AMI è stato modificato tra le esecuzioni dello script, copierà nuovamente l'immagine.
6. Se l'AMI di origine non è già stato copiato, copia l'AMI di origine sull'account e attendi che la nuova AMI sia disponibile.
7. Aggiorna la configurazione dell'ambiente per utilizzare la nuova AMI.
8. Attendi il completamento dell'aggiornamento dell'ambiente.

Dopo aver estratto lo script dal file copy_ami_and_update_env.zip, eseguilo procedendo come nell'esempio proposto. Sostituisci il nome dell'applicazione e il nome dell'ambiente nell'esempio con i propri valori.

```
>sh copy_ami_and_update_env.sh \
  --application-name my-application \
  --environment-name my-environment \
  --region us-east-1
```

> **Note**
>
> Per ridurre al minimo i costi di archiviazione, si consiglia di ripulire l'AMI personalizzata quando non è più necessaria per avviare gli ambienti Elastic Beanstalk. Per ulteriori informazioni, consulta Pulizia di un'AMI personalizzata.

Fornire i file statici

Per migliorare le prestazioni puoi configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Elastic Beanstalk supporta la configurazione del proxy per servire file statici sulla maggior parte dei rami della piattaforma basati su Amazon Linux 2. L'unica eccezione è Docker.

> **Note**
>
> Sulle piattaforme Python e Ruby, Elastic Beanstalk configura alcune cartelle di file statici per impostazione predefinita. Per i dettagli, consulta le sezioni di configurazione dei file statici per Python e Ruby. È possibile configurare cartelle aggiuntive come spiegato in questa pagina.

Configurare i file statici utilizzando la console

Per configurare il server proxy per fornire file statici

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri fino alla sezione Platform software (Software della piattaforma) e individua il gruppo di file statici.

   a. Per aggiungere una mappatura statica dei file, seleziona Add static files (Aggiungi file statici). Nella riga aggiuntiva che appare inserirai un percorso per la pubblicazione dei file statici e la directory che contiene i file statici da servire.

      - Nel campo Percorso, inizia il nome del percorso con una barra (/) (ad esempio, "/images").
      - Nel campo Directory, specificate un nome di directory situato nella radice del codice sorgente dell'applicazione. Non iniziarlo con una barra (ad esempio, "static/image-files").

      > **Note**
      >
      > Se non trovi la sezione File statici, devi aggiungere almeno una mappatura utilizzando un file di configurazione. Per i dettagli, vedere the section called "Configurare i file statici utilizzando le opzioni di configurazione" in questa pagina.

   b. Per rimuovere una mappatura, seleziona Remove (Rimuovi).
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurare i file statici utilizzando le opzioni di configurazione

È possibile utilizzare un file di configurazione per configurare percorsi di file statici e percorsi di directory utilizzando le opzioni di configurazione. È possibile aggiungere un file di configurazione al bundle di origine dell'applicazione e distribuirlo durante la creazione dell'ambiente o una distribuzione successiva.

Se l'ambiente utilizza un ramo di piattaforma basato su Amazon Linux 2, utilizza lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles`.

Il seguente file di configurazione di esempio indica al server proxy di servire i file nella cartella `statichtml` nel percorso `/html` e i file nella cartella `staticimages` nel percorso `/images`.

Example.ebextensions/static-files.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
```

Se il tuo ambiente Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le seguenti informazioni aggiuntive:

Spazi dei nomi specifici della piattaforma AMI Amazon Linux

Sui rami della piattaforma AMI Amazon Linux gli spazi dei nomi di configurazione dei file statici variano in base alla piattaforma. Per i dettagli, consulta le seguenti pagine:

- Spazi dei nomi per la configurazione di Go
- Spazio dei nomi per la configurazione di Java SE
- Spazio dei nomi della configurazione Tomcat
- Spazi dei nomi per la configurazione di Node.js
- Spazio dei nomi di configurazione di Python

Configurazione del protocollo HTTPS per l'ambiente Elastic Beanstalk

Gli argomenti di questa sezione spiegano come configurare HTTPS per l'ambiente Elastic Beanstalk. Il protocollo HTTPS è obbligatorio per tutte le applicazioni che trasmettono i dati utente o le informazioni di accesso.

Se hai acquistato e configurato un nome di dominio personalizzato per l'ambiente Elastic Beanstalk, puoi utilizzare il protocollo HTTPS per permettere agli utenti di collegarsi al tuo sito Web in modo sicuro.

Se non possiedi un nome di dominio, puoi comunque utilizzare il protocollo HTTPS con un certificato autofirmato a scopo di sviluppo e test. Per ulteriori informazioni, consulta Certificati server.

Configurazione della terminazione HTTPS sul sistema di bilanciamento del carico

Un load balancer distribuisce le richieste alle istanze EC2 che eseguono l'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet. Il modo più semplice per utilizzare HTTPS con un ambiente multiistanza Elastic Beanstalk consiste nel configurare un listener sicuro per il load balancer. La connessione tra il client e il sistema di bilanciamento del carico rimane sicura, quindi puoi configurare il load balancer per terminare HTTPS. Le connessioni back-end tra il load balancer e le istanze EC2 utilizzano HTTP, quindi non è richiesta alcuna configurazione aggiuntiva delle istanze. Per istruzioni dettagliate su come configurare un listenter sicuro, consulta. Configurazione della terminazione HTTPS sul sistema di bilanciamento del carico

Configurazione della terminazione HTTPS sull'istanza EC2

Se esegui l'applicazione in un ambiente a istanza singola o se intendi proteggere l'intera connessione alle istanze EC2 che usano il sistema di bilanciamento del carico, puoi configurare il server proxy in esecuzione nell'istanza per terminare il protocollo HTTPS. La configurazione delle istanze per terminare le connessioni HTTPS richiede l'uso di file di configurazione per modificare il software in esecuzione nelle istanze e i gruppi di sicurezza al fine di permettere le connessioni sicure. Per ulteriori informazioni, consulta Configurazione della terminazione HTTPS sull'istanza.

Configurazione di HTTPS end-to-end

Per end-to-end HTTPS in un ambiente con bilanciamento del carico, puoi combinare la terminazione dell'istanza e del load balancer per crittografare entrambe le connessioni. Per impostazione predefinita, se configuri il sistema di bilanciamento del carico per inoltrare il traffico tramite HTTPS, quest'ultimo considererà attendibile qualsiasi certificato presentato dalle istanze back-end. Per la massima sicurezza, puoi collegare policy al sistema di bilanciamento del carico per impedire la connessione alle istanze che non dispongono di un certificato pubblico considerato attendibile. Per ulteriori informazioni, consulta Configurazione della crittografia end-to-end in un ambiente Elastic Beanstalk con bilanciamento del carico.

Configurazione di HTTPS con TCP Passthrough

È inoltre possibile configurare il sistema di bilanciamento del carico per l'inoltro del traffico HTTPS senza decrittografarlo. Per ulteriori informazioni, consulta Configurazione del sistema di bilanciamento del carico dell'ambiente per il passthrough TCP.

> **Note**
>
> Il Ha serpenti? l'applicazione di esempio GitHub include file di configurazione e istruzioni per ogni metodo di configurazione HTTPS con un'applicazione web Tomcat. Consulta il file readme e le istruzioni relative ad HTTPS per i dettagli.

Argomenti

- Certificati server
- Configurazione della terminazione HTTPS sul sistema di bilanciamento del carico
- Configurazione della terminazione HTTPS sull'istanza
- Configurazione della crittografia end-to-end in un ambiente Elastic Beanstalk con bilanciamento del carico
- Configurazione del sistema di bilanciamento del carico dell'ambiente per il passthrough TCP
- Configurazione del reindirizzamento da HTTP a HTTPS

Certificati server

Questo argomento descrive i diversi tipi di certificati che è possibile utilizzare per configurare HTTPS e quando applicarli. Gli argomenti secondari di questa sezione forniscono istruzioni per creare il proprio certificato e come caricarlo.

AWS Certificate Manager (ACM)

ACM è lo strumento preferito per il provisioning, la gestione e la distribuzione dei certificati del server. È possibile farlo a livello di codice o utilizzando. AWS CLI Con ACM puoi creare gratuitamente un certificato affidabile per i tuoi nomi di dominio.

I certificati ACM possono essere utilizzati solo con sistemi di bilanciamento del AWS carico e CloudFront distribuzioni Amazon e ACM è disponibile solo in alcune regioni. AWS Per utilizzare un certificato ACM con Elastic Beanstalk, consulta Configurazione della terminazione HTTPS sul sistema di bilanciamento del carico. Per ulteriori informazioni su ACM, consulta la Guida per l'utente.AWS Certificate Manager

> **Note**
>
> Per un elenco delle regioni in cui è disponibile ACM, consulta Endpoint e quote ACM nel. Riferimenti generali di Amazon Web Services

Se ACM non è disponibile nella tua AWS regione, puoi caricare un certificato di terze parti o autofirmato e una chiave privata su (IAM). AWS Identity and Access Management Puoi utilizzare il AWS CLI per caricare il certificato. I certificati archiviati in IAM possono essere utilizzati con sistemi di bilanciamento del carico e CloudFront distribuzioni. Per ulteriori informazioni, consulta Caricamento di un certificato in IAM.

Certificato di terze parti

Se ACM non è disponibile nella tua regione, puoi acquistare un certificato attendibile da una terza parte. Puoi utilizzare un certificato di terze parti per decrittografare il traffico HTTPS a livello di sistema di bilanciamento del carico e/o delle istanze back-end.

Certificato autofirmato

Per i processi di sviluppo e test, puoi creare e firmare un certificato utilizzando strumenti open source. I certificati autofirmati sono gratuiti e semplici da creare, ma non possono essere utilizzati per la decrittografia front-end nei siti pubblici. Se tenti di utilizzare un certificato autofirmato per una connessione HTTPS a un client, il browser dell'utente mostra un messaggio di errore che indica che il sito Web non è sicuro. Puoi, tuttavia, utilizzare un certificato autofirmato per proteggere le connessioni back-end senza alcun problema.

Creazione e firma di un certificato X509

Puoi creare un certificato X509 per la tua applicazione con `OpenSSL`. OpenSSL è una libreria open source standard che supporta un'ampia gamma di funzioni di crittografia, tra cui la creazione e la firma di certificati X509. Per ulteriori informazioni su OpenSSL, visita la pagina Web all'indirizzo www.openssl.org.

> **Note**
>
> È sufficiente creare un certificato in locale se si desidera utilizzare HTTPS in un ambiente a istanza singola o eseguire nuovamente la crittografia al back-end con un certificato autofirmato. Se possiedi un nome di dominio, puoi creare un certificato AWS e utilizzarlo

gratuitamente in un ambiente con bilanciamento del carico utilizzando AWS Certificate Manager (ACM). Per istruzioni, consulta la pagina relativa alla richiesta di un certificato nella Guida per l'utente di AWS Certificate Manager .

Esegui `openssl version` alla riga di comando per vedere se OpenSSL è già installato. In caso contrario, puoi compilare e installare il codice sorgente seguendo le istruzioni disponibili nell' GitHub archivio pubblico o utilizzare il tuo gestore di pacchetti preferito. OpenSSL è installato anche sulle immagini Linux di Elastic Beanstalk, quindi un'alternativa rapida è connettersi a EC2 un'istanza in un ambiente in esecuzione utilizzando il comando dell'EB CLI: eb ssh

```
~/eb$ eb ssh
[ec2-user@ip-255-55-55-255 ~]$ openssl version
OpenSSL 1.0.1k-fips 8 Jan 2015
```

È necessario creare una chiave privata RSA per creare la richiesta di firma del certificato (CSR). Per creare la chiave privata, utilizza il comando openssl genrsa:

```
[ec2-user@ip-255-55-55-255 ~]$ openssl genrsa 2048 > privatekey.pem
Generating RSA private key, 2048 bit long modulus
.....................................................................................
+++
...............+++
e is 65537 (0x10001)
```

`privatekey.pem`

Il nome del file in cui desideri salvare la chiave privata. Di solito, il comando openssl genrsa stampa il contenuto della chiave privata sullo schermo, ma questo memorizza l'output in un file. Scegli un nome di file e memorizza il file in uno spazio sicuro, in modo che sia possibile recuperarlo in un secondo momento. Se perdi la chiave privata, non potrai più utilizzare il tuo certificato.

Un CSR è un file inviato a un'autorità di certificazione (CA) per richiedere un certificato server digitale. Per creare un CSR, utilizza il comando openssl req:

```
$ openssl req -new -key privatekey.pem -out csr.pem
You are about to be asked to enter information that will be incorporated
```

```
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Inserisci le informazioni richieste e premi Enter (Invio). La tabella seguente descrive e mostra esempi per ogni campo:

Nome	Descrizione	Esempio
Nome paese	L'abbreviazione ISO di due lettere per il tuo paese.	US = Stati Uniti
Lo stato o la provincia	Il nome dello stato o della provincia in cui si trova la tua organizzazione. Questo nome non può essere abbreviato.	Washington
Locality Name (Nome località)	Il nome della città in cui si trova la tua organizzazione.	Seattle
Nome organizzazione	La denominazione legale completa della tua organizzazione. Non abbreviare il nome dell'organizzazione.	Example Corporation
Unità organizzativa	Facoltativo, per informazioni aggiuntive sull'organizzazione.	Marketing
Common Name (Nome comune)	Il nome di dominio completo del tuo sito Web. Deve corrispondere al nome di dominio che gli utenti vedono quando visitano il sito, in caso contrario verranno mostrati gli errori di certificato.	www.example.com
Indirizzo e-mail	L'indirizzo e-mail dell'amministratore del sito.	someone@example.com

È possibile inviare la richiesta di firma a una terza parte per la firma o firmare personalmente per lo sviluppo e il test. I certificati autofirmati possono essere utilizzati anche per il backend HTTPS tra un sistema di bilanciamento del carico e le istanze. EC2

Per firmare il certificato, utilizza il comando openssl x509. L'esempio seguente utilizza la chiave privata del passaggio precedente (*privatekey.pem*) e la richiesta di firma (*csr.pem*) per creare un certificato pubblico denominato valido per *public.crt* giorni. *365*

```
$ openssl x509 -req -days 365 -in csr.pem -signkey privatekey.pem -out public.crt
Signature ok
subject=/C=us/ST=washington/L=seattle/O=example corporation/OU=marketing/
CN=www.example.com/emailAddress=someone@example.com
Getting Private key
```

Mantieni la chiave privata e il certificato pubblico per un utilizzo successivo. Puoi annullare la richiesta di firma. Archivia sempre la chiave privata in un percorso sicuro ed evita di aggiungerla al tuo codice sorgente.

Per utilizzare il certificato con la piattaforma Windows Server, è necessario convertirlo in formato PFX. Utilizza il comando seguente per creare un certificato PFX dai file della chiave privata e del certificato pubblico:

```
$ openssl pkcs12 -export -out example.com.pfx -inkey privatekey.pem -in public.crt
Enter Export Password: password
Verifying - Enter Export Password: password
```

Ora che hai un certificato, puoi caricarlo in IAM per utilizzarlo con un sistema di bilanciamento del carico oppure configurare le istanze nel tuo ambiente per terminare HTTPS.

Caricamento di un certificato in IAM

Per utilizzare il certificato con il sistema di bilanciamento del carico dell'ambiente Elastic Beanstalk, carica il certificato e la chiave privata su (IAM). AWS Identity and Access Management Puoi utilizzare un certificato archiviato in IAM con i sistemi di bilanciamento del carico Elastic Load Balancing e le distribuzioni Amazon. CloudFront

> **Note**
>
> AWS Certificate Manager (ACM) è lo strumento preferito per fornire, gestire e distribuire i certificati del server. Per ulteriori informazioni sulla richiesta di un certificato ACM, consulta Richiesta di un certificato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di

certificati nella Guida per l'utente di AWS Certificate Manager . Utilizza IAM per caricare un certificato solo se ACM non è disponibile nella tua regione. AWS

Puoi usare AWS Command Line Interface (AWS CLI) per caricare il tuo certificato. Il comando seguente carica un certificato autofirmato denominato *`https-cert.crt`* con una chiave privata denominata: *`private-key.pem`*

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-body file://https-cert.crt --private-key file://private-key.pem
{
    "ServerCertificateMetadata": {
        "ServerCertificateId": "AS5YBEIONO2Q7CAIHKNGC",
        "ServerCertificateName": "elastic-beanstalk-x509",
        "Expiration": "2017-01-31T23:06:22Z",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:server-certificate/elastic-beanstalk-x509",
        "UploadDate": "2016-02-01T23:10:34.167Z"
    }
}
```

Il `file://` prefisso indica AWS CLI a di caricare il contenuto di un file nella directory corrente. *`elastic-beanstalk-x509`*specifica il nome per chiamare il certificato in IAM.

Se hai acquistato un certificato da un'autorità di certificazione e hai ricevuto un file della catena di certificati, carica anche questo file includendo l'opzione `--certificate-chain`:

```
$ aws iam upload-server-certificate --server-certificate-name elastic-beanstalk-x509 --
certificate-chain file://certificate-chain.pem --certificate-body file://https-cert.crt
 --private-key file://private-key.pem
```

Annota l'Amazon Resource Name (ARN) per il certificato. Lo userai quando aggiorni le impostazioni di configurazione del sistema di bilanciamento del carico per l'uso di HTTPS.

> **Note**
>
> Un certificato caricato in IAM rimane archiviato anche quando non è più in uso in alcun sistema di bilanciamento del carico dell'ambiente. Il certificato contiene dati sensibili. Quando non è più necessario per nessun ambiente, assicurati di eliminarlo. Per informazioni

dettagliate sull'eliminazione di un certificato da IAM, consulta https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html#delete-server-certificate.

Per ulteriori informazioni sui certificati server in IAM, consulta l'argomento relativo all'utilizzo dei certificati server nella Guida per l'utente di IAM.

Archiviazione di chiavi private in modo sicuro in Amazon S3

La chiave privata che usi per firmare i certificati pubblici è privata e non deve esserne eseguito il commit nel codice sorgente. Puoi evitare di archiviare le chiavi private nei file di configurazione caricandoli in Amazon S3 e configurando Elastic Beanstalk in modo che scarichi il file da Amazon S3 durante la distribuzione dell'applicazione.

L'esempio seguente mostra le sezioni Resources e files di un file di configurazione che scarica un file di chiavi private da un bucket Amazon S3.

Example.ebextensions/privatekey.config

```
Resources:
  AWSEBAutoScalingGroup:
    Metadata:
      AWS::CloudFormation::Authentication:
        S3Auth:
          type: "s3"
          buckets: ["elasticbeanstalk-us-west-2-123456789012"]
          roleName:
            "Fn::GetOptionSetting":
              Namespace: "aws:autoscaling:launchconfiguration"
              OptionName: "IamInstanceProfile"
              DefaultValue: "aws-elasticbeanstalk-ec2-role"
files:
  # Private key
  "/etc/pki/tls/certs/server.key":
    mode: "000400"
    owner: root
    group: root
    authentication: "S3Auth"
    source: https://elasticbeanstalk-us-west-2-123456789012.s3.us-west-2.amazonaws.com/
server.key
```

Sostituisci il nome del bucket e l'URL dell'esempio con quelli personalizzati. La prima voce di questo file aggiunge un metodo di autenticazione denominato `S3Auth` ai metadati del gruppo Auto Scaling dell'ambiente. Se hai configurato un profilo di istanza personalizzato per il tuo ambiente, verrà utilizzato. In caso contrario, viene applicato il valore predefinito `aws-elasticbeanstalk-ec2-role`. Il profilo di istanza predefinito è dotato dell'autorizzazione di leggere dal bucket di storage di Elastic Beanstalk. Se utilizzi un altro bucket, aggiungi le autorizzazioni al profilo dell'istanza.

La seconda voce usa il metodo di autenticazione `S3Auth` per scaricare la chiave privata dall'URL specificato e salvarla `/etc/pki/tls/certs/server.key`. Il server proxy è in grado di leggere la chiave privata da questa posizione per terminare le connessioni HTTPS presso l'istanza.

Il profilo dell'istanza assegnato alle EC2 istanze del tuo ambiente deve avere l'autorizzazione a leggere l'oggetto chiave dal bucket specificato. Verifica che il profilo dell'istanza abbia l'autorizzazione per leggere l'oggetto in IAM e che le autorizzazioni del bucket e dell'oggetto non impediscano l'accesso al profilo dell'istanza.

Per visualizzare le autorizzazioni di un bucket

1. Apri la console di gestione Amazon S3.
2. Scegli un bucket.
3. Scegli Properties (Proprietà), quindi Permissions (Autorizzazioni).
4. Verifica che il tuo account sia un assegnatario per il bucket con autorizzazione di lettura.
5. Se una policy del bucket è collegata, seleziona Bucket policy (Policy bucket) per visualizzare le autorizzazioni assegnate al bucket.

Configurazione della terminazione HTTPS sul sistema di bilanciamento del carico

Per aggiornare AWS Elastic Beanstalk l'ambiente in modo da utilizzare HTTPS, è necessario configurare un listener HTTPS per il sistema di bilanciamento del carico dell'ambiente. Due tipi di sistema di bilanciamento del carico supportano un listener HTTPS: Classic Load Balancer e Application Load Balancer.

Puoi utilizzare la console Elastic Beanstalk o un file di configurazione per impostare un listener protetto a cui assegnare il certificato.

> **Note**
>
> Single-instance gli ambienti non dispongono di un sistema di bilanciamento del carico e non supportano la terminazione HTTPS sul sistema di bilanciamento del carico.

Configurazione di un listener protetto utilizzando la console Elastic Beanstalk

Per assegnare un certificato al sistema di bilanciamento del carico del tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Load balancer (Sistema di bilanciamento del carico), scegliere Edit (Modifica).

   > **Note**
   >
   > Se la categoria di configurazione di Load balancer (Sistema di bilanciamento del carico) non presenta un pulsante Edit (Modifica), il tuo ambiente non dispone di un sistema di bilanciamento del carico.

5. Nella pagina Modify load balancer (Modifica sistema di bilanciamento del carico), la procedura varia a seconda del tipo di sistema di bilanciamento del carico associato al tuo ambiente.
   - Classic Load Balancer
     a. Scegli Add listener (Aggiungi listener).
     b. Nella finestra di dialogo Classic Load Balancer listener (Listener del sistema di bilanciamento del carico) configura le impostazioni seguenti:
        - In Listener port (Porta listener), indica la porta per il traffico in entrata, generalmente `443`.
        - In Listener protocol (Protocollo listener), scegli HTTPS.
        - In Instance port (Porta istanza), digita `80`.
        - In Instance protocol (Protocollo istanza), scegli HTTP.

    - In SSL certificate (Certificato SSL), scegli il tuo certificato.

  c. Scegliere Aggiungi.

- Application Load Balancer

  a. Scegli Add listener (Aggiungi listener).

  b. Nella finestra di dialogo Application Load Balancer listener (Listener del sistema di bilanciamento del carico applicazioni) configura le impostazioni seguenti:

    - In Port (Porta), indica la porta per il traffico in entrata, generalmente 443.
    - In Protocol (Protocollo), scegli HTTPS.
    - In SSL certificate (Certificato SSL), scegli il tuo certificato.

  c. Scegliere Aggiungi.

> **Note**
>
> Se il menu a discesa di Classic Load Balancer o Application Load Balancer non mostra alcun certificato, devi creare o caricare un certificato per il nome di dominio personalizzato in AWS Certificate Manager (ACM) (scelta consigliata). In alternativa, carica un certificato in IAM con l' AWS CLI.

- Network Load Balancer

  a. Scegli Add listener (Aggiungi listener).

  b. Nella finestra di dialogo Network Load Balancer listener (Listener del Sistema di bilanciamento del carico di rete) specifica la porta di traffico in entrata, generalmente la 443.

  c. Scegliere Aggiungi.

6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurazione di un listener protetto utilizzando un file di configurazione

Puoi configurare un listener sicuro sul tuo sistema di bilanciamento del carico utilizzando uno dei seguenti file di configurazione.

Example. ebextensions/securelistener-clb.config

Utilizza questo esempio quando l'ambiente dispone di un sistema Classic Load Balancer. L'esempio usa le opzioni dello spazio dei nomi `aws:elb:listener` per configurare un listener HTTPS sulla porta 443 con il certificato specificato e inoltrare il traffico decrittografato alle istanze nel tuo ambiente sulla porta 80.

```
option_settings:
  aws:elb:listener:443:
    SSLCertificateId: arn:aws:acm:us-east-2:1234567890123:certificate/
####################################
    ListenerProtocol: HTTPS
    InstancePort: 80
```

Sostituisci il testo evidenziato con l'ARN del tuo certificato. Il certificato può essere uno che hai creato o caricato in AWS Certificate Manager (ACM) (preferito) o uno che hai caricato su IAM con. AWS CLI

Per ulteriori informazioni sulle opzioni di configurazione di Classic Load Balancer, consulta Spazi dei nomi di configurazione di Classic Load Balancer.

Example. ebextensions/securelistener-alb.config

Utilizza questo esempio quando l'ambiente dispone di un sistema Application Load Balancer. L'esempio utilizza le opzioni dello spazio dei nomi `aws:elbv2:listener` per configurare un listener HTTPS sulla porta 443 con il certificato specificato. Il listener instrada il traffico verso il processo predefinito.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
####################################
```

Example. ebextensions/securelistener-nlb.config

Utilizza questo esempio quando l'ambiente dispone di un sistema Network Load Balancer. L'esempio utilizza le opzioni nello spazio dei `aws:elbv2:listener` nomi per configurare un listener TLS sulla porta 443 con un certificato SSL. Il listener gestisce la connessione sicura e indirizza il traffico verso il processo predefinito.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
    Protocol: TLS
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
##################################
```

Configurazione di un gruppo di sicurezza

Se configuri il sistema di bilanciamento del carico per inoltrare il traffico verso una porta dell'istanza diversa dalla porta 80, devi aggiungere una regola al gruppo di sicurezza che consente il traffico in entrata sulla porta dell'istanza dal sistema di bilanciamento del carico. Se crei il tuo ambiente in un VPC personalizzato, questa regola viene aggiunta automaticamente da Elastic Beanstalk.

Puoi aggiungere la regola inserendo una chiave `Resources` a un file di configurazione nella directory `.ebextensions` per la tua applicazione.

Il seguente file di configurazione di esempio aggiunge una regola in ingresso per il gruppo di sicurezza `AWSEBSecurityGroup`. Ciò consente il traffico sulla porta 1000 dal gruppo di sicurezza del sistema di bilanciamento del carico.

Example. ebextensions/sg- ingresso da lb.config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 1000
      FromPort: 1000
      SourceSecurityGroupId: {"Fn::GetAtt" : ["AWSEBLoadBalancerSecurityGroup",
 "GroupId"]}
```

Configurazione della terminazione HTTPS sull'istanza

Puoi utilizzare i file di configurazione per configurare il server proxy che trasferisce il traffico verso la tua applicazione in modo da terminare le connessioni HTTPS. Questa funzione è utile se desideri utilizzare HTTPS con un ambiente di istanza singola oppure se configuri il sistema di bilanciamento del carico per inoltrare il traffico senza decrittografarlo.

Per abilitare HTTPS, devi consentire il traffico in entrata sulla porta 443 verso l' EC2 istanza su cui è in esecuzione l'applicazione Elastic Beanstalk. A tale scopo, utilizza la `Resources` chiave nel file di configurazione per aggiungere una regola per la porta 443 alle regole di ingresso per il gruppo di sicurezza del gruppo. AWSEBSecurity

Il seguente frammento aggiunge una regola di ingresso al gruppo di sicurezza `AWSEBSecurityGroup` che apre la porta 443 a tutto il traffico per un ambiente di istanza singola:

`.ebextensions/https-instance-securitygroup.config`

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

In un ambiente con bilanciamento del carico con un Amazon Virtual Private Cloud (Amazon VPC) predefinito, puoi modificare questa policy per accettare solo il traffico proveniente dal sistema di bilanciamento del carico. Consulta Configurazione della crittografia end-to-end in un ambiente Elastic Beanstalk con bilanciamento del carico per un esempio.

Piattaforme

- Terminazione di HTTPS su istanze EC2 in cui è in esecuzione Docker
- Terminazione di HTTPS sulle EC2 istanze che eseguono Go
- Terminazione di HTTPS su istanze EC2 che eseguono Java SE
- Terminazione di HTTPS nelle istanze EC2 che eseguono Node.js
- Terminazione di HTTPS su EC2 istanze che eseguono PHP
- Terminazione di HTTPS su istanze EC2 che eseguono Python
- Terminazione di HTTPS su istanze EC2 che eseguono Ruby
- Terminazione di HTTPS su istanze EC2 che eseguono Tomcat
- Terminazione di HTTPS su istanze Amazon EC2 che eseguono .NET Core su Linux
- Terminazione di HTTPS su EC2 istanze Amazon che eseguono.NET

Terminazione di HTTPS su istanze EC2 in cui è in esecuzione Docker

Per i container Docker, è possibile utilizzare un file di configurazione per abilitare HTTPS.

Aggiungi il frammento di codice seguente al file di configurazione, sostituendo il materiale del certificato e della chiave privata come indicato, quindi salvalo nella directory `.ebextensions` del bundle di origine. Il file di configurazione esegue le seguenti attività:

- La chiave `files` crea i seguenti file sull'istanza:

  `/etc/nginx/conf.d/https.conf`

  Configura il server nginx. Questo file viene caricato quando si avvia il servizio nginx.

  `/etc/pki/tls/certs/server.crt`

  Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del certificato.

  > **Note**
  >
  > YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

  Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example.ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      # HTTPS Server

      server {
        listen 443;
        server_name localhost;

        ssl on;
        ssl_certificate /etc/pki/tls/certs/server.crt;
        ssl_certificate_key /etc/pki/tls/certs/server.key;

        ssl_session_timeout 5m;

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_prefer_server_ciphers on;

        location / {
          proxy_pass http://docker;
          proxy_http_version 1.1;

          proxy_set_header Connection "";
          proxy_set_header Host $host;
          proxy_set_header X-Real-IP $remote_addr;
          proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
          proxy_set_header X-Forwarded-Proto https;
        }
      }

  /etc/pki/tls/certs/server.crt:
```

```
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS sulle EC2 istanze che eseguono Go

Per i tipi di container Go, devi attivare HTTPS con un file di configurazione e un file di configurazione nginx che configura il server nginx per l'utilizzo di HTTPS.

Aggiungi il seguente frammento al file di configurazione, sostituendo il certificato e il materiale della chiave privata come indicato, e salvalo nella directory `.ebextensions` del bundle di origine. Il file di configurazione esegue le seguenti attività:

- La chiave `Resources` attiva la porta 443 sul gruppo di sicurezza utilizzato dall'istanza dell'ambiente.
- La chiave `files` crea i seguenti file sull'istanza:

  `/etc/pki/tls/certs/server.crt`

  Crea il file di certificato sull'istanza. *`certificate file contents`*Sostituiscilo con il contenuto del tuo certificato.

  > **Note**
  >
  > YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

  Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

Inserisci quanto segue in un file con estensione `.conf` nella directory `.ebextensions/nginx/conf.d/` del bundle di origine (ad esempio, `.ebextensions/nginx/conf.d/https.conf`). *`app_port`*Sostituiscilo con il numero di porta su cui l'applicazione è in ascolto. Questo esempio configura il server nginx per l'ascolto sulla porta 443 tramite SSL. Per ulteriori informazioni su questi file di configurazione sulla piattaforma Go, consulta Configurazione del server proxy.

Example. ebextensions/nginx/conf.d/https.conf

```
# HTTPS server
```

```
server {
    listen       443;
    server_name  localhost;

    ssl                  on;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout  5m;

    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers   on;

    location / {
        proxy_pass  http://localhost:app_port;
        proxy_set_header   Connection "";
        proxy_http_version 1.1;
        proxy_set_header        Host            $host;
        proxy_set_header        X-Real-IP       $remote_addr;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto https;
    }
}
```

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su istanze EC2 che eseguono Java SE

Per i tipi di container Java SE, devi attivare HTTPS con un file di configurazione .ebextensions e un file di configurazione nginx che configura il server nginx per l'utilizzo di HTTPS.

Tutte le AL2 piattaforme AL2023/supportano una funzionalità di configurazione proxy uniforme. Per ulteriori informazioni sulla configurazione del server proxy sulle versioni della piattaforma che eseguono AL2023/AL2, consultaConfigurazione del proxy inverso.

Aggiungi il seguente frammento al file di configurazione, sostituendo i segnaposti di certificato e chiave privata come indicato, e salvalo nella directory `.ebextensions`. Il file di configurazione esegue le seguenti attività:

- La chiave `files` crea i seguenti file sull'istanza:

  `/etc/pki/tls/certs/server.crt`

  Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del certificato.

  > **Note**
  >
  > YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

  Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
```

```
    -----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

Inserisci quanto segue in un file con estensione `.conf` nella directory `.ebextensions/nginx/conf.d/` del bundle di origine (ad esempio, `.ebextensions/nginx/conf.d/https.conf`). *`app_port`*Sostituiscilo con il numero di porta su cui l'applicazione è in ascolto. Questo esempio configura il server nginx per l'ascolto sulla porta 443 tramite SSL. Per ulteriori informazioni su questi file di configurazione sulla piattaforma Java SE, consulta Configurazione del server proxy.

Example. ebextensions/nginx/conf.d/https.conf

```
# HTTPS server
```

```
server {
    listen       443;
    server_name  localhost;

    ssl                  on;
    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout  5m;

    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers   on;

    location / {
        proxy_pass  http://localhost:app_port;
        proxy_set_header   Connection "";
        proxy_http_version 1.1;
        proxy_set_header        Host            $host;
        proxy_set_header        X-Real-IP       $remote_addr;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto https;
    }
}
```

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS nelle istanze EC2 che eseguono Node.js

Il seguente file di configurazione di esempio estende la configurazione nginx predefinita per ascoltare sulla porta 443 e SSL/TLS terminare le connessioni con un certificato pubblico e una chiave privata.

Se hai configurato l'ambiente per il reporting sullo stato migliorato, è necessario configurare nginx per generare i log di accesso. A tale scopo, rimuovi i commenti dal blocco di linee sotto il commento la cui dicitura è `# For enhanced health...` rimuovendo i caratteri iniziali #.

Example.ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      # HTTPS server

      server {
          listen       443;
          server_name  localhost;

          ssl                  on;
          ssl_certificate      /etc/pki/tls/certs/server.crt;
          ssl_certificate_key  /etc/pki/tls/certs/server.key;

          ssl_session_timeout  5m;

          ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
          ssl_prefer_server_ciphers   on;

          # For enhanced health reporting support, uncomment this block:

          #if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
          #    set $year $1;
          #    set $month $2;
          #    set $day $3;
          #    set $hour $4;
```

```
        #}
        #access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
healthd;
        #access_log  /var/log/nginx/access.log  main;

        location / {
            proxy_pass  http://nodejs;
            proxy_set_header   Connection "";
            proxy_http_version 1.1;
            proxy_set_header        Host            $host;
            proxy_set_header        X-Real-IP       $remote_addr;
            proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header        X-Forwarded-Proto https;
        }
    }

 /etc/pki/tls/certs/server.crt:
   mode: "000400"
   owner: root
   group: root
   content: |
     -----BEGIN CERTIFICATE-----
     certificate file contents
     -----END CERTIFICATE-----

 /etc/pki/tls/certs/server.key:
   mode: "000400"
   owner: root
   group: root
   content: |
     -----BEGIN RSA PRIVATE KEY-----
     private key contents # See note below.
     -----END RSA PRIVATE KEY-----
```

La chiave `files` crea i seguenti file sull'istanza:

`/etc/nginx/conf.d/https.conf`

Configura il server nginx. Questo file viene caricato quando si avvia il servizio nginx.

`/etc/pki/tls/certs/server.crt`

Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del tuo certificato.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituiscilo *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su EC2 istanze che eseguono PHP

Per i tipi di container PHP, devi usare un file di configurazione per abilitare Apache HTTP Server per l'utilizzo di HTTPS.

Aggiungi il frammento di codice seguente al file di configurazione, sostituendo il materiale del certificato e della chiave privata come indicato, quindi salvalo nella directory `.ebextensions` del bundle di origine.

Il file di configurazione esegue le seguenti attività:

- La chiave `packages` utilizza yum per installare `mod24_ssl`.
- La chiave `files` crea i seguenti file sull'istanza:

  `/etc/httpd/conf.d/ssl.conf`

  Configura il server Apache. Questo file viene caricato all'avvio del servizio Apache.

  `/etc/pki/tls/certs/server.crt`

  Crea il file di certificato sull'istanza. *`certificate file contents`*Sostituiscilo con il contenuto del tuo certificato.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example.ebextensions/https-instance.config

```
packages:
  yum:
    mod24_ssl : []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      LoadModule ssl_module modules/mod_ssl.so
      Listen 443
      <VirtualHost *:443>
```

```
        <Proxy *>
          Order deny,allow
          Allow from all
        </Proxy>

        SSLEngine             on
        SSLCertificateFile    "/etc/pki/tls/certs/server.crt"
        SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
        SSLCipherSuite        EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
        SSLProtocol           All -SSLv2 -SSLv3
        SSLHonorCipherOrder   On
        SSLSessionTickets     Off

        Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains; preload"
        Header always set X-Frame-Options DENY
        Header always set X-Content-Type-Options nosniff

        ProxyPass / http://localhost:80/ retry=0
        ProxyPassReverse / http://localhost:80/
        ProxyPreserveHost on
        RequestHeader set X-Forwarded-Proto "https" early

      </VirtualHost>

  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su istanze EC2 che eseguono Python

Per i tipi di container Python che utilizzano Apache HTTP Server con l'interfaccia WSGI (Web Server Gateway Interface), devi utilizzare un file di configurazione per consentire ad Apache HTTP Server di utilizzare HTTPS.

Aggiungi il seguente frammento al file di configurazione, sostituendo i segnaposto di certificato e chiave privata come indicato, quindi salvalo nella directory `.ebextensions` del bundle di origine. Il file di configurazione esegue le seguenti attività:

- La chiave `packages` utilizza yum per installare `mod_ssl`.
- La chiave `files` crea i seguenti file sull'istanza:

`/etc/httpd/conf.d/ssl.conf`

Configura il server Apache. Se la tua applicazione non è denominata `application.py`, sostituisci il testo evidenziato nel valore per `WSGIScriptAlias` con il percorso locale alla tua applicazione. Ad esempio, un'applicazione django può essere in `django/wsgi.py`. La posizione deve corrispondere al valore dell'opzione `WSGIPath` impostata per il tuo ambiente.

A seconda dei requisiti dell'applicazione, potrebbe essere necessario aggiungere anche altre directory al parametro python-path.

`/etc/pki/tls/certs/server.crt`

Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del certificato.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituiscilo *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

- La chiave `container_commands` interrompe il servizio httpd dopo che tutto è stato configurato in modo che il servizio utilizzi il nuovo certificato e il nuovo file `https.conf`.

> **Note**
>
> L'esempio funziona solo in ambienti che utilizzano la piattaforma Python.

Example.ebextensions/https-instance.config

```
packages:
  yum:
    mod_ssl: []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      LoadModule wsgi_module modules/mod_wsgi.so
      WSGIPythonHome /var/app/venv/staging-LQM1lest
      WSGISocketPrefix run/wsgi
      WSGIRestrictEmbedded On
      Listen 443
      <VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile "/etc/pki/tls/certs/server.crt"
        SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

        Alias /static/ /var/app/current/static/
        <Directory /var/app/current/static>
        Order allow,deny
        Allow from all
        </Directory>

        WSGIScriptAlias / /var/app/current/application.py

        <Directory /var/app/current>
        Require all granted
        </Directory>
```

```
        WSGIDaemonProcess wsgi-ssl processes=1 threads=15 display-name=%{GROUP} \
          python-path=/var/app/current \
          python-home=/var/app/venv/staging-LQM1lest \
          home=/var/app/current \
          user=webapp \
          group=webapp
        WSGIProcessGroup wsgi-ssl

      </VirtualHost>

  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----

container_commands:
  01killhttpd:
    command: "killall httpd"
  02waitforhttpddeath:
    command: "sleep 3"
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

> **Nota per gli ambienti Amazon Linux 2023**
>
> Su Amazon Linux 2023, `mod_wsgi` deve essere installato separatamente in quanto non è disponibile negli archivi dei pacchetti. Per le istruzioni di installazione, vedi mod_wsgi su PyPI.

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su istanze EC2 che eseguono Ruby

Per i tipi di container Ruby, il modo in cui abiliti HTTPS dipende dal tipo di server di applicazioni utilizzato.

Argomenti

- Configurare HTTPS per Ruby con Puma
- Configurare HTTPS per Ruby con Passenger

Configurare HTTPS per Ruby con Puma

Per i tipi di container Ruby che utilizzano Puma come server di applicazioni, utilizza un file di configurazione per abilitare il protocollo HTTPS.

Aggiungi il frammento di codice seguente al file di configurazione, sostituendo il materiale del certificato e della chiave privata come indicato, quindi salvalo nella directory `.ebextensions` del bundle di origine. Il file di configurazione esegue le seguenti attività:

- La chiave `files` crea i seguenti file sull'istanza:

  `/etc/nginx/conf.d/https.conf`

  Configura il server nginx. Questo file viene caricato quando si avvia il servizio nginx.

  `/etc/pki/tls/certs/server.crt`

  Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del certificato.

  > **Note**
  >
  > YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

  Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

  ```
      -----BEGIN CERTIFICATE-----
  certificate file contents
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  first intermediate certificate
  -----END CERTIFICATE-----
  -----BEGIN CERTIFICATE-----
  second intermediate certificate
  -----END CERTIFICATE-----
  ```

  `/etc/pki/tls/certs/server.key`

  Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example.ebextensions/https-instance.config

```
files:
  /etc/nginx/conf.d/https.conf:
    content: |
      # HTTPS server

      server {
          listen       443;
          server_name  localhost;

          ssl                  on;
          ssl_certificate      /etc/pki/tls/certs/server.crt;
          ssl_certificate_key  /etc/pki/tls/certs/server.key;

          ssl_session_timeout  5m;

          ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
          ssl_prefer_server_ciphers   on;

          location / {
              proxy_pass  http://my_app;
              proxy_set_header        Host            $host;
              proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
              proxy_set_header        X-Forwarded-Proto https;
          }

          location /assets {
            alias /var/app/current/public/assets;
            gzip_static on;
            gzip on;
            expires max;
            add_header Cache-Control public;
          }

          location /public {
            alias /var/app/current/public;
            gzip_static on;
            gzip on;
            expires max;
            add_header Cache-Control public;
          }
      }
```

```
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Configurare HTTPS per Ruby con Passenger

Per i tipi di container Ruby che utilizzano Passenger come server di applicazioni, utilizza un file di configurazione e un file JSON per abilitare il protocollo HTTPS.

Per configurare HTTPS per Ruby con Passenger

1. Aggiungi il frammento di codice seguente al file di configurazione, sostituendo il materiale del certificato e della chiave privata come indicato, quindi salvalo nella directory `.ebextensions` del bundle di origine. Il file di configurazione esegue le seguenti attività:

   - La chiave `files` crea i seguenti file sull'istanza:

     `/etc/pki/tls/certs/server.crt`

     Crea il file di certificato sull'istanza. Sostituiscilo con il contenuto del certificato. *`certificate file contents`*

     > **Note**
     >
     > YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

     Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
second intermediate certificate
-----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example Frammento .Ebextensions per la configurazione di HTTPS per Ruby con Passenger

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

2. Crea un file di testo e aggiungi il seguente JSON al file. Salvalo nella directory principale del bundle di origine con il nome `passenger-standalone.json`. Questo file JSON configura Passenger per l'utilizzo di HTTPS.

> **Important**
>
> Questo file JSON non deve contenere un contrassegno ordine di byte (BOM, Byte Order Mark). In caso contrario, la libreria JSON di Passenger non leggerà il file correttamente e il servizio Passenger non verrà avviato.

Example passenger-standalone.json

```
{
  "ssl" : true,
  "ssl_port" : 443,
  "ssl_certificate" : "/etc/pki/tls/certs/server.crt",
  "ssl_certificate_key" : "/etc/pki/tls/certs/server.key"
}
```

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su istanze EC2 che eseguono Tomcat

Per i tipi di container Tomcat, utilizza un file di configurazione per autorizzare Apache HTTP Server a utilizzare il protocollo HTTPS quando svolge la funzione di proxy inverso per Tomcat.

Aggiungi il frammento di codice seguente al file di configurazione, sostituendo il materiale del certificato e della chiave privata come indicato, quindi salvalo nella directory `.ebextensions` del bundle di origine. Il file di configurazione esegue le seguenti attività:

- La chiave `files` crea i seguenti file sull'istanza:

`/etc/pki/tls/certs/server.crt`

Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del certificato.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituiscilo *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

`/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh`

Crea uno script hook post-distribuzione per riavviare il servizio httpd.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
```

```
      -----END RSA PRIVATE KEY-----

  /opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh:
    mode: "000755"
    owner: root
    group: root
    content: |
      #!/usr/bin/env bash
      sudo service httpd restart
```

È inoltre necessario configurare il server proxy dell'ambiente affinché ascolti la porta 443. La seguente configurazione Apache 2.4 aggiunge un listener sulla porta 443. Per ulteriori informazioni, consulta Configurazione del server proxy.

Example. ebextensions/httpd/conf.d/ssl.conf

```
Listen 443
<VirtualHost *:443>
  ServerName server-name
  SSLEngine on
  SSLCertificateFile "/etc/pki/tls/certs/server.crt"
  SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"

  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-ssl-error_log

</VirtualHost>
```

Il fornitore del certificato può includere certificati intermedi che puoi installare per migliorare la compatibilità con i client mobili. Configura Apache con il bundle di un'autorità di certificazione (CA) intermedia aggiungendo il seguente file di configurazione SSL (vedi Estensione e sovrascrittura della configurazione predefinita di Apache — Amazon Linux AMI () AL1 per la posizione):

- Nei contenuti del file `ssl.conf` specifica il file della catena:

```
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCertificateChainFile "/etc/pki/tls/certs/gd_bundle.crt"
SSLCipherSuite        EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

- Aggiungi una nuova voce alla chiave `files` con i contenuti dei certificati intermedi:

```
files:
  /etc/pki/tls/certs/gd_bundle.crt:
    mode: "000400"
    owner: root
    group: root
    content: |
      -----BEGIN CERTIFICATE-----
      First intermediate certificate
      -----END CERTIFICATE-----
      -----BEGIN CERTIFICATE-----
      Second intermediate certificate
      -----END CERTIFICATE-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
```

```
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su istanze Amazon EC2 che eseguono .NET Core su Linux

Per i tipi di container .NET Core su Linux devi abilitare HTTPS con un file di configurazione `.ebextensions` e un file di configurazione nginx che configura il server nginx per l'utilizzo di HTTPS.

Aggiungi il seguente frammento al file di configurazione, sostituendo i segnaposti di certificato e chiave privata come indicato, e salvalo nella directory `.ebextensions`. Il file di configurazione esegue le seguenti attività:

- La chiave `files` crea i seguenti file sull'istanza:

  `/etc/pki/tls/certs/server.crt`

  Crea il file di certificato sull'istanza. Sostituiscilo *`certificate file contents`* con il contenuto del certificato.

  > **Note**
  >
  > YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

  Se disponi di certificati intermedi, includili in `server.crt` dopo il certificato del sito.

```
    -----BEGIN CERTIFICATE-----
certificate file contents
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
first intermediate certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

```
    second intermediate certificate
    -----END CERTIFICATE-----
```

`/etc/pki/tls/certs/server.key`

Crea il file della chiave privata sull'istanza. Sostituisci *`private key contents`* con il contenuto della chiave privata utilizzata per creare la richiesta di certificato o il certificato autofirmato.

Example.ebextensions/https-instance.config

```
files:
  /etc/pki/tls/certs/server.crt:
    content: |
      -----BEGIN CERTIFICATE-----
      certificate file contents
      -----END CERTIFICATE-----

  /etc/pki/tls/certs/server.key:
    content: |
      -----BEGIN RSA PRIVATE KEY-----
      private key contents # See note below.
      -----END RSA PRIVATE KEY-----
```

> **Note**
>
> Non utilizzare un file di configurazione che contiene la tua chiave per il controllo delle origini. Una volta eseguito il test e verificato che la configurazione funzioni, archivia la tua chiave privata in Amazon S3 e modifica la configurazione per scaricarla durante la distribuzione. Per istruzioni, consulta Archiviazione di chiavi private in modo sicuro in Amazon S3.

Inserisci quanto segue in un file con estensione `.conf` nella directory `.platform/nginx/conf.d/` del bundle di origine (ad esempio, `.platform/nginx/conf.d/https.conf`). *`app_port`*Sostituiscilo con il numero di porta su cui l'applicazione è in ascolto. Questo esempio configura il server nginx per l'ascolto sulla porta 443 tramite SSL. Per ulteriori informazioni su questi file di configurazione sulla piattaforma .NET Core su Linux, consulta the section called “Server proxy”.

Example. platform/nginx/conf.d/https.conf

```
# HTTPS server

server {
    listen       443 ssl;
    server_name  localhost;

    ssl_certificate      /etc/pki/tls/certs/server.crt;
    ssl_certificate_key  /etc/pki/tls/certs/server.key;

    ssl_session_timeout  5m;

    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers   on;

    location / {
        proxy_pass  http://localhost:app_port;
        proxy_set_header   Connection "";
        proxy_http_version 1.1;
        proxy_set_header        Host            $host;
        proxy_set_header        X-Real-IP       $remote_addr;
        proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header        X-Forwarded-Proto https;
    }
}
```

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Terminazione di HTTPS su EC2 istanze Amazon che eseguono.NET

Il seguente file di configurazione crea ed esegue uno PowerShell script di Windows che esegue le seguenti attività:

- Verifica la presenza di un'associazione di certificati HTTPS esistente alla porta 443.
- Ottiene il certificato PFX da un bucket Amazon S3.

> **Note**
>
> Aggiungi una `AmazonS3ReadOnlyAccess` policy per accedere `aws-elasticbeanstalk-ec2-role` al certificato SSL nel bucket Amazon S3.

- Ottiene la password da. Gestione dei segreti AWS

> **Note**
>
> Aggiunge un'istruzione `aws-elasticbeanstalk-ec2-role` che consenta l'`secretsmanager:GetSecretValue`azione relativa al segreto che contiene la password del certificato

- Installa il certificato.
- Associa il certificato alla porta 443.

> **Note**
>
> Per rimuovere gli endpoint HTTP (porta 80), includi il comando `Remove-WebBinding` nella sezione Rimuovi associazione HTTP dell'esempio.

Example.ebextensions/ .config https-instance-dotnet

```
files:
  "C:\\certs\\install-cert.ps1":
    content: |
```

```
     import-module webadministration
     ## Settings - replace the following values with your own
     $bucket = "amzn-s3-demo-bucket"  ## S3 bucket name
     $certkey = "example.com.pfx"    ## S3 object key for your PFX certificate
     $secretname = "example_secret"  ## Gestione dei segreti AWS name for a secret
that contains the certificate's password
     ##

     # Set variables
     $certfile = "C:\cert.pfx"
     $pwd = Get-SECSecretValue -SecretId $secretname | select -expand SecretString

     # Clean up existing binding
     if ( Get-WebBinding "Default Web Site" -Port 443 ) {
       Echo "Removing WebBinding"
       Remove-WebBinding -Name "Default Web Site" -BindingInformation *:443:
     }
     if ( Get-Item -path IIS:\SslBindings\0.0.0.0!443 ) {
       Echo "Deregistering WebBinding from IIS"
       Remove-Item -path IIS:\SslBindings\0.0.0.0!443
     }

     # Download certificate from S3
     Read-S3Object -BucketName $bucket -Key $certkey -File $certfile

     # Install certificate
     Echo "Installing cert..."
     $securepwd = ConvertTo-SecureString -String $pwd -Force -AsPlainText
     $cert = Import-PfxCertificate -FilePath $certfile cert:\localMachine\my -Password
$securepwd

     # Create site binding
     Echo "Creating and registering WebBinding"
     New-WebBinding -Name "Default Web Site" -IP "*" -Port 443 -Protocol https
     New-Item -path IIS:\SslBindings\0.0.0.0!443 -value $cert -Force

     ## Remove the HTTP binding
     ## (optional) Uncomment the following line to unbind port 80
     # Remove-WebBinding -Name "Default Web Site" -BindingInformation *:80:
     ##

     # Update firewall
     netsh advfirewall firewall add rule name="Open port 443" protocol=TCP
localport=443 action=allow dir=OUT
```

```
commands:
  00_install_ssl:
    command: powershell -NoProfile -ExecutionPolicy Bypass -file C:\\certs\\install-
cert.ps1
```

In un ambiente con una singola istanza, devi anche modificare il gruppo di sicurezza dell'istanza per consentire il traffico sulla porta 443. Il seguente file di configurazione recupera l'ID del gruppo di sicurezza utilizzando una CloudFormation funzione e vi aggiunge una regola.

Example.ebextensions/ https-instance-single .config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Per un ambiente con bilanciamento del carico, è possibile configurare il sistema di bilanciamento del carico in modo da far passare il traffico sicuro in modo inalterato oppure decriptarlo e ricrittografarlo per la crittografia. end-to-end

Configurazione della crittografia end-to-end in un ambiente Elastic Beanstalk con bilanciamento del carico

L'arresto di connessioni sicure al sistema di bilanciamento del carico e l'utilizzo di HTTP nel back-end potrebbero essere sufficienti per la tua applicazione. Il traffico di rete tra risorse AWS non può essere ascoltato da istanze che non fanno parte della connessione, anche se sono in esecuzione con lo stesso account.

Tuttavia, se sviluppi un'applicazione che deve essere conforme a normative esterne rigorose, potresti dover proteggere tutte le connessioni di rete. Puoi utilizzare la console Elastic Beanstalk o i file di configurazione per connettere in modo sicuro il sistema di bilanciamento del carico dell'ambiente Elastic Beanstalk alle istanze di back-end e soddisfare questi requisiti. La seguente procedura è incentrata sui file di configurazione.

Anzitutto aggiungi un listener sicuro al sistema di bilanciamento del carico, se non lo hai già fatto.

Devi anche configurare le istanze nell'ambiente perché siano in ascolto sulla porta sicura e terminare le connessioni HTTPS. La configurazione varia in base alla piattaforma. Per istruzioni, consulta Configurazione della terminazione HTTPS sull'istanza. Puoi utilizzare un certificato autofirmato per le istanze EC2 senza alcun problema.

Configura quindi il listener per inoltrare il traffico tramite HTTPS sulla porta protetta utilizzata dall'applicazione. Utilizza uno dei seguenti file di configurazione, a seconda del tipo di sistema di bilanciamento del carico utilizzato dall'ambiente.

`.ebextensions/https-reencrypt-clb.config`

Utilizza questo file di configurazione con un sistema Classic Load Balancer. Oltre a configurare il sistema di bilanciamento del carico, il file di configurazione modifica anche i controlli dello stato predefiniti, in modo che utilizzino la porta 443 e HTTPS per garantire che il sistema di bilanciamento del carico sia in grado di connettersi in modo sicuro.

```
option_settings:
  aws:elb:listener:443:
    InstancePort: 443
    InstanceProtocol: HTTPS
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: HTTPS:443/
```

`.ebextensions/https-reencrypt-alb.config`

Utilizza questo file di configurazione con un sistema Application Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
    Protocol: HTTPS
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
    Protocol: HTTPS
```

`.ebextensions/https-reencrypt-nlb.config`

Utilizza questo file di configurazione con un sistema Network Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

L'opzione `DefaultProcess` è così denominata perché un sistema Application Load Balancer può avere listener non predefiniti sulla stessa porta per il traffico verso percorsi specifici (vedi Application Load Balancer per i dettagli). Per un sistema Network Load Balancer, l'opzione specifica l'unico processo di destinazione per questo listener.

In questo esempio, abbiamo denominato il processo `https` perché ascolta per proteggere il traffico (HTTPS). Il listener invia il traffico al processo sulla porta designata utilizzando il protocollo TCP. Puoi anche configurare un listener TLS sul tuo Network Load Balancer per gestire la connessione sicura al load balancer. Per ulteriori informazioni, consulta Network Load Balancer.

> **Note**
>
> La console Elastic Beanstalk e la CLI EB applicano i valori consigliati per le opzioni precedenti. Rimuovi queste impostazioni se desideri utilizzare i file di configurazione per configurare le stesse opzioni. Per informazioni dettagliate, consulta Valori consigliati.

Nell'attività successiva, è necessario modificare il gruppo di sicurezza del sistema di bilanciamento del carico per consentire il traffico. A seconda dell'Amazon Virtual Private Cloud (Amazon VPC) in cui viene avviato l'ambiente, ovvero il VPC predefinito o un VPC personalizzato, varia il gruppo di sicurezza del sistema di bilanciamento del carico. In un VPC predefinito, Elastic Load Balancing fornisce un gruppo di sicurezza predefinito che può essere utilizzato da tutti i sistemi di bilanciamento del carico. In un Amazon VPC creato da te, Elastic Beanstalk fornisce un gruppo di sicurezza che viene utilizzato dal sistema di bilanciamento del carico.

Per supportare entrambi gli scenari, puoi creare un gruppo di sicurezza e indicare a Elastic Beanstalk di utilizzarlo. Il file di configurazione seguente crea un gruppo di sicurezza e lo collega al sistema di bilanciamento del carico.

`.ebextensions/https-lbsecuritygroup.config`

```
option_settings:
```

```
  # Use the custom security group for the load balancer
  aws:elb:loadbalancer:
    SecurityGroups: '`{ "Ref" : "loadbalancersg" }`'
    ManagedSecurityGroup: '`{ "Ref" : "loadbalancersg" }`'

Resources:
  loadbalancersg:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: load balancer security group
      VpcId: vpc-########
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
      SecurityGroupEgress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
```

Sostituisci il testo evidenziato con l'ID VPC predefinito o personalizzato. L'esempio precedente include l'ingresso e l'uscita sulla porta 80 per permettere connessioni HTTP. Puoi rimuovere queste proprietà se vuoi permettere solo connessioni sicure.

Infine, aggiungi regole di ingresso e di uscita per permettere la comunicazione sulla porta 443 tra il gruppo di sicurezza del sistema di bilanciamento del carico e il gruppo di sicurezza delle istanze.

`.ebextensions/https-backendsecurity.config`

```
Resources:
  # Add 443-inbound to instance security group (AWSEBSecurityGroup)
  httpsFromLoadBalancerSG:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
```

```
      FromPort: 443
      SourceSecurityGroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
  # Add 443-outbound to load balancer security group (loadbalancersg)
  httpsToBackendInstances:
    Type: AWS::EC2::SecurityGroupEgress
    Properties:
      GroupId: {"Fn::GetAtt" : ["loadbalancersg", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      DestinationSecurityGroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

Eseguendo questa operazione separatamente dalla creazione del gruppo di sicurezza, potrai limitare i gruppi di sicurezza di origine e destinazione senza creare una dipendenza circolare.

Una volta completate tutte le attività sopra descritte, il sistema di bilanciamento del carico si connette alle istanze di back-end in modo sicuro utilizzando HTTPS. Al sistema di bilanciamento del carico non importa se il certificato dell'istanza è autofirmato o rilasciato da un'autorità di certificazione attendibile, ma accetta semplicemente qualsiasi certificato presentato.

Puoi modificare questo comportamento aggiungendo delle policy al sistema di bilanciamento del carico che gli indichino come attendibile un solo certificato specifico. Il file di configurazione seguente crea due policy. Una policy specifica un certificato pubblico e l'altra indica al sistema di bilanciamento del carico di considerare affidabile solo il certificato per le connessioni alla porta 443 dell'istanza.

`.ebextensions/https-backendauth.config`

```
option_settings:
  # Backend Encryption Policy
  aws:elb:policies:backendencryption:
    PublicKeyPolicyNames: backendkey
    InstancePorts:  443
  # Public Key Policy
  aws:elb:policies:backendkey:
    PublicKey: |
      -----BEGIN CERTIFICATE-----
      ################################################################
      ################################################################
      ################################################################
      ################################################################
      ################################################
      -----END CERTIFICATE-----
```

Sostituisci il testo evidenziato con il contenuto del certificato pubblico dell'istanza EC2.

Configurazione del sistema di bilanciamento del carico dell'ambiente per il passthrough TCP

Se non desideri che il sistema di bilanciamento del carico del tuo AWS Elastic Beanstalk ambiente decrittografi il traffico HTTPS, puoi configurare il listener sicuro per inoltrare le richieste alle istanze di backend così come sono.

> **Important**
>
> La configurazione del load balancer per inoltrare il traffico HTTPS senza decrittografarlo presenta uno svantaggio. Il load balancer non è in grado di visualizzare le richieste crittografate e quindi non può ottimizzare il routing o riportare le metriche di risposta.

Per prima cosa configura le EC2 istanze del tuo ambiente per terminare HTTPS. Testa la configurazione su un ambiente di istanza singola per assicurarti che tutto funzioni prima di aggiungere un sistema di bilanciamento del carico alla combinazione.

Aggiungi un file di configurazione al tuo progetto per configurare un listener sulla porta 443 che trasferisce pacchetti TCP così come sono alla porta 443 sulle istanze di back-end:

`.ebextensions/https-lb-passthrough.config`

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: TCP
    InstancePort: 443
    InstanceProtocol: TCP
```

In un Amazon Virtual Private Cloud (Amazon VPC) predefinito devi anche aggiungere una regola al gruppo di sicurezza delle istanze per consentire il traffico in entrata sulla porta 443 dal sistema di bilanciamento del carico:

`.ebextensions/https-instance-securitygroup.config`

```
Resources:
  443inboundfromloadbalancer:
    Type: AWS::EC2::SecurityGroupIngress
```

```
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
 "SourceSecurityGroup.GroupName"] }
```

In un VPC personalizzato, Elastic Beanstalk aggiorna automaticamente la configurazione del gruppo di sicurezza.

Configurazione del reindirizzamento da HTTP a HTTPS

Questo argomento descrive come gestire il traffico HTTP verso l'applicazione se gli utenti finali continuano ad avviarlo. A questo scopo, configura il reindirizzamento da HTTP a HTTPS, talvolta definito come forzamento di HTTPS.

Per configurare il reindirizzamento, è necessario configurare prima l'ambiente per gestire il traffico HTTPS. Quindi reindirizza il traffico HTTP a HTTPS. Queste due fasi sono descritte nelle seguenti sottosezioni.

Configurare l'ambiente per gestire il traffico HTTPS

A seconda della configurazione del bilanciamento del carico dell'ambiente, effettua una delle seguenti operazioni:

- Ambiente con bilanciamento del carico: configura il sistema di bilanciamento del carico per terminare HTTPS.
- Ambiente a istanza singola: configura la tua applicazione per terminare le connessioni HTTPS nell'istanza. Questa configurazione dipende dalla piattaforma dell'ambiente.

Reindirizzare il traffico HTTP a HTTPS

Per reindirizzare il traffico HTTP verso HTTPS per la tua applicazione, puoi configurare i server Web sulle istanze del tuo ambiente oppure configurare l'Application Load Balancer dell'ambiente.

Configura i server web dell'istanza

Questo metodo funziona su qualsiasi ambiente di server Web. Configura i server Web sulle tue EC2 istanze Amazon per rispondere al traffico HTTP con uno stato di risposta di reindirizzamento HTTP.

Questa configurazione dipende dalla piattaforma dell'ambiente. Trova la cartella per la tua piattaforma nella `https-redirect`raccolta su GitHub e usa il file di configurazione di esempio contenuto in quella cartella.

Se l'ambiente utilizza i controlli dello stato di Elastic Load Balancing, il sistema di bilanciamento del carico richiede un'istanza integra per rispondere ai messaggi di controllo dello stato HTTP con risposte HTTP 200 (OK). Di conseguenza, il server Web non deve reindirizzare questi messaggi ad HTTPS. I file di configurazione di esempio in `https-redirect` gestiscono questo requisito correttamente.

Configura il sistema di bilanciamento del carico

Questo metodo funziona se si dispone di un ambiente con carico bilanciato che utilizza un Application Load Balancer. Un Application Load Balancer può inviare risposte di reindirizzamento man mano che arriva il traffico HTTP. In questo caso, non è necessario configurare il reindirizzamento sulle istanze dell'ambiente.

Abbiamo due file di configurazione di esempio GitHub che mostrano come configurare un Application Load Balancer per il reindirizzamento.

- Il file di configurazione `alb-http-to-https-redirection-full.config` crea un listener HTTPS sulla porta 443 e modifica il listener predefinito della porta 80 per reindirizzare il traffico HTTP in ingresso a HTTPS.
- Il file `alb-http-to-https-redirection.config`di configurazione prevede la definizione del listener 443. Per definirlo, puoi utilizzare i namespace di configurazione Elastic Beanstalk standard o la console Elastic Beanstalk. Quindi apporta la modifica al listener della porta 80 per il reindirizzamento.

Piattaforme Elastic Beanstalk

AWS Elastic Beanstalk offre una varietà di piattaforme su cui è possibile creare le proprie applicazioni. Tu progetti l'applicazione Web per una di queste piattaforme ed Elastic Beanstalk distribuisce il codice sulla versione della piattaforma che hai selezionato per creare un ambiente applicativo attivo.

Elastic Beanstalk fornisce piattaforme per diversi linguaggi di programmazione, server applicazioni e container Docker. Alcune piattaforme dispongono di più versioni supportate contemporaneamente.

Argomenti

- Glossario delle piattaforme Elastic Beanstalk
- Modello di responsabilità condivisa per la manutenzione della piattaforma Elastic Beanstalk
- Policy di supporto per la piattaforma Elastic Beanstalk
- Pianificazione del rilascio della piattaforma Elastic Beanstalk
- Piattaforme supportate da Elastic Beanstalk
- Piattaforme Elastic Beanstalk Linux
- Estensione delle piattaforme Elastic Beanstalk Linux

Glossario delle piattaforme Elastic Beanstalk

Di seguito sono riportati i termini chiave relativi alle AWS Elastic Beanstalk piattaforme e al loro ciclo di vita.

Runtime

Il software di runtime specifico del linguaggio di programmazione (framework, librerie, interprete, vm e così via) richiesto per eseguire il codice dell'applicazione.

Componenti Elastic Beanstalk

Componenti software aggiunti da Elastic Beanstalk a una piattaforma per abilitare la funzionalità Elastic Beanstalk. Ad esempio, l'agente di stato migliorato è necessario per raccogliere e segnalare informazioni sullo stato.

Platform (Piattaforma)

Una combinazione di sistema operativo, runtime, server Web, server di applicazioni e componenti Elastic Beanstalk. Le piattaforme forniscono componenti disponibili per eseguire l'applicazione.

Versione piattaforma

Una combinazione di versioni specifiche di sistema operativo, runtime, server Web, server di applicazioni e componenti Elastic Beanstalk. Crea un ambiente Elastic Beanstalk basato su una versione della piattaforma e distribuisci qui l'applicazione.

Una versione della piattaforma dispone di un numero di versione semantico di formato X.Y.Z, in cui X è la versione principale, Y è la versione secondaria e Z è la versione della patch.

Una versione della piattaforma può trovarsi in uno dei seguenti stati:

- Consigliata: la versione più recente della piattaforma disponibile in un ramo della piattaforma supportata. Questa versione contiene la maggior parte dei up-to-date componenti ed è consigliata per l'uso in ambienti di produzione. Quando Elastic Beanstalk rilascia una nuova versione della piattaforma, la nuova versione sostituisce la versione precedente e diventa la versione della piattaforma consigliata per il ramo della piattaforma corrispondente.
- Non consigliata: qualsiasi versione della piattaforma che non sia la versione più recente nel ramo della piattaforma. Sebbene queste versioni possano rimanere funzionali, consigliamo vivamente di eseguire l'aggiornamento alla versione più recente della piattaforma. Puoi utilizzare gli aggiornamenti gestiti della piattaforma per aiutarti a rimanere up-to-date automaticamente.

Puoi verificare se è consigliata una versione della piattaforma utilizzando il comando AWS CLI describe-platform-version e controllando il `PlatformLifecycleState` campo.

Ramo della piattaforma

Una linea di versioni di piattaforma che condividono versioni specifiche (in genere principali) di alcuni dei loro componenti, come il sistema operativo (OS), il runtime o i componenti Elastic Beanstalk. Ad esempio: Python 3.13 in esecuzione su Amazon Linux 2023 a 64 bit; IIS 10.0 in esecuzione su Windows Server 2025 a 64 bit. Le filiali della piattaforma ricevono aggiornamenti sotto forma di nuove versioni della piattaforma. Ogni versione successiva della piattaforma in una filiale è un aggiornamento della precedente.

La versione consigliata in ogni ramo della piattaforma supportata è disponibile incondizionatamente per la creazione dell'ambiente. Le versioni precedenti della piattaforma rimangono accessibili agli account con ambienti attivi o terminati che le utilizzano nel momento

in cui sono state sostituite da una nuova versione. Le versioni precedenti della piattaforma non dispongono della maggior parte dei up-to-date componenti e non ne è consigliato l'uso.

> **Note**
>
> Se hai bisogno di accedere a versioni precedenti della piattaforma oltre alla disponibilità standard sopra descritta, puoi contattare il AWS Support Center per ricevere assistenza.

Un ramo di piattaforma può trovarsi in uno dei seguenti stati:

- Supported (Supportato): un ramo della piattaforma corrente. Consiste interamente di componenti supportati. I componenti supportati non hanno raggiunto la fine del ciclo di vita (EOL), come indicato dai rispettivi fornitori. Riceve aggiornamenti della piattaforma in corso ed è consigliato per l'uso in ambienti di produzione. Per un elenco dei rami di piattaforma supportati, consulta le piattaforme supportate da Elastic Beanstalk nella guida Platforms (Piattaforme)AWS Elastic Beanstalk .
- Beta: un'anteprima del ramo della piattaforma pre-release. È di natura sperimentale. Potrebbe ricevere aggiornamenti della piattaforma in corso per un periodo, ma non ha supporto a lungo termine. Un ramo della piattaforma beta non è raccomandato per l'uso in ambienti di produzione. Usalo solo per la valutazione. Per un elenco dei rami della piattaforma beta, consulta Versioni della piattaforma Elastic Beanstalk in versione beta pubblica nella guida Platforms (Piattaforme)AWS Elastic Beanstalk .
- Obsoleto: ramo della piattaforma in cui uno o più componenti (come il runtime o il sistema operativo) si stanno avvicinando alla fine del ciclo di vita (EOL) o hanno raggiunto la fine del ciclo di vita, come indicato dai rispettivi fornitori. Sebbene un ramo della piattaforma obsoleto continui a ricevere nuove versioni della piattaforma fino alla data di ritiro, i componenti che hanno raggiunto l'EOL non ricevono aggiornamenti. Ad esempio, se una versione di runtime raggiunge l'EOL, il ramo della piattaforma verrà contrassegnato come obsoleto ma continuerà a ricevere gli aggiornamenti del sistema operativo fino alla data di pensionamento del ramo della piattaforma. Il ramo della piattaforma non continuerà a ricevere aggiornamenti alla versione runtime EOL. Si sconsiglia l'uso di un ramo della piattaforma obsoleto.
- Ritirato: un ramo della piattaforma che non riceve più aggiornamenti. Le filiali della piattaforma ritirate non sono disponibili per creare nuovi ambienti Elastic Beanstalk utilizzando la console Elastic Beanstalk. Se il tuo ambiente utilizza un ramo di piattaforma ritirato, devi eseguire l'aggiornamento a un ramo della piattaforma supportato per continuare a ricevere gli aggiornamenti. Non è consigliabile utilizzare un ramo della piattaforma ritirato. Per ulteriori

dettagli sulle filiali della piattaforma ritirate, consulta. the section called “Policy di supporto per la piattaforma” Per un elenco delle filiali della piattaforma previste per il pensionamento, consulta Pianificazione delle filiali della piattaforma in pensione. Per visualizzare le filiali della piattaforma ritirate in passato, consulta Cronologia delle filiali della piattaforma ritirate.

Se l'ambiente utilizza un ramo di piattaforma obsoleto o ritirato, si consiglia di aggiornarlo a una versione di piattaforma in un ramo di piattaforma supportato. Per informazioni dettagliate, vedi the section called “Aggiornamenti della piattaforma”.

Puoi verificare lo stato di un ramo della piattaforma utilizzando il comando AWS CLI describe-platform-version e controllando il `PlatformBranchLifecycleState` campo.

Aggiornamento della piattaforma

Una versione di una nuova versione della piattaforma che contiene aggiornamenti ad alcuni componenti della piattaforma: sistema operativo, runtime, server web, server applicazioni e componenti Elastic Beanstalk. Quando Elastic Beanstalk rilascia una nuova versione della piattaforma, la nuova versione sostituisce la versione precedente e diventa la versione della piattaforma consigliata per il ramo della piattaforma corrispondente. Gli aggiornamenti della piattaforma seguono la tassonomia delle versioni semantiche e possono avere tre livelli:

- Aggiornamento principale: un aggiornamento che contiene modifiche non compatibili con le versioni della piattaforma esistenti. Potrebbe essere necessario modificare l'applicazione per farla funzionare correttamente su una nuova versione principale. Un aggiornamento principale dispone di un nuovo numero di versione della piattaforma principale.
- Aggiornamento minore: aggiornamento che presenta modifiche retrocompatibili con le versioni precedenti della piattaforma nella maggior parte dei casi. A seconda dell'applicazione, potrebbe essere necessario modificarla per farla funzionare correttamente su una nuova versione secondaria. Un aggiornamento secondario dispone di un nuovo numero di versione della piattaforma secondario.
- Aggiornamento delle patch: un aggiornamento costituito da release di manutenzione (correzioni di bug, aggiornamenti di sicurezza e miglioramenti delle prestazioni) che sono compatibili con le versioni precedenti della piattaforma esistente. Un aggiornamento delle patch dispone di un nuovo numero di versione della piattaforma patch.

Managed Updates (Aggiornamenti gestiti)

Una caratteristica di Elastic Beanstalk che applica automaticamente aggiornamenti delle patch e secondari al sistema operativo (OS), runtime, server Web, server applicazioni e componenti Elastic Beanstalk per una versione della piattaforma supportata da Elastic Beanstalk. Un

aggiornamento gestito applica una versione della piattaforma più recente nello stesso ramo della piattaforma all'ambiente. Puoi configurare gli aggiornamenti gestiti per applicare solo aggiornamenti delle patch o aggiornamenti delle patch secondari. Puoi anche disabilitare completamente gli aggiornamenti gestiti.

Per ulteriori informazioni, consulta Aggiornamenti gestiti della piattaforma.

Modello di responsabilità condivisa per la manutenzione della piattaforma Elastic Beanstalk

AWS e i nostri clienti condividono la responsabilità di raggiungere un elevato livello di sicurezza e conformità dei componenti software. Questo modello condiviso riduce gli oneri operativi.

Per i dettagli, consulta il modello di responsabilità AWS condivisa.

AWS Elastic Beanstalk ti aiuta a svolgere la tua parte del modello di responsabilità condivisa fornendo una funzionalità di aggiornamento gestito. Questa caratteristica applica automaticamente aggiornamenti secondari e patch per una versione della piattaforma supportata da Elastic Beanstalk. Se un aggiornamento gestito non riesce, Elastic Beanstalk ti segnala l'errore in modo che potrai agire immediatamente.

Per ulteriori informazioni, consulta Aggiornamenti gestiti della piattaforma.

Inoltre, Elastic Beanstalk effettua le seguenti operazioni:

- Pubblica la sua policy di supporto piattaforma e pianificazione di ritiro per i prossimi 12 mesi.
- Rilascia patch, aggiornamenti secondari e principali dei componenti del sistema operativo (OS), runtime, server applicazioni e server Web di solito entro 30 giorni dalla loro disponibilità. Elastic Beanstalk è responsabile della creazione di aggiornamenti per i componenti Elastic Beanstalk presenti nelle versioni della piattaforma supportate. Tutti gli altri aggiornamenti vengono forniti direttamente da loro fornitori (proprietari o community).

Annunciamo tutti gli aggiornamenti alle nostre piattaforme supportate nelle note di rilascio nella guida delle note di rilascio di AWS Elastic Beanstalk . Nella guida Piattaforme AWS Elastic Beanstalk è inoltre disponibile un elenco di tutte le piattaforme supportate e dei relativi componenti, insieme a una cronologia delle piattaforme. Per ulteriori informazioni, consulta Piattaforme supportate e cronologia dei componenti.

È responsabilità dell'utente:

- Aggiorna tutti i componenti che controlli (identificati come Cliente nel modello di responsabilità AWS condivisa). Questo include garantire la sicurezza dell'applicazione, dei dati e dei componenti richiesti dall'applicazione e che sono stati scaricati.
- Assicurarsi che gli ambienti Elastic Beanstalk siano in esecuzione su una versione della piattaforma supportata e migra l'eventuale ambiente in esecuzione su una versione della piattaforma ritirata in una versione supportata.
- Se utilizzi un'immagine macchina Amazon (AMI) personalizzata per il tuo ambiente Elastic Beanstalk, applica patch, gestisci e testa l'AMI personalizzata in modo che rimanga aggiornata e compatibile con una versione della piattaforma Elastic Beanstalk supportata. Per ulteriori informazioni sulla gestione degli ambienti con un'AMI personalizzata, vedereUtilizzo di un'immagine Amazon Machine Image (AMI) personalizzata nell'ambiente Elastic Beanstalk.
- Risolvere tutti i problemi che si verificano nei tentativi di aggiornamento gestiti non riusciti e riprovare a eseguire l'aggiornamento.
- Applicare manualmente patch a sistema operativo, runtime, server applicazioni e server Web se sono stati esclusi gli aggiornamenti gestiti di Elastic Beanstalk. Ciò è possibile applicando manualmente gli aggiornamenti della piattaforma o applicando direttamente le patch ai componenti su tutte le risorse dell'ambiente pertinenti.
- Gestisci la sicurezza e la conformità di tutti AWS i servizi che utilizzi al di fuori di Elastic Beanstalk secondo AWS il modello di responsabilità condivisa.

Policy di supporto per la piattaforma Elastic Beanstalk

Elastic Beanstalk supporta i rami della piattaforma che ancora ricevono aggiornamenti secondari e patch dai provider (proprietari o community). Per una definizione completa dei termini correlati, consulta Glossario delle piattaforme Elastic Beanstalk.

Rami della piattaforma ritirati

Quando un componente di un ramo della piattaforma supportato viene contrassegnato come End of Life (EOL) dal fornitore, Elastic Beanstalk contrassegna il ramo della piattaforma come ritirato. I componenti di un ramo della piattaforma includono: sistema operativo (OS), versione del linguaggio di runtime, server delle applicazioni o server Web.

Una volta che un ramo della piattaforma viene contrassegnato come ritirato, si applicano le seguenti politiche:

- Elastic Beanstalk smette di fornire aggiornamenti di manutenzione, inclusi gli aggiornamenti di sicurezza.
- Elastic Beanstalk non fornisce più supporto tecnico per le filiali della piattaforma dismesse.
- Elastic Beanstalk non rende più il ramo della piattaforma disponibile ai nuovi clienti Elastic Beanstalk per le implementazioni in nuovi ambienti. Esiste un periodo di tolleranza di 90 giorni dalla data di ritiro pubblicata per i clienti esistenti con ambienti attivi in esecuzione su rami di piattaforma ritirata.

> **Note**
>
> Un ramo della piattaforma ritirato non sarà disponibile nella console Elastic Beanstalk. Tuttavia, sarà disponibile tramite EB CLI AWS CLI e EB API per i clienti che dispongono di ambienti esistenti basati sulla filiale della piattaforma ritirata. I clienti esistenti possono anche utilizzare le console dell'ambiente Clone e Rebuild.

Per un elenco delle filiali della piattaforma il cui ritiro è previsto, consulta l'argomento sulla Ritiro della pianificazione dei rami della piattaforma pianificazione della piattaforma Elastic Beanstalk che segue.

Per ulteriori informazioni su cosa aspettarsi quando il ramo della piattaforma del tuo ambiente verrà ritirato, consulta. Domande frequenti sul ritiro della piattaforma

Oltre il periodo di tolleranza di 90 giorni

La nostra politica per le filiali della piattaforma ritirate non rimuove l'accesso agli ambienti né elimina le risorse. Tuttavia, i clienti esistenti che utilizzano un ambiente Elastic Beanstalk su una filiale di piattaforma dismessa devono essere consapevoli dei rischi che ciò comporta. Tali ambienti possono finire in una situazione imprevedibile, perché Elastic Beanstalk non è in grado di fornire aggiornamenti di sicurezza, supporto tecnico o hotfix per le filiali della piattaforma dismesse a causa del fatto che il fornitore ha contrassegnato il termine EOL dei componenti.

Ad esempio, una vulnerabilità di sicurezza dannosa e critica potrebbe emergere in un ambiente in esecuzione su un ramo di piattaforma ritirata. Oppure un'azione API EB potrebbe smettere di funzionare per l'ambiente se nel tempo diventa incompatibile con il servizio Elastic Beanstalk. La possibilità di incorrere in questi tipi di rischi aumenta più a lungo rimane attivo un ambiente su una piattaforma ritirato. Per continuare a beneficiare degli importanti miglioramenti di sicurezza,

prestazioni e funzionalità offerti dai provider dei componenti nelle versioni più recenti, ti invitiamo ad aggiornare tutti gli ambienti Elastic Beanstalk a una versione della piattaforma supportata.

Se la tua applicazione dovesse riscontrare problemi durante l'esecuzione su un ramo della piattaforma dismesso e non riesci a migrarla su una piattaforma supportata, dovrai prendere in considerazione altre alternative. Le soluzioni alternative includono l'incapsulamento dell'applicazione in un'immagine Docker per eseguirlo come container Docker. Ciò consentirebbe a un cliente di utilizzare una qualsiasi delle nostre soluzioni Docker, come le nostre piattaforme AL2 Elastic AL2 Beanstalk 023/Docker, o altri servizi basati su Docker come Amazon ECS o Amazon EKS. Le alternative non Docker includono il nostro AWS CodeDeploy servizio, che consente la personalizzazione completa dei runtime desiderati.

Pianificazione del rilascio della piattaforma Elastic Beanstalk

Oltre al rilascio mensile a cadenza periodica delle nuove versioni di Platform Branch, la nostra manutenzione delle release include anche i seguenti processi:

- Rilascio di nuovi rami della piattaforma: in genere introducono una nuova versione principale di un linguaggio di runtime, un sistema operativo o un server di applicazioni.
- Ritiro delle filiali della piattaforma: dobbiamo ritirare una filiale della piattaforma quando uno dei suoi componenti raggiunge la fine del ciclo di vita (EOL). Per ulteriori informazioni sulla nostra politica per le filiali in pensione, consulta Policy di supporto per la piattaforma Elastic Beanstalk

Argomenti

- Risorse per la pianificazione
- Prossime versioni di Platform Branch
- Ritiro della pianificazione dei rami della piattaforma
- Cronologia dei rami di piattaforma ritirati
- Cronologia del server e del sistema operativo ritirati

Risorse per la pianificazione

Le seguenti risorse possono aiutarti a pianificare la manutenzione e il supporto per la tua applicazione in esecuzione su una piattaforma Elastic Beanstalk.

- AWS Elastic Beanstalk Guida alle piattaforme: questa guida fornisce un elenco dettagliato dei componenti per ciascuna delle nostre filiali della piattaforma. Fornisce inoltre una cronologia della piattaforma per data di rilascio con gli stessi dettagli. Questa guida può informarti quando sono cambiati componenti specifici del ramo della tua piattaforma. Se l'applicazione inizia a comportarsi in modo diverso, puoi anche fare un riferimento incrociato alla data dell'evento nella guida alla piattaforma per vedere se ci sono state modifiche alla piattaforma che potrebbero aver influito sulla tua applicazione.
- AWS Elastic Beanstalk Note di rilascio: le nostre note di rilascio annunciano tutte le versioni della nostra piattaforma, sia minori che principali. Ciò include gli aggiornamenti mensili della piattaforma, le versioni di sicurezza, gli hotfix e gli annunci di ritiro. È possibile abbonarsi ai nostri feed RSS dalla documentazione delle note di rilascio.

Prossime versioni di Platform Branch

La tabella seguente elenca le prossime filiali della piattaforma Elastic Beanstalk e la data di rilascio prevista. Queste date sono provvisorie e soggette a modifiche.

Versione runtime /ramo della piattaforma	Sistema operativo	Data di rilascio prevista
Python 3.15	Amazon Linux 2023	Novembre 2026
Node.js 26	Amazon Linux 2023	Novembre 2026
.NET 11	Amazon Linux 2023	dicembre 2026
PHP 8.6	Amazon Linux 2023	gennaio 2027
Ruby 4.1	Amazon Linux 2023	febbraio 2027

Ritiro della pianificazione dei rami della piattaforma

La tabella seguente elenca le filiali della piattaforma Elastic Beanstalk il cui ritiro è previsto, poiché alcuni componenti stanno raggiungendo la fine del ciclo di vita (EOL). Tutte le filiali AL2 basate sulle piattaforme hanno una data di pensionamento non oltre il 30 giugno 2026, quando Amazon Linux 2 raggiungerà l'EOL. Per ulteriori informazioni su Amazon Linux 2, consulta Amazon Linux 2 FAQs.

Per un elenco più dettagliato delle filiali di piattaforme che verranno ritirate e che include i relativi componenti specifici, consulta la sezione relativa al ritiro AWS Elastic Beanstalk delle versioni della piattaforma nella guida Piattaforme.

Versione runtime/ramo della piattaforma	Data prevista per il pensionamento
Docker AL2	30 giugno 2026
ECS AL2	30 giugno 2026
Go 1 AL2	30 giugno 2026
Corretto 8 AL2	30 giugno 2026
Corretto 11 AL2	30 giugno 2026
Corretto 17 AL2	30 giugno 2026
Corretto 8 with Tomcat 9 AL2	30 giugno 2026
Corretto 11 with Tomcat 9 AL2	30 giugno 2026
.NET Core AL2	30 giugno 2026
Python 3.9 AL2023	31 luglio 2026
Ruby 3.2 AL2023	31 luglio 2026
Node.js 20 AL2023	31 luglio 2026
IIS 10.0 on Windows Server 2016 (& Core)	30 settembre 2026
PHP 8.2 AL2023	31 marzo 2027
.NET 9 AL2023	31 marzo 2027
.NET 8 AL2023	31 marzo 2027

Cronologia dei rami di piattaforma ritirati

Le tabelle seguenti elencano i rami della piattaforma Elastic Beanstalk che sono già in stato di ritiro. Puoi vedere una cronologia dettagliata di questi rami della piattaforma e dei relativi componenti nella guida Cronologia delle piattaforme.AWS Elastic Beanstalk

Amazon Linux 2023 (AL2023)

Versione runtime /ramo della piattaforma	Data di ritiro		
.NET 6 AL2023	8 aprile 2025		
Node.js 18 AL2023	11 agosto 2025		
PHP 8.1 AL2023	16 aprile 2026		

Amazon Linux (2AL2)

Versione runtime /ramo della piattaforma	Data di ritiro		
Corretto 11 with Tomcat 8.5 AL2	10 ottobre 2024		
Corretto 8 with Tomcat 8.5 AL2	10 ottobre 2024		
Corretto 11 with Tomcat 7 AL2	29 giugno 2022		
Corretto 8 with Tomcat 7 AL2	29 giugno 2022		
Node.js 18 AL2	11 agosto 2025		
Node.js 16 AL2	10 ottobre 2024		

Versione runtime /ramo della piattaforma	Data di ritiro		
Node.js 14 AL2	10 ottobre 2024		
Node.js 12 AL2	23 dicembre 2022		
Node.js 10 AL2	29 giugno 2022		
PHP 8.1 AL2	16 aprile 2026		
PHP 8.0 AL2	10 ottobre 2024		
PHP 7.4 AL2	9 giugno 2023		
PHP 7.3 AL2	29 giugno 2022		
PHP 7.2 AL2	29 giugno 2022		
Python 3.8 AL2	8 aprile 2025		
Python 3.7 AL2	10 ottobre 2024		
Ruby 3.0 AL2	10 ottobre 2024		
Ruby 2.7 AL2	10 ottobre 2024		
Ruby 2.6 AL2	23 dicembre 2022		
Ruby 2.5 AL2	29 giugno 2022		

AMI Amazon Linux (AL1)

Versione runtime /ramo della piattaforma	Data di ritiro		
Single Container Docker	18 luglio 2022		

Versione runtime /ramo della piattaforma	Data di ritiro		
Multicontainer Docker	18 luglio 2022		
Preconfig ured Docker - GlassFish 5.0 with Java 8	18 luglio 2022		
Go 1	18 luglio 2022		
Java 8	18 luglio 2022		
Java 7	18 luglio 2022		
Java 8 with Tomcat 8.5	18 luglio 2022		
Java 7 with Tomcat 7	18 luglio 2022		
Node.js	18 luglio 2022		
PHP 7.2 - 7.3	18 luglio 2022		
Python 3.6	18 luglio 2022		
Ruby 2,4, 2.5, 2.6 with Passenger	18 luglio 2022		
Ruby 2.4, 2.5, 2.6 with Puma	18 luglio 2022		
Go 1.3-1.10	31 ottobre 2020		
Java 6	31 ottobre 2020		

Versione runtime /ramo della piattaforma	Data di ritiro		
Node.js 4.x-8.x	31 ottobre 2020		
PHP 5.4-5.6	31 ottobre 2020		
PHP 7.0-7.1	31 ottobre 2020		
Python 2.6, 2.7, 3.4	31 ottobre 2020		
Ruby 1.9.3	31 ottobre 2020		
Ruby 2.0-2.3	31 ottobre 2020		

> **Note**
>
> Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni, consulta Domande frequenti sul ritiro della piattaforma.

Windows Server

Versione runtime /ramo della piattaforma	Data di ritiro		
IIS 8.5 in esecuzione su Windows Server (& Core) 2012 R2 a 64 bit	04 dicembre 2023		
IIS 8.5 in esecuzione su	29 giugno 2022		

Versione runtime /ramo della piattaforma	Data di ritiro		
Windows Server (& Core) (64 bit) 2012 R2 versione 0.1.0			
IIS 8.5 in esecuzione su Windows Server (& Core) (64 bit) 2012 R2 versione 1.2.0	29 giugno 2022		
IIS 10.0 in esecuzione su Windows Server 2016 (& Core) (64 bit) versione 1.2.0	29 giugno 2022		
IIS 8 in esecuzione su ramo di piattaforma di Windows Server 2012 R1 (64 bit)	22 giugno 2022		
IIS 8 in esecuzione su Windows Server 2012 R1 (64 bit) versione 0.1.0	22 giugno 2022		

Versione runtime /ramo della piattaforma	Data di ritiro		
IIS 8 in esecuzione su Windows Server 2012 R1 (64 bit) versione 1.2.0	22 giugno 2022		

> **Note**
>
> Per ulteriori informazioni sul ritiro delle ramificazioni della piattaforma Windows 2012 R2, consulta Ramificazioni della piattaforma Windows Server 2012 R2 ritirate nelle Note di rilascio di AWS Elastic Beanstalk .

Cronologia del server e del sistema operativo ritirati

Le tabelle seguenti forniscono una cronologia dei sistemi operativi, dei server delle applicazioni e dei server Web che non sono più supportati dalle piattaforme Elastic Beanstalk. Tutte le filiali della piattaforma che utilizzavano questi componenti sono ora ritirate. Le date riflettono la data di ritiro dell'ultimo ramo della piattaforma Elastic Beanstalk che includeva il componente.

Sistemi operativi

Versione SO	Data di ritiro della piattaforma		
Windows Server 2012 R2 running IIS 8.5	4 dicembre 2023		
Windows Server Core 2012 R2 running IIS 8.5	04 dicembre 2023		

Versione SO	Data di ritiro della piattaforma		
AMI Amazon Linux (AL1)	18 luglio 2022		
Windows Server 2012 R1	22 giugno 2022		
Windows Server 2008 R2	28 ottobre 2019		

Server di applicazioni

Versione del server applicazioni	Data di ritiro della piattaforma		
Tomcat 7	29 giugno 2022 per le piattaforme Amazon Linux 2 (AL2) 18 luglio 2022 per le piattaforme Amazon Linux AMI (AL1)		
Tomcat 8	31 ottobre 2020		
Tomcat 6	31 ottobre 2020		

Server Web

Versione server Web	Data di ritiro della piattaforma		
IIS 8 in esecuzione su Windows Server (64 bit)	22 giugno 2022		

Versione server Web	Data di ritiro della piattaforma		
Apache HTTP Server 2.2	31 ottobre 2020		
Nginx 1.12.2	31 ottobre 2020		

Piattaforme supportate da Elastic Beanstalk

AWS Elastic Beanstalk offre una varietà di piattaforme su cui è possibile creare le proprie applicazioni. Tu progetti l'applicazione Web per una di queste piattaforme ed Elastic Beanstalk distribuisce il codice sulla versione della piattaforma che hai selezionato per creare un ambiente applicativo attivo.

Elastic Beanstalk fornisce le risorse necessarie per eseguire l'applicazione, incluse una o più istanze Amazon. EC2 Lo stack software in esecuzione sulle EC2 istanze Amazon dipende dalla versione della piattaforma specifica che hai selezionato per il tuo ambiente.

Il nome dello stack di soluzioni per un ramo della piattaforma

È possibile utilizzare il nome dello stack di soluzioni per una determinata versione del ramo della piattaforma per avviare un ambiente con l'EB CLI, l'API Elastic Beanstalk o la CLI.AWS La guida alle AWS Elastic Beanstalk piattaforme elenca il nome dello stack di soluzioni sotto la versione del ramo della piattaforma nelle sezioni Piattaforme supportate da Elastic Beanstalk e Cronologia della piattaforma.

Per recuperare tutti i nomi dello stack di soluzioni che puoi utilizzare per creare un ambiente, utilizza l'ListAvailableSolutionStacksAPI o la `aws elasticbeanstalk list-available-solution-stacks` AWS CLI.

Puoi personalizzare e configurare il software da cui dipende la tua applicazione sulla piattaforma. Per ulteriori informazioni, consulta Personalizzazione di software su server Linux e Personalizzazione di software su server Windows. Le note di rilascio dettagliate sono disponibili per i rilasci recenti in Note di rilascio di AWS Elastic Beanstalk.

Piattaforme supportate e cronologia dei componenti

La guida alle AWS Elastic Beanstalk piattaforme elenca tutte le versioni correnti dei rami della piattaforma nella sezione Piattaforme supportate da Elastic Beanstalk. La guida alle piattaforme elenca anche una cronologia delle piattaforme per ogni piattaforma, che include un elenco delle versioni precedenti della piattaforma Branch. Per visualizzare la cronologia della piattaforma per ogni piattaforma, seleziona uno dei seguenti link.

- Docker
- Go
- Java SE
- Tomcat (che esegue Java SE)
- .NET Core su Linux
- .NET on Windows Server
- Node.js
- PHP
- Python
- Ruby

Piattaforme Elastic Beanstalk Linux

Le piattaforme Linux Elastic Beanstalk offrono una vasta gamma di funzionalità pronte all'uso. Puoi estendere le piattaforme in diversi modi per supportare la tua applicazione. Per informazioni dettagliate, vedi the section called “Estensione delle piattaforme Linux”.

La maggior parte delle piattaforme supportate da Elastic Beanstalk sono basate sul sistema operativo Linux. In particolare, queste piattaforme sono basate su Amazon Linux, una distribuzione Linux fornita da AWS. Le piattaforme Linux Elastic Beanstalk utilizzano istanze Amazon Elastic Compute Cloud EC2 (Amazon) e queste istanze eseguono Amazon Linux.

Argomenti

- Versioni di Amazon Linux supportate
- Elenco delle piattaforme Elastic Beanstalk Linux
- Flusso di lavoro di distribuzione dell'istanza

- Flusso di lavoro di implementazione di istanze per ECS in esecuzione su Amazon Linux 2 e versioni successive
- Strumenti di script di piattaforma per i tuoi ambienti Elastic Beanstalk

Versioni di Amazon Linux supportate

AWS Elastic Beanstalk supporta piattaforme basate su Amazon Linux 2 e Amazon Linux 2023.

Per ulteriori informazioni su Amazon Linux 2 e Amazon Linux 2023, consulta quanto segue:

- Amazon Linux 2 — Amazon Linux nella Amazon EC2 User Guide.
- Amazon Linux 2023: Cos'è Amazon Linux 2023? nella Guida per l'utente di Amazon Linux 2023.

Per dettagli sulle versioni della piattaforma supportate, consulta Piattaforme supportate da Elastic Beanstalk.

> **Note**
>
> È possibile migrare l'applicazione da un ramo della piattaforma AL1 Elastic AL2 Beanstalk o della piattaforma al ramo della piattaforma 023 equivalente. AL2 Per ulteriori informazioni, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Amazon Linux 2023

AWS ha annunciato la disponibilità generale di Amazon Linux 2023 a marzo del 2023. La Guida per l'utente di Amazon Linux 2023 riassume le principali differenze tra Amazon Linux 2 e Amazon Linux 2023. Per ulteriori informazioni, consulta la sezione Confronto tra Amazon Linux 2 e Amazon Linux 2023 nella Guida per l'utente.

Esiste un alto grado di compatibilità tra le piattaforme Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Sebbene ci siano alcune differenze da notare:

- Instance Metadata Service versione 1 (IMDSv1): l'impostazione predefinita dell'IMDSv1opzione Disabilita è `true` sulle AL2 piattaforme 023. L'impostazione predefinita è sulle piattaforme. `false` AL2

- strumento di istanza pkg-repo: lo pkg-repo strumento non è disponibile per gli ambienti che eseguono su AL2 piattaforme 023. Tuttavia, è comunque possibile applicare manualmente gli aggiornamenti del pacchetto e del sistema operativo a un'istanza 023. AL2 Per ulteriori informazioni, consulta Gestione dei pacchetti e degli aggiornamenti del sistema operativo nella Guida per l'utente di Amazon Linux 2023
- HTTPd Configurazione Apache: il `httpd.conf` file Apache per le piattaforme AL2 023 presenta alcune impostazioni di configurazione diverse da quelle per: AL2
  - Per impostazione predefinita, nega l'accesso all'intero file system del server. Queste impostazioni sono descritte in Protezione dei file del server per impostazione predefinita nella pagina Suggerimenti per la sicurezza del sito Web Apache.
  - Nega l'accesso alla configurazione `.htaccess` in tutte le directory, ad eccezione di quelle specificamente abilitate. Questa impostazione è descritta in Protezione delle impostazioni di sistema nella pagina Suggerimenti per la sicurezza del sito Web Apache. Nella pagina Tutorial Apache HTTP Server: file .htaccess è riportato che questa impostazione può aiutare a migliorare le prestazioni.
  - Nega l'accesso ai file con lo schema dei nomi `.ht*`. Questa impostazione impedisce ai client Web di visualizzare i file `.htaccess` e `.htpasswd`.

  È possibile modificare qualsiasi impostazione di configurazione precedente per l'ambiente. Per ulteriori informazioni, consulta Configurare HTTPD di Apache.
- Supporto per variabili di ambiente multilinea: le piattaforme AL2 023 supportano valori multilinea per variabili di ambiente e segreti nelle configurazioni dei servizi systemd. Le piattaforme Amazon Linux 2 non supportano valori di variabili di ambiente multilinea. Questo miglioramento consente di utilizzare segreti e valori di configurazione multilinea su piattaforme 023. AL2 Per ulteriori informazioni sull'utilizzo delle variabili e dei segreti di ambiente, vedere. the section called “Valori multilinea”
- CloudWatch inoltro di log personalizzato: l'agente (`awslogs`pacchetto) obsoleto di CloudWatch Logs non è disponibile sulle piattaforme 023. AL2 Se disponi di configurazioni di inoltro dei log personalizzate che installano e utilizzano l'agente obsoleto, devi aggiornare i file di configurazione per utilizzare l'`awslogs` CloudWatch agente unificato durante la migrazione da Amazon Linux 2 a 023. AL2 Per ulteriori informazioni, consulta the section called “Streaming di file di log personalizzati”.

Differenze specifiche della piattaforma

Oltre alle differenze del sistema operativo di base, esistono differenze specifiche della piattaforma tra le piattaforme di runtime Amazon Linux 2 e AL2 023:

- Ramificazione della piattaforma .NET: la strategia di ramificazione della piattaforma .NET differisce tra Amazon Linux 2 e 023. AL2 Su Amazon Linux 2, la piattaforma .NET Core mantiene una finestra rotante delle versioni principali.NET all'interno di un unico ramo della piattaforma. Nella versione AL2 023, ogni ramo della piattaforma è associato a una versione principale.NET specifica (ad esempio, .NET 9, .NET 10).

  Se distribuisci applicazioni dipendenti dal framework (applicazioni che si basano sul runtime .NET installato sulla piattaforma), devi selezionare un ramo della piattaforma che corrisponda alla versione.NET di destinazione dell'applicazione. Se distribuisci applicazioni autonome (applicazioni che raggruppano il proprio runtime .NET), puoi utilizzare qualsiasi ramo della piattaforma AL2 023 .NET indipendentemente dalla versione.NET dell'applicazione, poiché l'applicazione non dipende dal runtime installato sulla piattaforma. Per ulteriori informazioni, consulta the section called “Raggruppamento delle applicazioni”.
- Selezione della versione di Node.js: la piattaforma Node.js su Amazon Linux 2 supporta la specificazione di una versione Node.js nel `package.json` file dell'applicazione. La piattaforma Node.js su AL2 023 non supporta questa funzionalità. È necessario utilizzare la versione Node.js predefinita fornita dal ramo della piattaforma. Per ulteriori informazioni sulla gestione delle versioni di Node.js, vederethe section called “Configurazione delle dipendenze”.
- Versione del server Ruby Puma: la piattaforma Ruby su Amazon Linux 2 ignora la versione di Puma specificata nel `Gemfile.lock` file dell'applicazione e utilizza la versione Puma predefinita della piattaforma. La piattaforma Ruby su AL2 023 rispetta la versione Puma specificata in, se presente. `Gemfile.lock` Se non viene specificata alcuna versione, la piattaforma installa la versione Puma predefinita della piattaforma.
- Disponibilità dei pacchetti PHP: alcuni pacchetti disponibili sulle piattaforme PHP Amazon Linux 2 non sono disponibili sulle piattaforme PHP AL2 023:
  - Pacchetti client MySQL: *`mysql`* i pacchetti client `mysql-devel` e da riga di comando non sono installati su 023 piattaforme PHP. AL2 Se la tua applicazione richiede la connettività del database MySQL, usa `mysqli` PHP `pdo_mysql` o le estensioni, disponibili su entrambe le piattaforme.
  - Strumenti Compass e Ruby: i `rubygems` pacchetti `ruby-devel` e per il supporto del framework Compass CSS non sono installati sulle piattaforme PHP 023. AL2 Compass è obsoleto. Prendi in considerazione l'utilizzo di moderni strumenti di preelaborazione CSS come alternative.

- Strumenti di controllo della versione Go: il sistema di controllo della versione di Bazaar (`bzr`) non è disponibile sulle piattaforme AL2 023 Go. Bazaar è obsoleto e non è incluso nel repository dei pacchetti 023. AL2 Usa invece Git, Mercurial o Subversion per il controllo della versione, tutti disponibili sulle piattaforme AL2 023 Go.

Elenco delle piattaforme Elastic Beanstalk Linux

L'elenco seguente fornisce le piattaforme Linux supportate da Elastic Beanstalk per diversi linguaggi di programmazione, così come per i container Docker. Elastic Beanstalk offre piattaforme basate su Amazon Linux 2 e Amazon Linux 2023 per tutti i linguaggi. Per ulteriori informazioni su una piattaforma, seleziona il link corrispondente.

- Docker (e Docker ECS)
- Go
- Tomcat (che esegue Java SE)
- Java SE
- .NET Core su Linux
- Node.js
- PHP
- Python
- Ruby

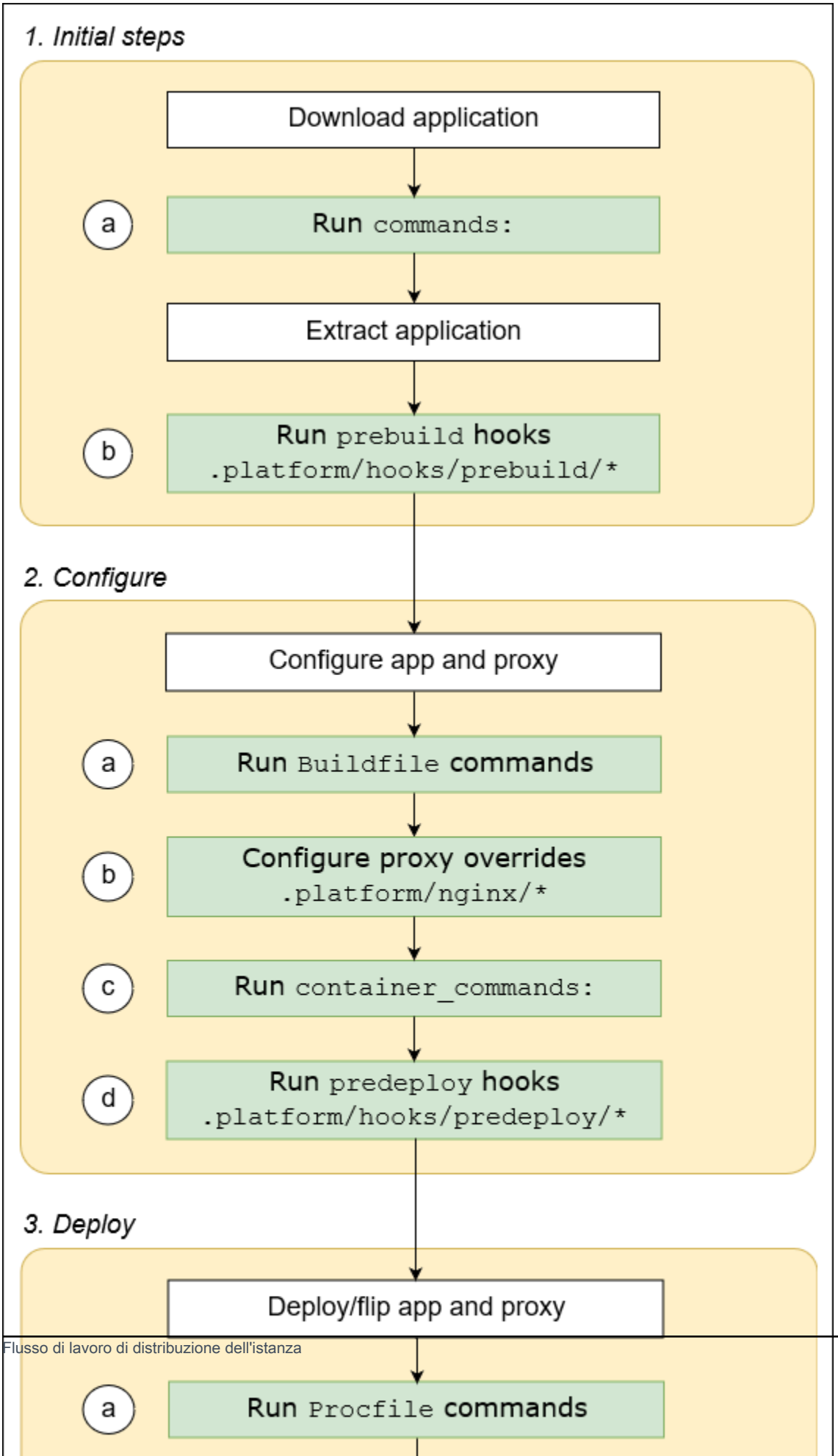
Flusso di lavoro di distribuzione dell'istanza

> **Note**
>
> Le informazioni contenute in questa sezione non si applicano alle ramificazioni della piattaforma ECS in esecuzione su Amazon Linux 2 e Amazon Linux 2023. Per ulteriori informazioni, consulta la sezione successiva, Flusso di lavoro di implementazione di istanze per ECS in esecuzione su Amazon Linux 2 e versioni successive

Grazie a molti modi per estendere la piattaforma dell'ambiente, è utile sapere cosa succede ogni volta che Elastic Beanstalk esegue il provisioning di un'istanza o esegue una distribuzione in

un'istanza. Il diagramma seguente mostra l'intero flusso di lavoro di distribuzione. Descrive le diverse fasi di una distribuzione e i passaggi eseguiti da Elastic Beanstalk in ogni fase.

> **Note**
>
> - Il diagramma non rappresenta il set completo di passaggi eseguiti da Elastic Beanstalk sulle istanze di ambiente durante la distribuzione. Forniamo questo diagramma per illustrazione, per fornirvi l'ordine e il contesto per l'esecuzione delle vostre personalizzazioni.
> - Per semplicità, il diagramma menziona solo le sottodirectory hook `.platform/hooks/*` (per le distribuzioni di applicazioni) e non le sottodirectory hook `.platform/confighooks/*` (per le distribuzioni di configurazione). Gli hook in queste ultime sottodirectory vengono eseguiti esattamente durante le stesse fasi degli hook nelle corrispondenti sottodirectory mostrate nel diagramma.


1. Initial steps
Download application
a
Run commands:
Extract application
b
Run prebuild hooks
.platform/hooks/prebuild/*
2. Configure
Configure app and proxy
a
Run Buildfile commands
b
Configure proxy overrides
.platform/nginx/*
c
Run container_commands:
d
Run predeploy hooks
.platform/hooks/predeploy/*
3. Deploy
Deploy/flip app and proxy
a
Run Procfile commands

Nell'elenco seguente vengono descritti in dettaglio le fasi e i passaggi di distribuzione.

1. Fasi iniziali

   Elastic Beanstalk scarica ed estrae la tua applicazione. Dopo ognuna di queste fasi, Elastic Beanstalk esegue una fase di estensibilità.

   a. Esegue i comandi trovati nella sezione comandi: di qualsiasi file di configurazione.

   b. Esegue tutti i file eseguibili trovati nella directory `.platform/hooks/prebuild` del bundle di origine (`.platform/confighooks/prebuild` per una distribuzione di configurazione).

2. Configura

   Elastic Beanstalk configura l'applicazione e il server proxy.

   a. Esegue i comandi trovati in `Buildfile` nel bundle di origine.

   b. Copia i file di configurazione proxy personalizzati, se ne sono presenti nella directory `.platform/nginx` del bundle sorgente, nella loro posizione di runtime.

   c. Esegue i comandi trovati nella sezione container_commands: di qualsiasi file di configurazione.

   d. Esegue tutti i file eseguibili trovati nella directory `.platform/hooks/predeploy` del bundle di origine (`.platform/confighooks/predeploy` per una distribuzione di configurazione).

3. Distribuzione

   Elastic Beanstalk distribuisce ed esegue l'applicazione e il server proxy.

   a. Esegue il comando trovato nel file `Procfile` nel bundle di origine.

   b. Esegue o riavvia il server proxy con i file di configurazione proxy personalizzati, se ne dispone.

   c. Esegue tutti i file eseguibili trovati nella directory `.platform/hooks/postdeploy` del bundle di origine (`.platform/confighooks/postdeploy` per una distribuzione di configurazione).

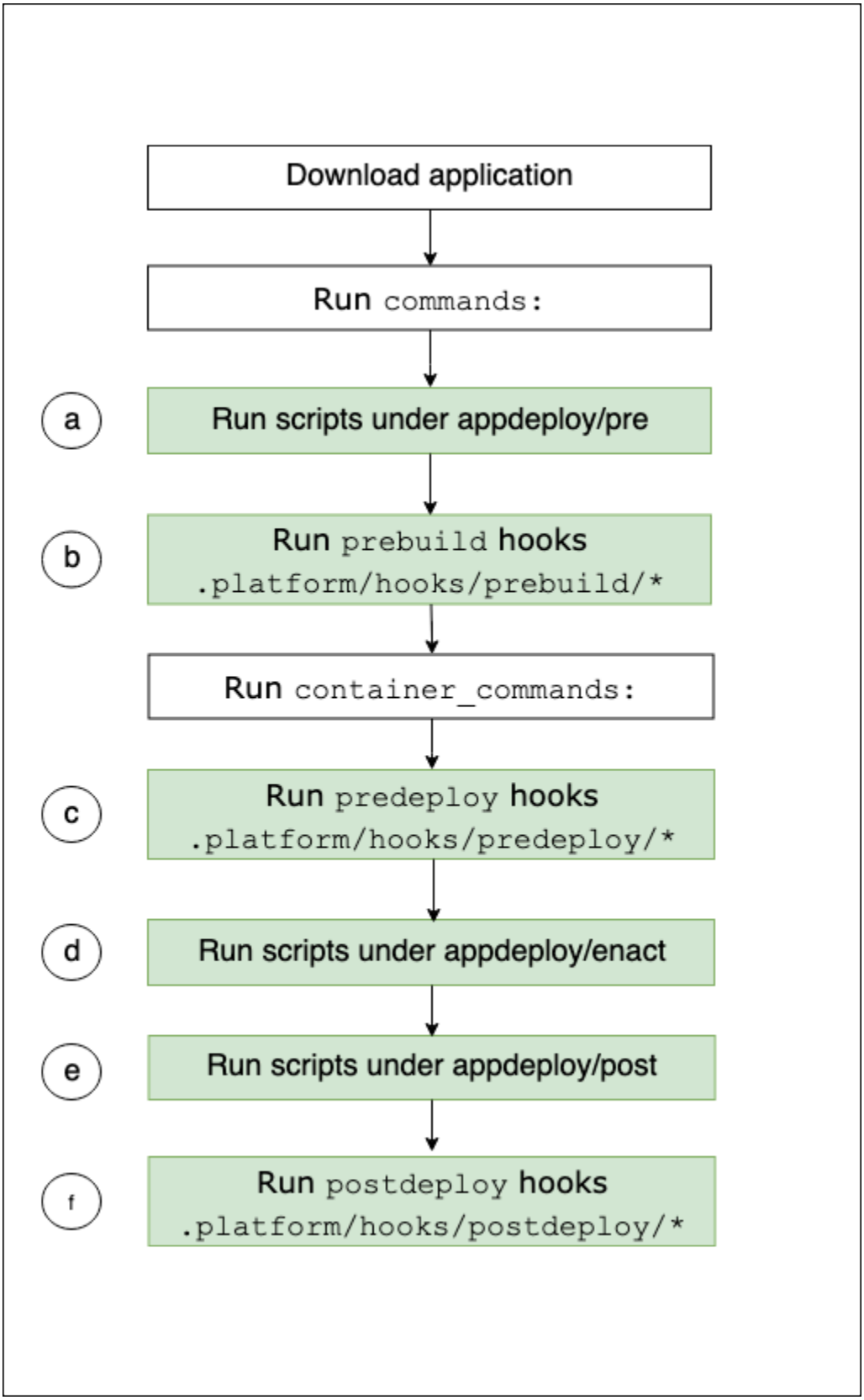
Flusso di lavoro di implementazione di istanze per ECS in esecuzione su Amazon Linux 2 e versioni successive

La sezione precedente descrive le funzionalità di estendibilità supportate durante le fasi del flusso di lavoro di distribuzione dell'applicazione. Ci sono alcune differenze per le ramificazioni della piattaforma Docker ECS in esecuzione su Amazon Linux 2 e versioni successive. Questa sezione spiega come questi concetti si applicano a questo specifico ramo della piattaforma.

Grazie a molti modi per estendere la piattaforma dell'ambiente, è utile sapere cosa succede ogni volta che Elastic Beanstalk esegue il provisioning di un'istanza o esegue una distribuzione in un'istanza. Il diagramma seguente mostra l'intero flusso di lavoro di implementazione per un ambiente basato sulle ramificazioni della piattaforma ECS in esecuzione su Amazon Linux 2 ed ECS in esecuzione su Amazon Linux 2023. Descrive le diverse fasi di una distribuzione e i passaggi eseguiti da Elastic Beanstalk in ogni fase.

A differenza del flusso di lavoro descritto nella sezione precedente, la fase di configurazione della distribuzione non supporta le seguenti funzionalità di estensibilità: comandi `Buildfile`, comandi `Procfile`, configurazione proxy inversa.

> **Note**
>
> - Il diagramma non rappresenta il set completo di passaggi eseguiti da Elastic Beanstalk sulle istanze di ambiente durante la distribuzione. Forniamo questo diagramma per illustrazione, per fornirvi l'ordine e il contesto per l'esecuzione delle vostre personalizzazioni.
> - Per semplicità, il diagramma menziona solo le sottodirectory hook `.platform/hooks/*` (per le distribuzioni di applicazioni) e non le sottodirectory hook `.platform/confighooks/*` (per le distribuzioni di configurazione). Gli hook in queste ultime sottodirectory vengono eseguiti esattamente durante le stesse fasi degli hook nelle corrispondenti sottodirectory mostrate nel diagramma.



Nell'elenco seguente vengono descritti in dettaglio le fasi e i passaggi di distribuzione.

a. Esegue tutti i file eseguibili trovati nella directory `appdeploy/pre` in `EBhooksDir`.
b. Esegue tutti i file eseguibili trovati nella directory `.platform/hooks/prebuild` del bundle di origine (`.platform/confighooks/prebuild` per una distribuzione di configurazione).
c. Esegue tutti i file eseguibili trovati nella directory `.platform/hooks/predeploy` del bundle di origine (`.platform/confighooks/predeploy` per una distribuzione di configurazione).
d. Esegue tutti i file eseguibili trovati nella directory `appdeploy/enact` in `EBhooksDir`.
e. Esegue tutti i file eseguibili trovati nella directory `appdeploy/post` in `EBhooksDir`.
f. Esegue tutti i file eseguibili trovati nella directory `.platform/hooks/postdeploy` del bundle di origine (`.platform/confighooks/postdeploy` per una distribuzione di configurazione).

Il riferimento a `EBhooksDir` rappresenta il percorso della directory degli hook della piattaforma. Per recuperare il nome del percorso della directory, utilizzare il get-config strumento di script sulla riga di comando dell'istanza dell'ambiente come mostrato:

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k EBhooksDir
```

Strumenti di script di piattaforma per i tuoi ambienti Elastic Beanstalk

Questo argomento descrive gli strumenti che AWS Elastic Beanstalk forniscono ambienti che utilizzano piattaforme Amazon Linux. Gli strumenti si trovano nelle EC2 istanze Amazon degli ambienti Elastic Beanstalk.

get-config

Utilizza `get-config` lo strumento per recuperare i valori delle variabili di ambiente in testo semplice e altre informazioni sulla piattaforma e sull'istanza. Questo strumento è disponibile in `/opt/elasticbeanstalk/bin/get-config`.

comandi get-config

Ogni comando dello strumento `get-config` restituisce un tipo specifico di informazioni. Utilizza la sintassi seguente per eseguire i comandi di uno qualsiasi degli strumenti.

```
$ /opt/elasticbeanstalk/bin/get-config command [ options ]
```

L'esempio seguente esegue il comando `environment`:

```
$ /opt/elasticbeanstalk/bin/get-config environment -k PORT
```

A seconda del comando e delle opzioni scelte, lo strumento restituisce un oggetto (JSON o YAML) con coppie chiave-valore o un singolo valore.

Puoi eseguire il test `get-config` utilizzando SSH per connetterti a un' EC2 istanza nel tuo ambiente Elastic Beanstalk.

> **Note**
>
> Quando si esegue `get-config` per il test, alcuni comandi potrebbero richiedere privilegi utente root per accedere alle informazioni sottostanti. Se viene visualizzato un errore di autorizzazione di accesso, eseguire nuovamente il comando in `sudo`.
> Non è necessario aggiungere `sudo` quando si utilizza lo strumento negli script distribuiti nell'ambiente. Elastic Beanstalk esegue tutti gli script come utente root.

Le sezioni seguenti descrivono i comandi degli strumenti.

optionsettings: opzioni di configurazione

Il comando `get-config optionsettings` restituisce un oggetto che elenca le opzioni di configurazione impostate nell'ambiente e utilizzate dalla piattaforma sulle istanze di ambiente. Sonoorganizzati per spazio dei nomi.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings
{"aws:elasticbeanstalk:application:environment":
{"JDBC_CONNECTION_STRING":""},"aws:elasticbeanstalk:container:tomcat:jvmoptions":{"JVM
 Options":"","Xms":"256m","Xmx":"256m"},"aws:elasticbeanstalk:environment:proxy":
{"ProxyServer":"nginx","StaticFiles":
[""]},"aws:elasticbeanstalk:healthreporting:system":
{"SystemType":"enhanced"},"aws:elasticbeanstalk:hostmanager":
{"LogPublicationControl":"false"}}
```

Per visualizzare un determinato valore dell'opzione di configurazione, utilizza l'opzione `--namespace` (`-n`) per specificare uno spazio dei nomi e l'opzione `--option-name` (`-o`) per specificare il nome di un'opzione.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings -
n aws:elasticbeanstalk:container:php:phpini -o memory_limit
```

```
256M
```

environment: proprietà dell'ambiente

Il `get-config environment` comando restituisce un oggetto contenente un elenco di proprietà dell'ambiente, incluse quelle configurate dall'utente e fornite da Elastic Beanstalk. Le proprietà configurate dall'utente sono definite nella console come testo normale o con l'opzione di configurazione namespace. aws:elasticbeanstalk:application:environment

```
$ /opt/elasticbeanstalk/bin/get-config environment
{"JDBC_CONNECTION_STRING":"","RDS_PORT":"3306","RDS_HOSTNAME":"anj9aw1b0tbj6b.cijbpanmxz5u.us-
west-2.rds.amazonaws.com","RDS_USERNAME":"testusername","RDS_DB_NAME":"ebdb","RDS_PASSWORD":"te
```

Ad esempio, Elastic Beanstalk fornisce proprietà di ambiente per la connessione a un'istanza database di Amazon RDS integrata (ad esempio `RDS_HOSTNAME`). Queste proprietà di connessione RDS vengono visualizzate nell'output di `get-config environment`. Tuttavia, non vengono visualizzate nell'output di `get-config optionsettings`. Questo perché non sono state impostate nelle opzioni di configurazione.

Per visualizzare una proprietà dell'ambiente specifico, utilizza l'opzione `--key` (`-k`) per specificare una chiave di proprietà.

```
$ /opt/elasticbeanstalk/bin/get-config environment -k TESTPROPERTY
testvalue
```

> **Note**
>
> Lo `get-config` strumento non è in grado di recuperare variabili di ambiente che memorizzano segreti. Per ulteriori informazioni su come recuperare a livello di codice i valori dagli archivi segreti o di parametri, vedere o. Utilizzo di Secrets Manager Utilizzo dell'archivio parametri di Systems Manager

container: valori di configurazione dell'istanza

Il comando `get-config container` restituisce un oggetto che elenca i valori di configurazione della piattaforma e dell'ambiente per le istanze di ambiente.

L'esempio seguente mostra l'output del comando su un ambiente Tomcat di Amazon Linux 2.

```
$ /opt/elasticbeanstalk/bin/get-config container
{"common_log_list":["/var/log/eb-engine.log","/var/log/eb-
hooks.log"],"default_log_list":["/var/log/nginx/access.log","/var/log/nginx/
error.log"],"environment_name":"myenv-1da84946","instance_port":"80","log_group_name_prefix":"/
aws/elasticbeanstalk","proxy_server":"nginx","static_files":
[""],"xray_enabled":"false"}
```

Per restituire il valore di una chiave specifica, utilizzare l'opzione `--key` (`-k`) per specificare la chiave.

```
$ /opt/elasticbeanstalk/bin/get-config container -k environment_name
myenv-1da84946
```

addons: valori di configurazione del componente aggiuntivo

Il comando `get-config addons` restituisce un oggetto contenente le informazioni di configurazione dei componenti aggiuntivi dell'ambiente. Utilizzalo per recuperare la configurazione di un database Amazon RDS associato all'ambiente.

```
$ /opt/elasticbeanstalk/bin/get-config addons
{"rds":{"Description":"RDS Environment variables","env":
{"RDS_DB_NAME":"ebdb","RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwvg.us-
east-2.rds.amazonaws.com","RDS_PASSWORD":"password","RDS_PORT":"3306","RDS_USERNAME":"user"}}}
```

È possibile limitare il risultato in due modi. Per recuperare i valori per un componente aggiuntivo specifico, utilizzare l'opzione `--add-on` (`-a`) per specificare il nome del componente aggiuntivo.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds
{"Description":"RDS Environment variables","env":
{"RDS_DB_NAME":"ebdb","RDS_HOSTNAME":"ea13k2wimu1dh8i.c18mnpu5rwvg.us-
east-2.rds.amazonaws.com","RDS_PASSWORD":"password","RDS_PORT":"3306","RDS_USERNAME":"user"}}
```

Per restituire il valore di una chiave specifica all'interno di un componente aggiuntivo, aggiungere l'opzione `--key` (`-k`) per specificare la chiave.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds -k RDS_DB_NAME
ebdb
```

platformconfig: valori di configurazione costanti

Il comando `get-config platformconfig` restituisce un oggetto contenente informazioni di configurazione della piattaforma che sono costanti per la versione della piattaforma. L'output è lo stesso in tutti gli ambienti che eseguono la stessa versione della piattaforma. L'oggetto di output del comando ha due oggetti incorporati:

- `GeneralConfig`: contiene informazioni che sono costanti per le versioni più recenti di tutte le ramificazioni della piattaforma Amazon Linux 2 e Amazon Linux 2023.
- `PlatformSpecificConfig`: contiene informazioni che sono costanti e specifiche per la versione della piattaforma.

L'esempio seguente mostra l'output del comando in un ambiente che utilizza il ramo della piattaforma Tomcat 8.5 che esegue Corretto 11.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig
{"GeneralConfig":{"AppUser":"webapp","AppDeployDir":"/var/app/
current/","AppStagingDir":"/var/app/
staging/","ProxyServer":"nginx","DefaultInstancePort":"80"},"PlatformSpecificConfig":
{"ApplicationPort":"8080","JavaVersion":"11","TomcatVersion":"8.5"}}
```

Per restituire il valore di una chiave specifica, utilizzare l'opzione `--key` (`-k`) per specificare la chiave. Queste chiavi sono univoche tra i due oggetti incorporati. Non è necessario specificare l'oggetto che contiene la chiave.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k AppStagingDir
/var/app/staging/
```

opzioni di output get-config

Utilizzare l'opzione `--output` per specificare il formato dell'oggetto di output. I valori validi sono `JSON` (default) e `YAML`. Questa è un'opzione globale. Devi specificarlo prima del nome del comando.

Nell'esempio seguente vengono restituiti i valori delle opzioni di configurazione in formato YAML.

```
$ /opt/elasticbeanstalk/bin/get-config --output YAML optionsettings
aws:elasticbeanstalk:application:environment:
  JDBC_CONNECTION_STRING: ""
aws:elasticbeanstalk:container:tomcat:jvmoptions:
```

```
  JVM Options: ""
  Xms: 256m
  Xmx: 256m
aws:elasticbeanstalk:environment:proxy:
  ProxyServer: nginx
  StaticFiles:
        - ""
aws:elasticbeanstalk:healthreporting:system:
  SystemType: enhanced
aws:elasticbeanstalk:hostmanager:
  LogPublicationControl: "false"
```

pkg-repo

> **Note**
>
> Lo strumento `pkg-repo` non è disponibile per ambienti basati su piattaforme Amazon Linux 2023. Tuttavia, è possibile applicare manualmente gli aggiornamenti del pacchetto e del sistema operativo a un' AL2istanza 023. Per ulteriori informazioni, consulta Gestione dei pacchetti e degli aggiornamenti del sistema operativo nella Guida per l'utente di Amazon Linux 2023

In alcune circostanze urgenti, potrebbe essere necessario aggiornare EC2 le istanze Amazon con una patch di sicurezza Amazon Linux 2 che non è ancora stata rilasciata con le versioni richieste della piattaforma Elastic Beanstalk. Non è possibile eseguire un aggiornamento manuale sugli ambienti Elastic Beanstalk per impostazione predefinita. Questo perché le versioni della piattaforma sono bloccate su una versione specifica del repository Amazon Linux 2. Questo blocco garantisce che le istanze eseguano versioni software supportate e coerenti. Per casi urgenti, lo strumento `pkg-repo` consente una soluzione alternativa per aggiornare manualmente i pacchetti yum su Amazon Linux 2 se è necessaria l'installazione in un ambiente prima del rilascio in una nuova versione della piattaforma Elastic Beanstalk.

La strumento `pkg-repo` su piattaforme Amazon Linux 2 offre la possibilità di sbloccare i repository di pacchetti `yum`. In questo modo è possibile eseguire manualmente un yum update per una patch di sicurezza. Viceversa, è possibile seguire l'aggiornamento utilizzando lo strumento per bloccare i repository di pacchetti yum per evitare ulteriori aggiornamenti. Lo `pkg-repo` strumento è disponibile nella `/opt/elasticbeanstalk/bin/pkg-repo` directory di tutte le EC2 istanze negli ambienti Elastic Beanstalk.

Le modifiche che utilizzano lo `pkg-repo` strumento vengono apportate solo sull' EC2 istanza in cui viene utilizzato lo strumento. Non influiscono su altre istanze o impediscono aggiornamenti futuri dell'ambiente. Gli esempi forniti più avanti in questo argomento spiegano come applicare le modifiche a tutte le istanze chiamando i comandi `pkg-repo` dagli script e dai file di configurazione.

> **Warning**
>
> Questo strumento è sconsigliato per la maggior parte degli utenti. Eventuali modifiche manuali applicate a una versione della piattaforma sbloccata sono considerate fuori banda. Questa opzione è valida solo per gli utenti in circostanze urgenti che sono in grado di accettare i seguenti rischi:
>
> - Non è possibile avere la certezza che i pacchetti siano coerenti in tutte le istanze degli ambienti.
> - Non è possibile avere la certezza che gli ambienti che vengono modificati utilizzando lo strumento `pkg-repo` funzionino correttamente. Non sono stati testati e verificati sulle piattaforme supportate da Elastic Beanstalk.
>
> Consigliamo vivamente di applicare le best practice che includono piani di test e backout. Per facilitare le best practice, puoi utilizzare la console Elastic Beanstalk e l'EB CLI per clonare un ambiente e scambiare un ambiente. URLs Per ulteriori informazioni sull'utilizzo di queste operazioni, consulta Implementazione blu/verde nel capitolo Gestione degli ambienti di questa guida.

Se prevedi di modificare manualmente i file di configurazione del repository yum, esegui lo strumento `pkg-repo` prima. Lo strumento `pkg-repo` potrebbe non funzionare come previsto in un ambiente Amazon Linux 2 con file di configurazione del repository yum modificati manualmente. Questo perché lo strumento potrebbe non riconoscere le modifiche apportate alla configurazione.

Per ulteriori informazioni sull'archivio di pacchetti Amazon Linux, consulta l'argomento Package repository nella Amazon EC2 User Guide.

comandi pkg-repo

Utilizza la sintassi seguente per eseguire i comandi dello strumento `pkg-repo`.

```
$ /opt/elasticbeanstalk/bin/pkg-repo command [options]
```

I comandi `pkg-repo` da eseguire sono i seguenti:

- lock: blocca il repository di pacchetti `yum` su una versione specifica
- unlock: sblocca il repository di pacchetti `yum` su una versione specifica
- status: elenca tutti i repository di pacchetti `yum` e il relativo stato di blocco corrente
- help: mostra la guida generale o la guida per un comando

Le opzioni si applicano ai comandi come segue:

- `lock`, `unlock` e `status` : opzioni: `-h`, `--help` o nessuna (predefinito).
- `help`: opzioni: `lock`, `unlock`, `status` o nessuna (predefinito).

L'esempio seguente esegue il comando unlock:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo unlock
Amazon Linux 2 core package repo successfully unlocked
Amazon Linux 2 extras package repo successfully unlocked
```

L'esempio seguente esegue il comando lock:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo lock
Amazon Linux 2 core package repo successfully locked
Amazon Linux 2 extras package repo successfully locked
```

L'esempio seguente esegue il comando status:

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo status
Amazon Linux 2 core package repo is currently UNLOCKED
Amazon Linux 2 extras package repo is currently UNLOCKED
```

L'esempio seguente esegue il comando help per il comando lock.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help lock
```

L'esempio seguente esegue il comando help per lo strumento `pkg-repo`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help
```

Puoi eseguire il test di `pkg-repo` utilizzando SSH per connetterti a un'istanza nell'ambiente Elastic Beanstalk. Un'opzione SSH è il comando della CLI di EB eb ssh.

> **Note**
>
> Lo strumento `pkg-repo` richiede privilegi utente root per l'esecuzione. Se viene visualizzato un errore di autorizzazione di accesso, eseguire nuovamente il comando in `sudo`.
> Non è necessario aggiungere `sudo` quando si utilizza lo strumento negli script o nei file di configurazione distribuiti nell'ambiente. Elastic Beanstalk esegue tutti gli script come utente root.

esempi pkg-repo

La sezione precedente fornisce esempi da riga di comando per testare una singola EC2 istanza di un ambiente Elastic Beanstalk. Questo approccio può essere utile per i test. Tuttavia, aggiorna solo un'istanza alla volta, quindi non è pratico per applicare modifiche a tutte le istanze in un ambiente.

Un approccio più pragmatico è quello di utilizzare script di hook della piattaforma o un file di configurazione `.ebextensions` per applicare le modifiche a tutte le istanze in modo coerente.

L'esempio seguente chiama `pkg-repo` da un file di configurazione nella cartella `.ebextensions`. Elastic Beanstalk esegue i comandi nel file `update_package.config` quando distribuisci il bundle di origine dell'applicazione.

```
.ebextensions
### update_package.config
```

Per ricevere la versione più recente del pacchetto docker, questa configurazione specifica il pacchetto docker nel comando yum update.

```
### update_package.config ###

commands:
  update_package:
    command: |
```

```
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update docker -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

Questa configurazione non specifica alcun pacchetto nel comando yum update. Di conseguenza, vengono applicati tutti gli aggiornamenti disponibili.

```
### update_package.config ###

commands:
  update_package:
    command: |
      /opt/elasticbeanstalk/bin/pkg-repo unlock
      yum update -y
      /opt/elasticbeanstalk/bin/pkg-repo lock
      yum clean all -y
      rm -rf /var/cache/yum
```

L'esempio seguente chiama `pkg-repo` da uno script bash come un hook della piattaforma. Elastic Beanstalk esegue il file di script `update_package.sh` che si trova nella sottodirectory `prebuild`.

```
.platform
### hooks
    ### prebuild
        ### update_package.sh
```

Per ricevere la versione più recente del pacchetto docker, questo script specifica il pacchetto docker nel comando yum update. Se il nome del pacchetto viene omesso, vengono applicati tutti gli aggiornamenti disponibili. L'esempio precedente del file di configurazione ne è la dimostrazione.

```
### update_package.sh ###

#!/bin/bash

/opt/elasticbeanstalk/bin/pkg-repo unlock
yum update docker -y
/opt/elasticbeanstalk/bin/pkg-repo lock
yum clean all -y
rm -rf /var/cache/yum
```

download-source-bundle (solo AMI Amazon Linux)

Per i rami della piattaforma AMI Amazon Linux (precedenti ad Amazon Linux 2), Elastic Beanstalk fornisce uno strumento aggiuntivo, `download-source-bundle`. Usalo per scaricare il codice sorgente dell'applicazione durante la distribuzione della piattaforma. Questo strumento è disponibile in `/opt/elasticbeanstalk/bin/download-source-bundle`.

Lo script di esempio `00-unzip.sh` si trova nella cartella `appdeploy/pre` nelle istanze di ambiente. Illustra come usare `download-source-bundle` per scaricare il codice sorgente dell'applicazione nella cartella `/opt/elasticbeanstalk/deploy/appsource` durante la distribuzione.

Estensione delle piattaforme Elastic Beanstalk Linux

Questo argomento descrive come estendere le piattaforme Linux con comandi, script, software e configurazioni personalizzati. Potrebbe essere necessario estendere la piattaforma per modificare il server proxy e la configurazione predefiniti. In alternativa, potrebbe essere necessario personalizzare il modo in cui la piattaforma crea o avvia l'applicazione.

Argomenti

- Buildfile e Procfile
- Hook della piattaforma
- File di configurazione
- Configurazione del proxy inverso
- Esempio di applicazione con estensioni

Buildfile e Procfile

Alcune piattaforme consentono di personalizzare la modalità di compilazione o preparazione dell'applicazione e di specificare i processi che eseguono l'applicazione. Ogni singolo argomento della piattaforma menziona specificamente Buildfile and/or Profile se la piattaforma li supporta. Cerca la tua piattaforma specifica sotto Piattaforme.

Per tutte le piattaforme di supporto, la sintassi e la semantica sono identiche e sono come descritto in questa pagina. I singoli argomenti della piattaforma menzionano l'uso specifico di questi file per la creazione e l'esecuzione di applicazioni nelle rispettive lingue.

Buildfile

Per specificare un comando di configurazione e compilazione personalizzato per l'applicazione, inserire un file denominato `Buildfile` nella directory principale dell'origine dell'applicazione. Per il nome del file è prevista la distinzione tra maiuscole e minuscole. Utilizzare la seguente sintassi per il tuo `Buildfile`.

```
<process_name>: <command>
```

Il comando nel `Buildfile` deve corrispondere alla seguente espressione regolare: `^[A-Za-z0-9_-]+:\s*[^\s].*$`.

Elastic Beanstalk non monitora l'applicazione eseguita con un `Buildfile`. Usa un `Buildfile` per comandi che vengono eseguiti per brevi periodi e terminano dopo aver completato le proprie attività. Per i processi di applicazioni a lungo termine da non terminare, utilizza il Procfile.

Tutti i percorsi nel `Buildfile` sono relativi alla radice del bundle di origine. Nel seguente esempio di `Buildfile`, `build.sh` è uno script di shell che si trova nella radice del bundle di origine:

Example Buildfile

```
make: ./build.sh
```

Se si desidera fornire passaggi di compilazione personalizzati, si consiglia di utilizzare gli hook della piattaforma `predeploy` per qualsiasi cosa tranne i comandi più semplici, invece di `Buildfile`. Gli hook della piattaforma consentono script più ricchi e una migliore gestione degli errori. Gli hook della piattaforma sono descritti nella sezione successiva.

Procfile

Per specificare comandi personalizzati per avviare ed eseguire l'applicazione, inserire un file denominato `Procfile` nella directory principale dell'origine dell'applicazione. Per il nome del file è prevista la distinzione tra maiuscole e minuscole. Utilizzare la seguente sintassi per il tuo `Procfile`. È possibile specificare uno o più comandi.

```
<process_name1>: <command1>
<process_name2>: <command2>
...
```

Ogni riga nel `Procfile` deve corrispondere alla seguente espressione regolare: `^[A-Za-z0-9_-]+:\s*[^\s].*$`

Utilizza un `Procfile` per i processi di applicazioni a lungo termine che non devono essere terminati. Elastic Beanstalk si aspetta che i processi del `Procfile` vengano eseguiti continuamente. Elastic Beanstalk monitora questi processi e riavvia qualsiasi processo che termina. Per i processi con esecuzione breve, utilizza un comando Buildfile.

Tutti i percorsi nel `Procfile` sono relativi alla radice del bundle di origine. Nell'esempio seguente `Procfile` definisce tre processi. Il primo, chiamato `web` nell'esempio, è l'applicazione web principale.

Example Procfile

```
web: bin/myserver
cache: bin/mycache
foo: bin/fooapp
```

Elastic Beanstalk configura il server proxy per inoltrare le richieste all'applicazione Web principale sulla porta 5000 ed è possibile configurare questo numero di porta. Un uso comune per un `Procfile` è quello di passare questo numero di porta all'applicazione come argomento di comando. Per i dettagli sulla configurazione del proxy, vedere. Configurazione del proxy inverso

Elastic Beanstalk acquisisce flussi di output e errori standard dai processi `Procfile` nei file di log. Elastic Beanstalk assegna i nomi dei file di log dopo il processo e li archivia in `/var/log`. Ad esempio, il processo `web` dell'esempio precedente genera log chiamati `web-1.log` e `web-1.error.log` per, rispettivamente, `stdout` e `stderr`.

Hook della piattaforma

Gli hook di piattaforma sono specificamente progettati per estendere la piattaforma del tuo ambiente. Si tratta di script personalizzati o di file eseguibili che vengono distribuiti come parte del codice sorgente dell'applicazione e che Elastic Beanstalk esegue durante varie fasi di provisioning delle istanze.

> **Note**
>
> Gli hook della piattaforma non sono supportati sulle versioni della piattaforma AMI Amazon Linux (precedenti ad Amazon Linux 2).

Hook della piattaforma di distribuzione delle applicazioni

Una distribuzione dell'applicazione si verifica quando si fornisce un nuovo bundle di origine per la distribuzione o quando si apportano modifiche alla configurazione che richiedono la terminazione e la nuova creazione di tutte le istanze dell'ambiente.

Per fornire hook di piattaforma eseguiti durante la distribuzione di un'applicazione, posizionare i file nella directory `.platform/hooks` del bundle di origine, in una delle seguenti sottodirectory.

- `prebuild`: i file vengono eseguiti dopo che il motore della piattaforma Elastic Beanstalk scarica ed estrae il bundle di origine dell'applicazione e prima che configuri l'applicazione e il server Web.

  I file `prebuild` vengono eseguiti dopo l'esecuzione di comandi trovati nella sezione comandi di qualsiasi file di configurazione e prima di eseguire i comandi `Buildfile`.
- `predeploy`: i file vengono eseguiti dopo che il motore della piattaforma Elastic Beanstalk ha impostato e configurato l'applicazione e il server Web e prima che li distribuisca nel percorso di runtime finale.

  I file `predeploy` vengono eseguiti dopo l'esecuzione di comandi trovati nella sezione container_commands di qualsiasi file di configurazione e prima di eseguire i comandi `Procfile`.
- `postdeploy`: i file vengono eseguiti dopo che il motore della piattaforma Elastic Beanstalk distribuisce l'applicazione e il server proxy.

  Questo è l'ultimo passaggio del flusso di lavoro di distribuzione.

Hook della piattaforma di distribuzione della configurazione

Una distribuzione di configurazione si verifica quando si apportano modifiche alla configurazione che aggiornano solo le istanze dell'ambiente senza ricrearle. I seguenti aggiornamenti delle opzioni causano un aggiornamento della configurazione.

- Proprietà dell'ambiente e impostazioni specifiche della piattaforma
- File statici
- AWS X-Ray demone
- Storage e streaming dei log
- Porta dell'applicazione (per i dettagli vedere) Configurazione del proxy inverso

Per fornire hook che vengono eseguiti durante una distribuzione di configurazione, posizionarli nella directory `.platform/confighooks` nel bundle di origine. Si applicano le stesse tre sottodirectory degli hook di distribuzione delle applicazioni.

Ulteriori informazioni sugli hook della piattaforma

I file hook possono essere file binari o file di script che iniziano con una riga `#!` contenente il percorso dell'interprete, ad esempio `#!/bin/bash`. Tutti i file devono disporre del permesso di esecuzione. Utilizzare `chmod +x` per impostare l'autorizzazione di esecuzione sui file hook. Per tutte le versioni della piattaforma basate su Amazon Linux 2023 e Amazon Linux 2 rilasciate a partire dal 29 aprile 2022, Elastic Beanstalk concede automaticamente le autorizzazioni di esecuzione a tutti gli script di hook della piattaforma. In questo caso non è necessario concedere manualmente le autorizzazioni di esecuzione. Per un elenco di queste versioni della piattaforma, fai riferimento alle note di rilascio di Linux del 29 aprile 2022 nella guida Note di rilascio di AWS Elastic Beanstalk .

Elastic Beanstalk esegue i file in ciascuna di queste directory in ordine lessicografico dei nomi dei file. Tutti i file vengono eseguiti come utente `root`. La directory di lavoro corrente (cwd) per gli hook della piattaforma è la directory principale dell'applicazione. Per i file `prebuild` e `predeploy` è la directory di staging dell'applicazione e per i file `postdeploy` è la directory dell'applicazione corrente. Se uno dei file non ha esito positivo (esce con un codice di uscita diverso da zero), la distribuzione si interrompe e fallisce.

Uno script di testo degli hook di una piattaforma può fallire se contiene caratteri di interruzione di riga Carriage Return/ Line Feed (CRLF) di Windows. Se un file è stato salvato in un host Windows e poi trasferito su un server Linux, può contenere interruzioni di riga CRLF di Windows. Per le piattaforme rilasciate a partire dal 29 dicembre 2022, Elastic Beanstalk converte automaticamente i caratteri CRLF di Windows in caratteri di interruzione di riga Line Feed (LF) di Linux nei file di testo degli hook della piattaforma. Se l'applicazione viene eseguita su qualsiasi piattaforma Amazon Linux 2 rilasciata prima di questa data, dovrai convertire i caratteri CRLF di Windows in caratteri LF di Linux. Un modo per farlo è creare e salvare il file di script su un host Linux. Gli strumenti che convertono questi caratteri sono disponibili anche su Internet.

I file hook hanno accesso a tutte le proprietà di ambiente definite nelle opzioni dell'applicazione e alle variabili di ambiente di sistema `HOME`, `PATH` e `PORT`.

Per ottenere valori di variabili di ambiente e altre opzioni di configurazione negli script di hook della piattaforma puoi utilizzare la utility `get-config` fornita da Elastic Beanstalk sulle istanze di ambiente. Per informazioni dettagliate, vedi the section called “Strumenti di script della piattaforma”.

File di configurazione

Puoi aggiungere i file di configurazione alla directory `.ebextensions` del codice sorgente dell'applicazione per configurare vari aspetti dell'ambiente Elastic Beanstalk. Tra le altre cose, i file di configurazione consentono di personalizzare software e altri file nelle istanze dell'ambiente ed eseguire comandi di inizializzazione sulle istanze. Per ulteriori informazioni, consulta the section called “Server Linux”.

È inoltre possibile impostare le opzioni di configurazione utilizzando i file di configurazione. Molte delle opzioni controllano il comportamento della piattaforma e alcune di queste opzioni sono specifiche della piattaforma.

Per le piattaforme basate su Amazon Linux 2 e Amazon Linux 2023, ti consigliamo di utilizzare Buildfile, Procfile e hook della piattaforma per configurare ed eseguire codice personalizzato sulle istanze dell'ambiente durante il provisioning delle istanze. Questi meccanismi sono descritti nelle sezioni precedenti di questa pagina. È comunque possibile utilizzare comandi e comandi container nei file di configurazione `.ebextensions`, ma non sono così facili da usare. Ad esempio, scrivere script di comando all'interno di un file YAML può essere difficile dal punto di vista della sintassi. È comunque necessario utilizzare i file di `.ebextensions` configurazione per qualsiasi script che richieda un riferimento a una AWS CloudFormation risorsa.

Configurazione del proxy inverso

Tutte le versioni della piattaforma Amazon Linux 2 e Amazon Linux 2023 usano nginx come server proxy inverso predefinito. Le piattaforme Tomcat, Node.js, PHP e Python supportano anche HTTPD di Apache come alternativa. Per selezionare Apache su queste piattaforme, impostare l'opzione `ProxyServer` nello spazio dei nomi `aws:elasticbeanstalk:environment:proxy` su `apache`. Tutte le piattaforme abilitano la configurazione del server proxy in modo uniforme, come descritto in questa sezione.

> **Note**
>
> Sulle versioni della piattaforma AMI Amazon Linux (precedenti ad Amazon Linux 2) potrebbe essere necessario configurare i server proxy in modo diverso. Puoi trovare questi dettagli legacy sotto i rispettivi argomenti della piattaforma in questa guida.

Elastic Beanstalk configura il server proxy nelle istanze dell'ambiente per inoltrare il traffico Web all'applicazione Web principale sull'URL root dell'ambiente, ad esempio `http://my-env.elasticbeanstalk.com`.

Per impostazione predefinita, Elastic Beanstalk configura il proxy per inoltrare le richieste in arrivo sulla porta 80 all'applicazione Web principale sulla porta 5000. È possibile configurare questo numero di porta impostando la proprietà ambiente `PORT` utilizzando lo psazio dei nomi aws:elasticbeanstalk:application:environment in un file di configurazione, come illustrato nell'esempio seguente.

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  PORT
    value:  <main_port_number>
```

Per ulteriori informazioni sull'impostazione delle variabili di ambiente per la tua applicazione, vedi the section called “Impostazioni opzioni”.

L'applicazione deve essere in ascolto sulla porta configurata per essa nel proxy. Se si modifica la porta predefinita utilizzando la proprietà dell'ambiente `PORT`, il codice può accedervi leggendo il valore della variabile di ambiente `PORT`. Ad esempio, si chiama `os.Getenv("PORT")` in Go o `System.getenv("PORT")` in Java. Se si configura il proxy per inviare traffico a più processi dell'applicazione, è possibile configurare diverse proprietà dell'ambiente e utilizzare i relativi valori sia nella configurazione del proxy che nel codice dell'applicazione. Un'altra opzione è quella di passare il valore della porta al processo come argomento di comando nel `Procfile`. Per ulteriori informazioni, consulta Buildfile e Procfile.

Configurazione di nginx

Elastic Beanstalk usa nginx come proxy inverso predefinito per mappare l'applicazione al sistema di bilanciamento del carico Elastic Load Balancing. Elastic Beanstalk fornisce una configurazione nginx predefinita che puoi estendere o sostituire completamente a favore di quella personalizzata.

> **Note**
>
> Quando aggiungi o modifichi un file di configurazione nginx `.conf`, assicurati di codificarlo come UTF-8.

Per estendere la configurazione nginx predefinita di Elastic Beanstalk, aggiungi i file di configurazione `.confa` una cartella denominata `.platform/nginx/conf.d/` nel bundle di origine dell'applicazione. La configurazione nginx di Elastic Beanstalk include automaticamente i file `.conf` in questa cartella.

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- conf.d
|           `-- myconf.conf
`-- other source files
```

I file di configurazione in `.platform/nginx/conf.d/` sono inclusi nel `http` blocco della configurazione di nginx. Usa questa posizione per le configurazioni che si applicano a livello globale.

Per estendere la configurazione predefinita del blocco `server` nginx, `.conf` aggiungi i file di configurazione a una cartella `.platform/nginx/conf.d/elasticbeanstalk/` denominata nel pacchetto sorgente dell'applicazione. La configurazione di Elastic Beanstalk nginx include i file in questa cartella all'interno del blocco`.conf`. `server`

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- conf.d
|           `-- elasticbeanstalk
|               `-- server.conf
`-- other source files
```

Utilizza questa posizione per aggiungere configurazioni specifiche del server, come blocchi di posizione aggiuntivi, pagine di errore personalizzate o direttive a livello di server. L'esempio seguente aggiunge un blocco di posizione personalizzato.

Example. platform/nginx/conf.d/elasticbeanstalk/server.conf

```
location /test {
    return 200 "Hello World!";
    add_header Content-Type text/plain;
}
```

Per sostituire completamente la configurazione nginx predefinita di Elastic Beanstalk, includi una configurazione nel bundle di origine in `.platform/nginx/nginx.conf`:

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- nginx.conf
`-- other source files
```

Se sostituisci la configurazione nginx predefinita di Elastic Beanstalk, aggiungi al file `nginx.conf` la riga seguente per inserire le configurazioni di Elastic Beanstalk per Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk, le mappature automatiche delle applicazioni e i file statici.

```
 include conf.d/elasticbeanstalk/*.conf;
```

Configurare HTTPD di Apache

Le piattaforme Tomcat, Node.js, PHP e Python consentono di scegliere il server proxy HTTDP di Apache come alternativa a nginx. Questo non è l'impostazione predefinita. L'esempio seguente configura Elastic Beanstalk per l'uso di Apache HTTPD.

Example.ebextensions/httpd-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

Puoi estendere la configurazione Apache predefinita di Elastic Beanstalk con file di configurazione personalizzati aggiuntivi. In alternativa, puoi sostituire completamente la configurazione Apache predefinita di Elastic Beanstalk.

Per estendere la configurazione Apache predefinita di Elastic Beanstalk, aggiungi i file di configurazione `.conf` a una cartella denominata `.platform/httpd/conf.d` nel bundle di origine dell'applicazione. La configurazione Apache di Elastic Beanstalk include automaticamente i file `.conf` in questa cartella.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
```

```
|-- .platform
|   -- httpd
|      -- conf.d
|         -- port5000.conf
|         -- ssl.conf
-- index.jsp
```

Ad esempio, la configurazione Apache 2.4 seguente aggiunge un listener sulla porta 5000.

Example. platform/httpd/conf.d/port5000. conf

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Per sostituire completamente la configurazione Apache predefinita di Elastic Beanstalk, includi una configurazione nel bundle di origine in `.platform/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
|-- .platform
|   `-- httpd
|       `-- conf
|           `-- httpd.conf
`-- index.jsp
```

> **Note**
>
> Se sostituisci la configurazione Apache predefinita di Elastic Beanstalk, aggiungi al file `httpd.conf` la riga seguente per inserire le configurazioni di Elastic Beanstalk per Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk, le mappature automatiche delle applicazioni e i file statici.

```
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Esempio di applicazione con estensioni

Nell'esempio seguente viene illustrato un bundle di origine applicazione con diverse caratteristiche di estensibilità supportate dalle piattaforme Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023: `Procfile`, file di configurazione `.ebextensions`, hook personalizzati e file di configurazione proxy.

```
~/my-app/
|-- web.jar
|-- Procfile
|-- readme.md
|-- .ebextensions/
|   |-- options.config        # Option settings
|   `-- cloudwatch.config     # Other .ebextensions sections, for example files and
 container commands
`-- .platform/
    |-- nginx/                # Proxy configuration
    |   |-- nginx.conf
    |   `-- conf.d/
    |       |-- custom.conf
    |       `-- elasticbeanstalk/
    |           `-- server.conf
    |-- hooks/                # Application deployment hooks
    |   |-- prebuild/
    |   |   |-- 01_set_secrets.sh
    |   |   `-- 12_update_permissions.sh
    |   |-- predeploy/
    |   |   `-- 01_some_service_stop.sh
    |   `-- postdeploy/
    |       |-- 01_set_tmp_file_permissions.sh
    |       |-- 50_run_something_after_app_deployment.sh
    |       `-- 99_some_service_start.sh
    `-- confighooks/          # Configuration deployment hooks
        |-- prebuild/
        |   `-- 01_set_secrets.sh
        |-- predeploy/
        |   `-- 01_some_service_stop.sh
        `-- postdeploy/
            |-- 01_run_something_after_config_deployment.sh
```

```
            `-- 99_some_service_start.sh
```

> **Note**
>
> Alcune di queste estensioni non sono supportate sulle versioni della piattaforma AMI Amazon Linux (precedenti ad Amazon Linux 2).

Distribuzione di applicazioni.NET Windows con Elastic Beanstalk

> **Dai un'occhiata a .NET on Developer Center AWS**
>
> Ti sei fermato al nostro.Net Developer Center? È il nostro sportello unico per tutto ciò che riguarda .NET AWS.
> Per ulteriori informazioni, consulta .NET on AWS Developer Center.

Questo capitolo fornisce istruzioni per la configurazione e la distribuzione delle applicazioni Web ASP.NET e.NET Core per Windows su. AWS Elastic Beanstalk Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni Web.NET (Windows) utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, è possibile continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure è possibile utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Questo capitolo fornisce i seguenti tutorial:

- QuickStart per.NET Core su Windows— Step-by-step istruzioni per creare e distribuire un'applicazione Windows Hello World .NET Core utilizzando l'EB CLI.
- QuickStart per ASP.NET— Step-by-step istruzioni per creare e distribuire un'applicazione Hello World ASP.NET utilizzando. AWS Toolkit for Visual Studio

Se hai bisogno di assistenza per lo sviluppo di applicazioni Windows.NET Core, puoi rivolgerti a diversi siti:

- Forum di sviluppo.NET: pubblica le tue domande e ottieni feedback.
- .NET Developer Center: sportello unico per codice di esempio, documentazione, strumenti e risorse aggiuntive.
- AWS Documentazione SDK for .NET: leggi la configurazione dell'SDK e l'esecuzione di esempi di codice, le funzionalità dell'SDK e informazioni dettagliate sulle operazioni API per l'SDK.

> **Note**
>
> Questa piattaforma non supporta gli ambienti di lavoro. Per informazioni dettagliate, vedi Ambienti worker Elastic Beanstalk.

Argomenti

- QuickStart: distribuzione di un'applicazione.NET Core su Windows su Elastic Beanstalk
- QuickStart: distribuzione di un' ASP.NET applicazione su Elastic Beanstalk
- Configurazione dell'ambiente di sviluppo .NET
- Utilizzo della piattaforma Windows Elastic Beanstalk .NET
- Aggiunta di un'istanza database Amazon RDS all'ambiente applicativo .NET
- La AWS Toolkit for Visual Studio
- Migrazione dell'applicazione.NET in locale a Elastic Beanstalk
- Consigli per i componenti ritirati di Windows Server su Elastic Beanstalk

QuickStart: distribuzione di un'applicazione.NET Core su Windows su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione.NET Core per Windows e la sua distribuzione in un AWS Elastic Beanstalk ambiente.

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Passaggio 1: Creare un'applicazione.NET Core per Windows
- Passaggio 2: Esegui l'applicazione localmente

- Passaggio 3: distribuire l'applicazione.NET Core su Windows con l'EB CLI
- Passaggio 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti a un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt (>) e dal nome della directory corrente, se appropriato.

```
C:\eb-project> this is a command
this is output
```

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

.NET Core su Windows

Se non hai installato .NET SDK sul tuo computer locale, puoi installarlo seguendo il link Download.NET sul sito Web della documentazione .NET.

Verifica l'installazione di.NET SDK eseguendo il comando seguente.

```
C:\> dotnet --info
```

Passaggio 1: Creare un'applicazione.NET Core per Windows

Crea una directory di progetto.

```
C:\> mkdir eb-dotnetcore
C:\> cd eb-dotnetcore
```

Quindi, crea un esempio di applicazione di servizio web Hello World RESTful eseguendo i seguenti comandi.

```
C:\eb-dotnetcore> dotnet new web --name HelloElasticBeanstalk
C:\eb-dotnetcore> cd HelloElasticBeanstalk
```

Passaggio 2: Esegui l'applicazione localmente

Esegui il comando seguente per eseguire l'applicazione localmente.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet run
```

L'output dovrebbe assomigliare al testo seguente.

```
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7222
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5228
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
```

```
    Content root path: C:\Users\Administrator\eb-dotnetcore\HelloElasticBeanstalk
```

> **Note**
>
> Il `dotnet` comando seleziona una porta a caso quando si esegue l'applicazione localmente. In questo esempio la porta è 5228. Quando distribuisci l'applicazione nell'ambiente Elastic Beanstalk, l'applicazione verrà eseguita sulla porta 5000.

Inserisci l'indirizzo URL `http://localhost:port` nel tuo browser web. Per questo esempio specifico, il comando è`http://localhost:5228`. Il browser web dovrebbe visualizzare «Hello World!».

Passaggio 3: distribuire l'applicazione.NET Core su Windows con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire l'applicazione.NET Core su Windows

1. Esegui i seguenti comandi nella `HelloElasticBeanstalk` directory per pubblicare e comprimere l'applicazione.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet publish -o site
C:\eb-dotnetcore\HelloElasticBeasntalk> cd site
C:\eb-dotnetcore\HelloElasticBeasntalk\site> Compress-Archive -Path * -
DestinationPath ../site.zip
C:\eb-dotnetcore\HelloElasticBeasntalk\site> cd ..
```

2. Crea un nuovo file nel file `HelloElasticBeanstalk` chiamato `aws-windows-deployment-manifest.json` con il seguente contenuto:

```
{
    "manifestVersion": 1,
    "deployments": {
        "aspNetCoreWeb": [
        {
            "name": "test-dotnet-core",
            "parameters": {
                "appBundle": "site.zip",
                "iisPath": "/",
```

```
                "iisWebSite": "Default Web Site"
            }
        }
        ]
    }
}
```

3. Inizializza il repository della CLI EB con il comando eb init.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb init -p iis dotnet-windows-server-
tutorial --region us-east-2
```

   Questo comando crea un'applicazione denominata `dotnet-windows-server-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma server Windows.

4. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb create dotnet-windows-server-env
```

   Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Passaggio 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb open
```

Complimenti! Hai distribuito un'applicazione.NET Core su Windows con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 5: rimozione

Puoi chiudere il tuo ambiente quando finisci di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. `subdomain region`.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova

versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

> **Prova altri tutorial**
>
> Se desideri provare altri tutorial con diverse applicazioni di esempio, consulta. QuickStart per ASP.NET

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire localmente .NET Core su applicazioni Windows, vedi Configurazione dell'ambiente di sviluppo .NET

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

QuickStart: distribuzione di un' ASP.NET applicazione su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un' ASP.NET applicazione e di distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Fase 1: Creare un' ASP.NET applicazione

- Passaggio 2: Esegui l'applicazione localmente
- Passaggio 3: Implementa l' ASP.NET applicazione con AWS Toolkit for Visual Studio
- Passaggio 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti a un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Questo QuickStart tutorial illustra come creare un'applicazione «Hello World» e distribuirla in un ambiente Elastic Beanstalk con Visual Studio e il. AWS Toolkit for Visual Studio

Visual Studio

Per scaricare e installare Visual Studio, segui le istruzioni nella pagina di download di Visual Studio. Questo esempio utilizza Visual Studio 2022. Durante l'installazione di Visual Studio, seleziona questi elementi specifici:

- Nella scheda Carichi di lavoro, seleziona ASP.NET e sviluppa web.
- Nella scheda Componenti individuali, seleziona gli strumenti di sviluppo.NET Framework 4.8 e i modelli di progetti.NET Framework e di elementi.

AWS Toolkit for Visual Studio

Per scaricare e configurare, AWS Toolkit for Visual Studio segui le istruzioni nell'argomento Guida introduttiva della Guida per l' AWS Toolkit for Visual Studio utente.

Fase 1: Creare un' ASP.NET applicazione

Successivamente, crea un'applicazione da distribuire in un ambiente Elastic Beanstalk. Creeremo un'applicazione web «Hello World». ASP.NET

Per creare un' ASP.NET applicazione

1. Avvia Visual Studio. Nel menu File, seleziona Nuovo, quindi Progetto.
2. Viene visualizzata la finestra di dialogo Crea un nuovo progetto. Seleziona l'applicazione ASP.NET web (.NET Framework), quindi seleziona Avanti.
3. Nella finestra di dialogo Configura il nuovo progetto, inserisci `eb-aspnet` il nome del progetto. Dal menu a discesa Framework seleziona.NET Framework 4.8, quindi seleziona Crea.

   Nota la directory del progetto. In questo esempio, la directory del progetto è`C:\Users\Administrator\source\repos\eb-aspnet\eb-aspnet`.
4. Viene visualizzata la finestra di dialogo Crea una nuova applicazione ASP.NET Web. Seleziona il modello vuoto. Quindi seleziona Crea.

   A questo punto, hai creato un progetto di applicazione ASP.NET web vuoto utilizzando Visual Studio. Successivamente, creeremo un modulo Web che fungerà da punto di ingresso per l'applicazione ASP.NET Web.
5. Dal menu Progetto, seleziona Aggiungi nuovo elemento. Nella pagina Aggiungi nuovo elemento, seleziona Modulo Web e assegnagli un nome`Default.aspx`. Quindi seleziona Aggiungi.
6. Aggiungi quanto segue a `Default.aspx:`

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
 Inherits="eb_aspnet.Default" %>

<!DOCTYPE html>

<html xmlns="https://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Hello Elastic Beanstalk!</title>
</head>
```

```
<body>
    <form id="body" runat="server">
        <div>
            Hello Elastic Beanstalk! This is an ASP.NET on Windows Server
 application.
        </div>
    </form>
</body>
</html>
```

Passaggio 2: Esegui l'applicazione localmente

In Visual Studio, dal menu Debug seleziona Avvia debug per eseguire l'applicazione localmente. La pagina dovrebbe mostrare «Hello Elastic Beanstalk! Questa è un'applicazione ASP.NET su Windows Server».

Passaggio 3: Implementa l' ASP.NET applicazione con AWS Toolkit for Visual Studio

Segui questi passaggi per creare un ambiente Elastic Beanstalk e distribuirvi la nuova applicazione.

Per creare un ambiente e distribuire l'applicazione ASP.NET

1. In Solution Explorer, fai clic con il pulsante destro del mouse sull'applicazione, quindi seleziona Pubblica su. AWS Elastic Beanstalk
2. Scegli un nome per la tua nuova applicazione e ambiente Elastic Beanstalk.
3. Oltre questo punto, puoi procedere con le impostazioni predefinite fornite da Elastic Beanstalk o modificare le opzioni e le impostazioni a tuo piacimento.
4. Nella pagina di revisione, seleziona Distribuisci. Questo comprimerà la tua applicazione ASP.NET web e la distribuirà su Elastic Beanstalk.

   Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente. La funzione di distribuzione di Elastic Beanstalk monitorerà il tuo ambiente esistente finché non sarà disponibile con il codice appena distribuito. Nella scheda Env: **< `environment name` >**, vedrai lo stato del tuo ambiente.

Passaggio 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, la scheda Env: **< `environment name` >** mostra le informazioni sull'ambiente e sull'applicazione, incluso l'URL del dominio per avviare l'applicazione. Seleziona questo URL in questa scheda o copialo e incollalo nel tuo browser web.

Complimenti! Hai distribuito un' ASP.NET applicazione con Elastic Beanstalk!

Fase 5: rimozione

Quando hai finito di lavorare con l'applicazione, puoi chiudere il tuo ambiente in. AWS Toolkit for Visual Studio

Per terminare l'ambiente

1. Espandi il nodo Elastic Beanstalk e il nodo dell'applicazione in Explorer.AWS Right-click il tuo ambiente applicativo e seleziona Terminate Environment.
2. Quando richiesto, seleziona Yes (Sì) per confermare che intendi terminare l'ambiente. Elastic Beanstalk impiegherà alcuni minuti per AWS terminare le risorse in esecuzione nell'ambiente.

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.

- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

> **Prova altri tutorial**
>
> Se desideri provare altri tutorial con diverse applicazioni di esempio, consulta. QuickStart per.NET Core su Windows

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire ASP.NET applicazioni localmente, vedi Configurazione dell'ambiente di sviluppo .NET

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Configurazione dell'ambiente di sviluppo .NET

Questo argomento fornisce istruzioni per configurare un ambiente di sviluppo.NET Windows per testare l'applicazione localmente prima di distribuirla su. AWS Elastic Beanstalk Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Sections

- Installazione di un IDE
- Installazione di AWS Toolkit for Visual Studio

Se devi gestire AWS le risorse dall'interno dell'applicazione, installa l' AWS SDK for .NET. Ad esempio, è possibile usare Amazon S3 per archiviare e recuperare i dati.

Con AWS SDK for .NET, puoi iniziare in pochi minuti con un unico pacchetto scaricabile completo di modelli di progetto di Visual Studio, libreria.NET, AWS esempi di codice C# e documentazione. In C# vengono forniti esempi pratici su come utilizzare le librerie per creare applicazioni. Tutorial video e documentazione di riferimento vengono forniti online per aiutarti a scoprire come utilizzare le librerie e gli esempi di codice.

Visita la homepage di SDK AWS per .NET per ulteriori informazioni e istruzioni di installazione.

Installazione di un IDE

Gli ambienti di sviluppo integrati (IDEs) offrono un'ampia gamma di funzionalità che facilitano lo sviluppo di applicazioni. Se non hai utilizzato un IDE per lo sviluppo .NET, prova Visual Studio Community per iniziare.

Visita la pagina di Visual Studio Community per scaricare e installare Visual Studio Community.

Installazione di AWS Toolkit for Visual Studio

AWS Toolkit for Visual StudioSi tratta di un plug-in open source per l'IDE di Visual Studio che semplifica lo sviluppo, il debug e la distribuzione di applicazioni.NET utilizzando. AWS Visita la home page di Toolkit for Visual Studio per le istruzioni di installazione.

Utilizzo della piattaforma Windows Elastic Beanstalk .NET

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Web ASP.NET e.NET Core Windows su Elastic Beanstalk.

AWS Elastic Beanstalk supporta diverse piattaforme per diverse versioni del framework di programmazione .NET e di Windows Server. Consulta la sezione relativa a .NET su Windows Server con IIS nel documento sulle piattaforme AWS Elastic Beanstalk per un elenco completo.

Elastic Beanstalk offre opzioni di configurazione che puoi utilizzare per personalizzare il software che viene eseguito EC2 sulle istanze nell'ambiente Elastic Beanstalk. Puoi configurare le variabili di ambiente necessarie all'applicazione, attivare la rotazione dei log in Amazon S3 e configurare le impostazioni del framework .NET.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Configurazione dell'ambiente.NET nella console Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3, configurare variabili che l'applicazione è in grado di leggere dall'ambiente e modificare le impostazioni del framework .NET.

Per configurare l'ambiente.NET nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni del contenitore

- Target .NET runtime (Runtime target .NET): impostare questa opzione su `2.0` per eseguire CLR v2.
- Enable 32-bit applications (Abilita le applicazioni a 32 bit): impostare questa opzione su `True` per eseguire applicazioni a 32 bit.

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Abilita la rotazione dei file di registro su Amazon S3: specifica se i file di log per le EC2 istanze Amazon dell'applicazione vengono copiati nel bucket Amazon S3 associato all'applicazione.

Proprietà dell'ambiente

La sezione Environment Properties consente di specificare le impostazioni di configurazione dell'ambiente sulle EC2 istanze Amazon su cui è in esecuzione l'applicazione. Queste impostazioni sono passate come coppie chiave-valore all'applicazione. Utilizza `System.GetEnvironmentVariable` per leggerle. In `web.config` e nelle proprietà dell'ambiente possono esservi chiavi identiche. Utilizza lo spazio dei nomi `System.Configuration` per leggere i valori da `web.config`.

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazio dei nomi aws:elasticbeanstalk:container:dotnet:apppool

Puoi utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

La piattaforma .NET definisce le opzioni nel namespace `aws:elasticbeanstalk:container:dotnet:apppool` che puoi utilizzare per configurare il runtime .NET.

Il seguente file di configurazione di esempio mostra le impostazioni per ognuna delle opzioni disponibili in questo namespace:

Example.ebextensions/dotnet-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:dotnet:apppool:
    Target Runtime: 2.0
    Enable 32-bit Applications: True
```

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Migrazione tra le versioni principali della piattaforma server Windows di Elastic Beanstalk

AWS Elastic Beanstalk ha avuto diverse versioni principali della sua piattaforma Windows Server. Questa pagina descrive i miglioramenti più significativi per ogni versione principale e cosa occorre considerare prima di eseguire la migrazione a una versione più recente.

La versione corrente della piattaforma Windows Server è la versione 2 (v2). Se l'applicazione usa qualsiasi versione della piattaforma Windows Server precedente a v2, ti consigliamo di eseguire la migrazione a v2.

Novità delle versioni principali della piattaforma Windows Server

Piattaforma Windows Server V2

La versione 2 (v2) della piattaforma Windows Server di Elastic Beanstalk è stata resa disponibile a febbraio 2019. La versione V2 avvicina il funzionamento della piattaforma Windows Server a quello delle piattaforme basate su Linux di Elastic Beanstalk in molti modi importanti. La versione V2 è completamente compatibile con la versione v1 e questo semplifica la migrazione dalla versione v1.

La piattaforma Windows Server ora supporta quanto segue:

- Funzione Versioni multiple: ogni versione ottiene un nuovo numero di versione e puoi fare riferimento alle versioni precedenti, ancora disponibili, quando crei e gestisci gli ambienti.
- Integrità migliorata: per informazioni dettagliate, consulta Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk.
- Distribuzioni Non modificabile e In sequenza con un batch aggiuntivo – Per informazioni dettagliate sulle policy di distribuzione, consulta Distribuzione di applicazioni in ambienti Elastic Beanstalk.
- Aggiornamenti non modificabili: per informazioni dettagliate sui tipi di aggiornamento, consulta Modifiche di configurazione.
- Aggiornamenti gestiti della piattaforma: per informazioni dettagliate, consulta Aggiornamenti gestiti della piattaforma.

> **Note**
>
> Le nuove caratteristiche di distribuzione e aggiornamento dipendono dallo stato avanzato. Abilita stato avanzato per utilizzarle. Per informazioni dettagliate, consulta Abilitazione del reporting dello stato avanzato Elastic Beanstalk.

Piattaforma Windows Server V1

La versione 1.0.0 (v1) della piattaforma Windows Server di Elastic Beanstalk è stata resa disponibile a ottobre 2015. Questa versione cambia l'ordine in cui Elastic Beanstalk elabora i comandi nei file di configurazione durante la creazione dell'ambiente e gli aggiornamenti.

Le versioni precedenti della piattaforma non dispongono di un numero di versione nel nome dello stack della soluzione:

- Windows Server 2012 R2 (64 bit) con IIS 8.5 in esecuzione
- Windows Server Core 2012 R2 (64 bit) con IIS 8.5 in esecuzione
- Windows Server 2012 (64 bit) con IIS 8 in esecuzione
- Windows Server 2008 R2 (64 bit) con IIS 7.5 in esecuzione

Nelle versioni precedenti l'ordine di elaborazione per i file di configurazione è incoerente. Durante la creazione dell'ambiente, `Container Commands` viene eseguito dopo che il codice sorgente dell'applicazione viene distribuito su IIS. Durante una distribuzione in un ambiente di esecuzione, i

comandi del container vengono eseguiti prima della distribuzione della nuova versione. Durante un aumento del dimensionamento, i file di configurazione non vengono elaborati.

Oltre a questo, IIS viene avviato prima dell'esecuzione dei comandi del container. Questo comportamento ha portato alcuni clienti a implementare soluzioni alternative nei comandi dei container, con una pausa del server IIS prima dell'esecuzione dei comandi e con un avvio successivo al completamento.

La versione 1 corregge l'incoerenza e avvicina il comportamento della piattaforma Windows Server a quello delle piattaforme basate su Linux di Elastic Beanstalk. Nella piattaforma v1, Elastic Beanstalk esegue sempre i comandi dei container prima di avviare il server IIS.

La soluzione della piattaforma v1 specifica v1 dopo la versione di Windows Server:

- Windows Server 2012 R2 (64 bit) v1.1.0 con IIS 8.5 in esecuzione
- Windows Server Core 2012 R2 (64 bit) v1.1.0 con IIS 8.5 in esecuzione
- Windows Server 2012 (64 bit) v1.1.0 con IIS 8 in esecuzione
- Windows Server 2008 R2 (64 bit) v1.1.0 con IIS 7.5 in esecuzione

Inoltre, la piattaforma v1 estrae il contenuto del bundle di origine dell'applicazione in `C:\staging\` prima dell'esecuzione dei comandi dei container. Quando i comandi container hanno completato l'operazione, i contenuti di questa cartella vengono compressi in un file .zip e distribuiti in IIS. Questo flusso di lavoro consente di modificare i contenuti del tuo bundle di origine dell'applicazione con comandi o con uno script prima della distribuzione.

Migrazione da versioni principali della piattaforma Windows Server

Questa sezione contiene alcune considerazioni sulla migrazione di cui tenere conto prima di aggiornare l'ambiente. Per aggiornare la piattaforma dell'ambiente a una versione più recente, consulta Aggiornamento della versione della piattaforma dell'ambiente Elastic Beanstalk.

Da V1 verso V2

La piattaforma Windows Server v2 non supporta .NET Core 1.x e 2.0. Se esegui la migrazione dell'applicazione da Windows Server v1 a v2 e l'applicazione utilizza una di queste versioni .NET Core, aggiorna l'applicazione a una versione .NET Core supportata da v2. Per un elenco delle versioni supportate, consulta .NET su Windows Server con IIS in Piattaforme AWS Elastic Beanstalk .

Se l'applicazione utilizza una Amazon Machine Image (AMI) personalizzata, crea una nuova AMI personalizzata in base all'AMI della piattaforma di Windows Server v2. Per ulteriori informazioni, vedi Utilizzo di un'immagine Amazon Machine Image (AMI) personalizzata nell'ambiente Elastic Beanstalk.

> **Note**
>
> Le caratteristiche di distribuzione e aggiornamento che sono nuove per Windows Server v2 dipendono dallo stato avanzato. Quando si esegue la migrazione di un ambiente a v2, lo stato avanzato è disabilitato. Abilitarlo per utilizzare queste caratteristiche. Per informazioni dettagliate, consulta Abilitazione del reporting dello stato avanzato Elastic Beanstalk.

Da versioni precedenti alla versione V1

Oltre alle considerazioni sulla migrazione da v1, se esegui la migrazione dell'applicazione da uno stack di soluzioni Windows Server che è precedente alla versione v1 e usi attualmente comandi dei container, rimuovi tutti i comandi aggiunti per risolvere le incoerenze di elaborazione quando esegui la migrazione a una versione più recente. A partire da v1, i comandi del container vengono sicuramente eseguiti completamente prima dell'origine dell'applicazione che viene distribuita e prima dell'avvio di IIS. Questo consente di apportare eventuali modifiche all'origine in `C:\staging` e modificare i file di configurazione IIS durante questa fase senza problemi.

Ad esempio, puoi utilizzare per AWS CLI scaricare un file DLL nella sorgente dell'applicazione da Amazon S3:

`.ebextensions\copy-dll.config`

```
container_commands:
  copy-dll:
    command: aws s3 cp s3://amzn-s3-demo-bucket/dlls/large-dll.dll .\lib\
```

Per ulteriori informazioni sull'utilizzo dei file di configurazione, consulta Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Esecuzione di più applicazioni e delle applicazioni ASP.NET Core con un manifest di distribuzione

È possibile utilizzare un manifest di distribuzione per indicare a Elastic Beanstalk come distribuire l'applicazione. Utilizzando questo metodo, non è necessario utilizzare `MSDeploy` per generare un

bundle di origine per una singola applicazione ASP.NET che viene eseguita nel percorso root del sito Web. Piuttosto, è possibile utilizzare un file manifest per eseguire più applicazioni in percorsi diversi. In alternativa, puoi chiedere a Elastic Beanstalk di distribuire ed eseguire l'app con ASP.NET Core. È inoltre possibile utilizzare un manifest di distribuzione per configurare un pool di applicazioni in cui eseguire le applicazioni.

I manifest di distribuzione aggiungono il supporto per Applicazioni .NET Core su Elastic Beanstalk. È possibile distribuire un'applicazione .NET Framework senza un manifest di distribuzione. Tuttavia, le applicazioni .NET Core richiedono un manifest di distribuzione per essere eseguite su Elastic Beanstalk. Quando utilizzi un manifest di distribuzione, è necessario creare un archivio del sito per ogni applicazione e quindi raggruppare gli archivi del sito in un secondo archivio ZIP che contiene il manifest di distribuzione.

I manifest di distribuzione aggiungono anche la possibilità di eseguire più applicazioni in diversi percorsi. Un manifest di distribuzione definisce una gamma di obiettivi di distribuzione, ognuno con un archivio del sito e un percorso su cui IIS dovrebbe eseguirlo. Ad esempio, è possibile eseguire un'API Web nel percorso `/api` per servire richieste asincrone e un'app Web nel percorso root che utilizza l'API.

È possibile utilizzare un manifesto di distribuzione per configurare i siti Web IIS con collegamenti e percorsi fisici personalizzati. Ciò consente di configurare siti Web che ascoltano su porte o nomi host specifici prima di distribuire le applicazioni.

È inoltre possibile utilizzare un manifest di distribuzione per eseguire più applicazioni utilizzando pool di applicazioni in IIS o Kestrel. È possibile configurare un pool di applicazioni per riavviare le proprie applicazioni periodicamente, eseguire applicazioni a 32 bit o utilizzare una versione specifica del runtime di .NET Framework.

Per una personalizzazione completa, puoi scrivere i tuoi script di distribuzione in Windows PowerShell e indicare a Elastic Beanstalk quali script eseguire per installare, disinstallare e riavviare l'applicazione.

I manifest di distribuzione e le relative caratteristiche richiedono una piattaforma Windows Server versione 1.2.0 o più recente.

Per informazioni dettagliate su tutte le opzioni di configurazione, le proprietà e le funzionalità avanzate disponibili, come ignorare le reimpostazioni di IIS, consulta il riferimento allo schema del manifesto di distribuzione.

Sections

- App .NET Core
- Esecuzione di più applicazioni
- Configurazione dei siti Web IIS
- Utilizzo dell'Application Request Routing (ARR)
- Configurazione dei pool delle applicazioni
- Definizione delle distribuzioni personalizzate
- Riferimento allo schema del manifesto di distribuzione

App .NET Core

È possibile utilizzare un manifest di implementazione per eseguire applicazioni .NET Core su Elastic Beanstalk. .NET Core è una versione multipiattaforma di .NET fornita con uno strumento a riga di comando (`dotnet`). È possibile utilizzarlo per generare un'applicazione, eseguirla in locale e prepararla per la pubblicazione.

Per eseguire un'applicazione .NET Core su Elastic Beanstalk, esegui `dotnet publish` e raggruppa l'output in un archivio ZIP senza includere le directory. Posiziona l'archivio del sito in un bundle di origine con un manifest di distribuzione con una destinazione di distribuzione di tipo `aspNetCoreWeb`.

Il seguente manifest di distribuzione esegue un'applicazione .NET Core da un archivio del sito denominato `dotnet-core-app.zip` al percorso root.

Example aws-windows-deployment-manifest.json - .NET core

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
```

```
}
```

Raggruppa il manifest e l'archivio del sito in un archivio ZIP per creare un bundle di origine.

Example dotnet-core-bundle.zip

```
.
|-- aws-windows-deployment-manifest.json
`-- dotnet-core-app.zip
```

L'archivio del sito contiene il codice compilato dell'applicazione, le dipendenze e il file `web.config`.

Example dotnet-core-app.zip

```
.
|-- Microsoft.AspNetCore.Hosting.Abstractions.dll
|-- Microsoft.AspNetCore.Hosting.Server.Abstractions.dll
|-- Microsoft.AspNetCore.Hosting.dll
|-- Microsoft.AspNetCore.Http.Abstractions.dll
|-- Microsoft.AspNetCore.Http.Extensions.dll
|-- Microsoft.AspNetCore.Http.Features.dll
|-- Microsoft.AspNetCore.Http.dll
|-- Microsoft.AspNetCore.HttpOverrides.dll
|-- Microsoft.AspNetCore.Server.IISIntegration.dll
|-- Microsoft.AspNetCore.Server.Kestrel.dll
|-- Microsoft.AspNetCore.WebUtilities.dll
|-- Microsoft.Extensions.Configuration.Abstractions.dll
|-- Microsoft.Extensions.Configuration.EnvironmentVariables.dll
|-- Microsoft.Extensions.Configuration.dll
|-- Microsoft.Extensions.DependencyInjection.Abstractions.dll
|-- Microsoft.Extensions.DependencyInjection.dll
|-- Microsoft.Extensions.FileProviders.Abstractions.dll
|-- Microsoft.Extensions.FileProviders.Physical.dll
|-- Microsoft.Extensions.FileSystemGlobbing.dll
|-- Microsoft.Extensions.Logging.Abstractions.dll
|-- Microsoft.Extensions.Logging.dll
|-- Microsoft.Extensions.ObjectPool.dll
|-- Microsoft.Extensions.Options.dll
|-- Microsoft.Extensions.PlatformAbstractions.dll
|-- Microsoft.Extensions.Primitives.dll
|-- Microsoft.Net.Http.Headers.dll
|-- System.Diagnostics.Contracts.dll
|-- System.Net.WebSockets.dll
```

```
|-- System.Text.Encodings.Web.dll
|-- dotnet-core-app.deps.json
|-- dotnet-core-app.dll
|-- dotnet-core-app.pdb
|-- dotnet-core-app.runtimeconfig.json
`-- web.config
```

Esecuzione di più applicazioni

È possibile eseguire più applicazioni con un manifest di distribuzione definendo più target di distribuzione.

Il seguente manifesto di distribuzione configura due applicazioni .NET Core. L'`WebApiSampleApp`applicazione implementa una semplice API web e serve richieste asincrone sul percorso. `/api` `DotNetSampleApp` è un'applicazione Web che fornisce richieste al percorso root.

Example aws-windows-deployment-manifest.json: più app

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "WebAPISample",
        "parameters": {
          "appBundle": "WebApiSampleApp.zip",
          "iisPath": "/api"
        }
      },
      {
        "name": "DotNetSample",
        "parameters": {
          "appBundle": "DotNetSampleApp.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Un'applicazione di esempio con più applicazioni è disponibile qui:

- Pacchetto sorgente distribuibile - -v2.zip dotnet-multiapp-sample-bundle

- Codice sorgente: -v2.zip dotnet-multiapp-sample-source

Configurazione dei siti Web IIS

È possibile configurare i siti Web IIS con collegamenti e percorsi fisici personalizzati utilizzando il manifesto di distribuzione. Ciò è utile quando è necessario configurare siti Web che ascoltano su porte specifiche, utilizzano nomi host personalizzati o forniscono contenuti da directory specifiche.

Il seguente manifesto di distribuzione configura un sito Web IIS personalizzato che ascolta su HTTP con un numero di porta specifico e un percorso fisico personalizzato:

Example aws-windows-deployment-manifest.json - configurazione del sito Web IIS

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "websites": [
      {
        "name": "MyCustomSite",
        "physicalPath": "C:\inetpub\wwwroot\mysite",
        "bindings": [
          {
            "protocol": "http",
            "port": 8080,
            "hostName": "mysite.local"
          }
        ]
      }
    ]
  },
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "appBundle": "dotnet-core-app.zip",
          "iisWebSite": "MyCustomSite",
          "iisPath": "/"
        }
      }
    ]
  }
```

```
}
```

In questo esempio:

- Un sito Web denominato "MyCustomSite" viene creato con un percorso fisico personalizzato
- Il sito Web ha un'associazione HTTP sulla porta 8080 con un nome host specifico
- L'applicazione ASP.NET Core viene distribuita su questo sito Web personalizzato utilizzando il parametro `iisWebSite`

Utilizzo dell'Application Request Routing (ARR)

I moduli Application Request Routing (ARR) e URL Rewrite sono preinstallati e disponibili in Elastic Beanstalk Windows. AMIs Questi moduli consentono scenari di routing avanzati e la manipolazione degli URL tramite la configurazione IIS utilizzando ebextensions o la configurazione dell'applicazione.

L'esempio seguente mostra un semplice manifesto di distribuzione che configura un sito Web con una porta personalizzata, combinato con una configurazione ebextensions che imposta il routing ARR di base:

Example aws-windows-deployment-manifest.json: configurazione ARR semplice

```
{
  "manifestVersion": 1,
  "iisConfig": {
    "websites": [
      {
        "name": "ARRSite",
        "physicalPath": "C:\\inetpub\\wwwroot\\arrsite",
        "bindings": [
          {
            "protocol": "http",
            "port": 8080,
            "hostName": "localhost"
          }
        ]
      }
    ]
  },
  "deployments": {
    "aspNetCoreWeb": [
      {
```

```
        "name": "BackendApp",
        "parameters": {
          "appBundle": "backend-app.zip",
          "iisWebSite": "ARRSite",
          "iisPath": "/backend"
        }
      }
    ]
  }
}
```

La configurazione ARR viene eseguita tramite ebextensions. La seguente configurazione imposta le regole di routing ARR di base:

Example.ebextensions/arr-config.config - Configurazione ARR di base

```
files:
  "C:\\temp\\configure-arr.ps1":
    content: |
      # Enable ARR proxy at server level
      Set-WebConfigurationProperty -PSPath 'MACHINE/WEBROOT/APPHOST' -Filter
 'system.webServer/proxy' -Name 'enabled' -Value 'True'

      # Clear any existing global rules to avoid conflicts
      Clear-WebConfiguration -PSPath 'MACHINE/WEBROOT/APPHOST' -Filter
 'system.webServer/rewrite/globalRules'

      # Add global rule to route all requests to backend
      Add-WebConfigurationProperty -PSPath 'MACHINE/WEBROOT/APPHOST' `
        -Filter 'system.webServer/rewrite/globalRules' `
        -Name '.' `
        -Value @{
          name = 'Route_to_Backend'
          stopProcessing = 'True'
          match = @{ url = '^(?!backend/)(.*)' }
          action = @{
            type = 'Rewrite'
            url = 'http://localhost:8080/backend/{R:1}'
          }
        }

container_commands:
  01_configure_arr:
```

```
    command: powershell -ExecutionPolicy Bypass -File "C:\\temp\\configure-arr.ps1"
    waitAfterCompletion: 0
```

Questa configurazione crea un sito Web sulla porta 8080 e configura ARR per indirizzare tutte le richieste in arrivo all'applicazione di backend in esecuzione su quel sito.

Configurazione dei pool delle applicazioni

È possibile supportare più applicazioni nell'ambiente Windows. Sono disponibili due approcci:

- È possibile utilizzare il modello di out-of-process hosting con il server web Kestrel. Con questo modello, è possibile configurare più applicazioni per l'esecuzione in un unico pool di applicazioni.
- È possibile utilizzare il modello di hosting in-process. Con questo modello, è possibile utilizzare più pool di applicazioni per eseguire più applicazioni con una sola applicazione in ogni pool. Se utilizzi il server IIS e desideri eseguire più applicazioni, devi utilizzare questo approccio.

Per configurare Kestrel ed eseguire più applicazioni in un pool di applicazioni, aggiungi `hostingModel="OutofProcess"` nel file `web.config`. Considera i seguenti esempi:

Example web.config - per il modello di hosting Kestrel out-of-process

```
<configuration>
<location path="." inheritInChildApplications="false">
<system.webServer>
<handlers>
<add
    name="aspNetCore"
    path="*" verb="*"
    modules="AspNetCoreModuleV2"
    resourceType="Unspecified" />
</handlers>
<aspNetCore
    processPath="dotnet"
    arguments=".\CoreWebApp-5-0.dll"
    stdoutLogEnabled="false"
    stdoutLogFile=".\logs\stdout"
    hostingModel="OutofProcess" />
</system.webServer>
</location>
</configuration>
```

Example aws-windows-deployment-manifest.json: più applicazioni

```
{
"manifestVersion": 1,
  "deployments": {"msDeploy": [
      {"name": "Web-app1",
        "parameters": {"archive": "site1.zip",
          "iisPath": "/"
        }
      },
      {"name": "Web-app2",
        "parameters": {"archive": "site2.zip",
          "iisPath": "/app2"
        }
      }
    ]
  }
}
```

IIS non supporta più applicazioni in un pool di applicazioni perché utilizza il modello di hosting in-process. Pertanto, è necessario configurare più applicazioni assegnando ciascuna applicazione a un pool di applicazioni. In altre parole, assegnare una sola applicazione a un pool di applicazioni.

È possibile configurare IIS per utilizzare diversi pool di applicazioni nel file `aws-windows-deployment-manifest.json`. Apporta i seguenti aggiornamenti consultando il file di esempio successivo:

- Aggiungi una sezione `iisConfig` che includa una sottosezione denominata `appPools`.
- Nel blocco `appPools`, elenca i pool di applicazioni.
- Nella sezione `deployments`, definisci una sezione `parameters` per ogni applicazione.
- Per ogni applicazione, la sezione `parameters` specifica un archivio, un percorso per eseguirla e un `appPool` in cui eseguirla.

Il seguente manifest di distribuzione consente di configurare due pool di applicazioni che riavviano l'applicazione ogni 10 minuti. Essi inoltre allegano le loro applicazioni a un'applicazione Web .NET Framework che viene eseguita nel percorso specificato.

Example aws-windows-deployment-manifest.json: un'applicazione per pool di applicazioni

```
{
```

```
"manifestVersion": 1,
  "iisConfig": {"appPools": [
      {"name": "MyFirstPool",
       "recycling": {"regularTimeInterval": 10}
      },
      {"name": "MySecondPool",
       "recycling": {"regularTimeInterval": 10}
      }
     ]
    },
  "deployments": {"msDeploy": [
      {"name": "Web-app1",
        "parameters": {
           "archive": "site1.zip",
           "iisPath": "/",
           "appPool": "MyFirstPool"
           }
      },
      {"name": "Web-app2",
        "parameters": {
           "archive": "site2.zip",
           "iisPath": "/app2",
           "appPool": "MySecondPool"
          }
      }
     ]
    }
}
```

Definizione delle distribuzioni personalizzate

Per un maggiore controllo, è possibile personalizzare completamente la distribuzione di un'applicazione definendo una distribuzione personalizzata.

Questo manifesto di distribuzione indica a Elastic Beanstalk PowerShell di eseguire gli script in modalità a 32 bit. Specifica tre script: `install` uno script (`siteInstall.ps1`) che viene eseguito durante l'avvio e la distribuzione dell'istanza, uno `uninstall` script (`siteUninstall.ps1`) che viene eseguito prima di installare nuove versioni durante le distribuzioni e uno `restart` script (`siteRestart.ps1`) che viene eseguito quando si seleziona Riavvia App Server nella console di gestione. AWS

Example aws-windows-deployment-manifest.json: distribuzione personalizzata

```
{
  "manifestVersion": 1,
  "deployments": {
    "custom": [
      {
        "name": "Custom site",
        "architecture" : 32,
        "scripts": {
          "install": {
            "file": "siteInstall.ps1"
          },
          "restart": {
            "file": "siteRestart.ps1"
          },
          "uninstall": {
            "file": "siteUninstall.ps1"
          }
        }
      }
    ]
  }
}
```

Include eventuali artefatti necessari per eseguire l'applicazione nel bundle di origine con il manifest e gli script.

Example C - .zip ustom-site-bundle

```
.
|-- aws-windows-deployment-manifest.json
|-- siteInstall.ps1
|-- siteRestart.ps1
|-- siteUninstall.ps1
`-- site-contents.zip
```

Riferimento allo schema del manifesto di distribuzione

Il manifesto di distribuzione è un file JSON che definisce in che modo Elastic Beanstalk deve distribuire e configurare le applicazioni Windows. Questa sezione fornisce un riferimento completo per tutte le proprietà e le opzioni di configurazione supportate nello schema del manifesto.

Struttura del manifesto

Il manifesto di distribuzione segue uno schema JSON specifico con la seguente struttura di primo livello:

Example Struttura del manifesto di base

```
{
  "manifestVersion": 1,
  "skipIISReset": false,
  "iisConfig": {
    "websites": [...],
    "appPools": [...]
  },
  "deployments": {
    "msDeploy": [...],
    "aspNetCoreWeb": [...],
    "custom": [...]
  }
}
```

Proprietà di primo livello

`manifestVersion`(richiesto)

Tipo: numero

Impostazione predefinita: 1

Valori validi: 1

Speciifica la versione dello schema del manifesto. Attualmente è supportata solo la versione 1.

`skipIISReset` (facoltativo)

Tipo: Booleano

Di default: false

Controlla se IIS viene ripristinato durante le distribuzioni delle applicazioni. Questo flag influisce su entrambi i tipi `msDeploy` di `aspNetCoreWeb` distribuzione.

Comportamento:

- Non specificato o *`false`* (impostazione predefinita): le reimpostazioni di IIS vengono eseguite durante le operazioni di installazione, disinstallazione e aggiornamento. Questo è il comportamento tradizionale.
- *`true`*: le reimpostazioni IIS vengono ignorate durante le operazioni di distribuzione.

Vantaggi:

- Tempi di inattività ridotti: le applicazioni subiscono interruzioni del servizio più brevi durante le implementazioni.
- Implementazioni più rapide: elimina il tempo necessario a IIS per riavviare e reinizializzare completamente IIS.

> **Note**
>
> Durante l'utilizzo`skipIISReset`, l'RestartAppServeroperazione esegue un ripristino di IIS indipendentemente dall'impostazione di questa bandiera.

Esempio:

```
{
  "manifestVersion": 1,
  "skipIISReset": true,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

`deployments`(obbligatorio)

Tipo: oggetto

Contiene le configurazioni di distribuzione per le tue applicazioni. Questo oggetto può includere `msDeploy` e `aspNetCoreWeb` tipi di `custom` distribuzione.

`iisConfig` (facoltativo)

Tipo: oggetto

Definisce le impostazioni di configurazione IIS da applicare prima di distribuire le applicazioni. Supporta la configurazione del sito Web e del pool di applicazioni.

Configurazione IIS

La `iisConfig` sezione consente di configurare le impostazioni IIS prima di distribuire le applicazioni. Ciò include la configurazione di pool di applicazioni con configurazioni specifiche e la configurazione di siti Web IIS con associazioni personalizzate.

Siti Web IIS

I siti Web IIS consentono di configurare impostazioni personalizzate dei siti Web, inclusi percorsi fisici e collegamenti di rete, prima di distribuire le applicazioni.

Considerazioni importanti per la creazione di diversi siti Web IIS

- Ordine di configurazione dei siti Web: i siti Web vengono configurati in sequenza nell'ordine in cui appaiono nell'`websites`array. La piattaforma elabora ogni configurazione del sito Web in sequenza, quindi assicurati un ordine corretto in caso di dipendenze tra i siti Web.
- Firewall e accesso alle porte: solo la porta 80 viene esposta automaticamente tramite la configurazione firewall Windows Elastic Beanstalk predefinita. Se si configurano siti Web per utilizzare porte non standard, è necessario definire regole firewall personalizzate tramite ebextensions o script di distribuzione personalizzati per consentire l'accesso esterno a queste porte.

Example Configurazione del sito Web

```
{
  "iisConfig": {
    "websites": [
      {
        "name": "MyCustomSite",
        "physicalPath": "C:\inetpub\wwwroot\mysite",
        "bindings": [
```

```
          {
            "protocol": "http",
            "port": 8080,
            "hostName": "mysite.local"
          },
          {
            "protocol": "https",
            "port": 8443
          }
        ]
      }
    ]
  }
}
```

Proprietà del sito Web

`name`(richiesto)

- Tipo: stringa

Il nome del sito Web IIS. Questo nome viene utilizzato per identificare il sito Web in IIS Manager e deve essere univoco all'interno della configurazione IIS.

`physicalPath`(richiesto)

- Tipo: stringa

Il percorso fisico sul server in cui sono archiviati i file del sito Web. Questo percorso deve essere accessibile al processo di lavoro IIS.

`bindings`(obbligatorio)

Tipo: matrice

Numero minimo di articoli: 1

Una serie di configurazioni vincolanti che definiscono il modo in cui il sito Web risponde alle richieste di rete. Ogni associazione specifica un protocollo, una porta e un nome host opzionale.

Associazioni a siti Web

Le associazioni ai siti Web definiscono gli endpoint di rete in cui il sito Web IIS ascolterà le richieste in arrivo.

`protocol`(richiesto)

- Tipo: stringa

Valori validi: «http», «https»

Il protocollo utilizzato per l'associazione.

`port`(richiesto)

Tipo: integer

Intervallo valido: 1-65535

Il numero di porta su cui il sito Web ascolterà le richieste.

`hostName` (facoltativo)

- Tipo: stringa

Il nome host (nome di dominio) per l'associazione.

Pool di applicazioni

I pool di applicazioni forniscono l'isolamento tra le applicazioni e consentono di configurare le impostazioni di runtime per gruppi di applicazioni.

Example Configurazione del pool di applicazioni

```
{
  "iisConfig": {
    "appPools": [
      {
        "name": "MyAppPool",
        "enable32Bit": false,
        "managedPipelineMode": "Integrated",
        "managedRuntimeVersion": "v4.0",
        "queueLength": 1000,
        "cpu": {
          "limitPercentage": 80,
          "limitAction": "Throttle",
          "limitMonitoringInterval": 5
```

```
        },
        "recycling": {
          "regularTimeInterval": 1440,
          "requestLimit": 10000,
          "memory": 1048576,
          "privateMemory": 524288
        }
      }
    ]
  }
}
```

Proprietà del pool di applicazioni

`name`(richiesto)

- Tipo: stringa

Il nome del pool di applicazioni. Questo nome viene utilizzato per fare riferimento al pool nelle configurazioni di distribuzione.

`enable32Bit` (facoltativo)

Tipo: Booleano

Consente l'esecuzione di un'applicazione a 32 bit su una versione a 64 bit di Windows. Impostato su `true` per le applicazioni legacy che richiedono la compatibilità a 32 bit.

`managedPipelineMode` (facoltativo)

- Tipo: stringa

Valori validi: «Integrato», «Classico»

Speciifica la modalità di elaborazione delle richieste per il pool di applicazioni.

`managedRuntimeVersion` (facoltativo)

- Tipo: stringa

Valori validi: «No Managed Code», «v2.0", «v4.0"

Speciifica la versione.NET Framework per il pool di applicazioni.

`queueLength` (facoltativo)

Tipo: integer

Numero massimo di richieste che HTTP.sys mette in coda per il pool di applicazioni prima di rifiutare richieste aggiuntive.

Configurazione della CPU

L'`cpu`oggetto configura i limiti di utilizzo della CPU e il monitoraggio per il pool di applicazioni.

`limitPercentage` (facoltativo)

Tipo: numero

Percentuale massima di tempo di CPU che i processi di lavoro nel pool di applicazioni possono consumare.

`limitAction` (facoltativo)

- Tipo: stringa

Valori validi: "NoAction«, «killW3WP», «Throttle», "» ThrottleUnderLoad

Azione da intraprendere quando viene raggiunto il limite della CPU.

`limitMonitoringInterval` (facoltativo)

Tipo: numero

Periodo di ripristino (in minuti) per il monitoraggio della CPU e i limiti di limitazione.

Configurazione del riciclaggio

L'`recycling`oggetto configura quando e come i processi di lavoro del pool di applicazioni vengono riciclati.

`regularTimeInterval` (facoltativo)

Tipo: integer

Intervallo di tempo (in minuti) dopo il quale il pool di applicazioni viene riciclato. Impostare su 0 per disabilitare il riciclo basato sul tempo.

`requestLimit` (facoltativo)

Tipo: integer

Numero massimo di richieste elaborate dal pool di applicazioni prima del riciclaggio.

`memory` (facoltativo)

Tipo: integer

Quantità di memoria virtuale (in kilobyte) che attiva il riciclaggio dei processi di lavoro.

`privateMemory` (facoltativo)

Tipo: integer

Quantità di memoria privata (in kilobyte) che attiva il riciclaggio dei processi di lavoro.

Tipi di implementazione

L'`deployments`oggetto contiene matrici di configurazioni di distribuzione per diversi tipi di applicazioni. Ogni tipo di distribuzione ha proprietà e casi d'uso specifici.

MSDeploy distribuzioni

MSDeploy le distribuzioni vengono utilizzate per le tradizionali applicazioni.NET Framework che possono essere distribuite utilizzando Web Deploy (). MSDeploy

Example MSDeploy configurazione di distribuzione

```
{
  "deployments": {
    "msDeploy": [
      {
        "name": "WebApp",
        "description": "Main web application",
        "parameters": {
          "appBundle": "webapp.zip",
          "iisPath": "/",
          "appPool": "DefaultAppPool"
        }
      }
```

```
    ]
  }
}
```

MSDeploy proprietà di distribuzione

`name`(richiesto)

- Tipo: stringa

Nome univoco per la distribuzione. Questo nome deve essere univoco in tutte le distribuzioni del manifesto.

`description` (facoltativo)

- Tipo: stringa

Descrizione della distribuzione in formato leggibile dall'uomo.

`parameters`(richiesto)

Tipo: oggetto

Parametri di configurazione per l' MSDeploy operazione.

`scripts` (facoltativo)

Tipo: oggetto

PowerShell script da eseguire in varie fasi del ciclo di vita dell'implementazione.

MSDeploy parametri

`appBundle`(richiesto)

- Tipo: stringa

Percorso del pacchetto dell'applicazione (file ZIP) relativo al file manifest. Questo pacchetto contiene i file dell'applicazione da distribuire.

`iisWebSite` (facoltativo)

- Tipo: stringa

Predefinito: «Sito Web predefinito»

Il sito Web IIS su cui distribuire l'applicazione. Per impostazione predefinita, le applicazioni vengono distribuite nel «Sito Web predefinito». Facoltativamente, è possibile specificare un nome di sito Web diverso, ad esempio quello configurato nella `iisConfig.websites` sezione.

`iisPath` (facoltativo)

- Tipo: stringa

Predefinito: «/»

Percorso della directory virtuale in IIS in cui verrà distribuita l'applicazione. Usa «/» per il percorso principale o «/api» per una sottodirectory.

`appPool` (facoltativo)

- Tipo: stringa

Nome del pool di applicazioni per eseguire questa applicazione.

Distribuzioni ASP.NET Core

Le distribuzioni ASP.NET Core sono progettate specificamente per le applicazioni.NET Core e.NET 5+.

Example Configurazione della distribuzione di ASP.NET Core

```
{
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "CoreAPI",
        "description": "ASP.NET Core Web API",
        "parameters": {
          "appBundle": "coreapi.zip",
          "iisPath": "/api",
          "appPool": "CoreAppPool"
        }
      }
    ]
  }
}
```

Le distribuzioni ASP.NET Core utilizzano la stessa struttura di proprietà delle MSDeploy distribuzioni, con la differenza fondamentale nell'ambiente di runtime e nel modello di hosting utilizzati per l'applicazione.

Parametri di distribuzione di ASP.NET Core

`appBundle`(richiesto)

- Tipo: stringa

  Percorso del bundle dell'applicazione relativo al file manifest. Può trattarsi di un archivio ZIP o di un percorso di directory contenente l'applicazione ASP.NET Core pubblicata.

`iisWebSite` (facoltativo)

- Tipo: stringa

  Predefinito: «Sito Web predefinito»

  Il sito Web IIS su cui distribuire l'applicazione ASP.NET Core. Per impostazione predefinita, le applicazioni vengono distribuite nel «Sito Web predefinito». Facoltativamente, è possibile specificare un nome di sito Web diverso, ad esempio quello configurato nella `iisConfig.websites` sezione.

`iisPath` (facoltativo)

- Tipo: stringa

  Predefinito: «/»

  Percorso della directory virtuale in IIS per l'applicazione ASP.NET Core.

`appPool` (facoltativo)

- Tipo: stringa

  Pool di applicazioni per l'applicazione ASP.NET Core. Il pool verrà configurato in modo appropriato per l'hosting ASP.NET Core.

Distribuzioni personalizzate

Le distribuzioni personalizzate forniscono il controllo completo sul processo di distribuzione tramite script. PowerShell Questo tipo di distribuzione è utile per scenari complessi che richiedono installazione, configurazione o logica di distribuzione personalizzate.

Example Configurazione di distribuzione personalizzata

```
{
  "deployments": {
    "custom": [
      {
        "name": "CustomService",
        "description": "Custom Windows service deployment",
        "architecture": 32,
        "scripts": {
          "install": {
            "file": "install-service.ps1"
          },
          "restart": {
            "file": "restart-service.ps1"
          },
          "uninstall": {
            "file": "uninstall-service.ps1",
            "ignoreErrors": true
          }
        }
      }
    ]
  }
}
```

Proprietà di distribuzione personalizzate

`name`(obbligatorio)

- Tipo: stringa

Nome univoco per la distribuzione personalizzata.

`description` (facoltativo)

- Tipo: stringa

Descrizione della distribuzione personalizzata.

`architecture` (facoltativo)

Tipo: integer

Valore predefinito: 32

Valori validi: 32, 64

Le specifiche di architettura per la modalità di esecuzione degli script PowerShell

`scripts`(richiesto)

Tipo: oggetto

PowerShell script che definiscono il comportamento di distribuzione. Le distribuzioni personalizzate supportano tipi di script aggiuntivi rispetto ad altri tipi di distribuzione.

Script di distribuzione

Gli script di distribuzione sono PowerShell script che vengono eseguiti in momenti specifici durante il ciclo di vita della distribuzione. Diversi tipi di distribuzione supportano diversi set di eventi di script.

Eventi di script

I seguenti eventi di script sono disponibili a seconda del tipo di distribuzione:

Script di distribuzione standard (MSDeploy e Web) aspNetCore

`preInstall`

Viene eseguito prima dell'installazione o dell'aggiornamento dell'applicazione.

`postInstall`

Viene eseguito dopo l'installazione o l'aggiornamento dell'applicazione.

`preRestart`

Viene eseguito prima del riavvio dell'applicazione.

`postRestart`

Viene eseguito dopo il riavvio dell'applicazione.

`preUninstall`

Viene eseguito prima della disinstallazione dell'applicazione.

`postUninstall`

Viene eseguito dopo la disinstallazione dell'applicazione.

Script di distribuzione personalizzati (solo distribuzioni personalizzate)

`install`

Script di installazione principale per distribuzioni personalizzate. Questo script è responsabile dell'installazione dell'applicazione o del servizio.

`restart`

Script per riavviare l'applicazione o il servizio. Chiamato al riavvio dell'ambiente.

`uninstall`

Script per disinstallare l'applicazione o il servizio. Chiamato durante la chiusura dell'ambiente o la rimozione dell'applicazione.

Proprietà dello script

Ogni script è definito come un oggetto con le seguenti proprietà:

`file`(obbligatorio)

- Tipo: stringa

Percorso del file di PowerShell script relativo al file manifest. Lo script deve avere un'`.ps1`estensione.

`ignoreErrors` (facoltativo)

Tipo: Booleano

Di default: false

Se impostato su`true`, la distribuzione continua anche se lo script fallisce. Utilizzalo per script o operazioni di pulizia non critici.

Example Esempio di configurazione dello script

```
{
  "scripts": {
    "preInstall": {
      "file": "backup-config.ps1",
      "ignoreErrors": true
    },
```

```
    "postInstall": {
      "file": "configure-app.ps1"
    }
  }
}
```

Utilizzo di EC2 Fast Launch con i rami della piattaforma Windows

La funzionalità EC2 Fast Launch riduce i tempi di avvio delle istanze Windows negli ambienti Elastic Beanstalk. Lo scopo di questo argomento è di guidarti nell'utilizzo di questa funzionalità con i tuoi ambienti Elastic Beanstalk. A partire dalla versione 2.16.2 della piattaforma Windows, rilasciata il 22 gennaio 2025, le versioni della piattaforma Elastic Beanstalk includono la versione base con Fast Launch abilitato. AMIs EC2

Disponibilità predefinita di Fast Launch EC2

Le ultime versioni della piattaforma Elastic Beanstalk Windows AMIs includono una EC2 base con Fast Launch abilitato automaticamente, senza costi aggiuntivi. Tuttavia, quando vengono rilasciate versioni più recenti della piattaforma, EC2 Fast Launch potrebbe non rimanere abilitato automaticamente sulla base delle versioni precedenti AMIs della piattaforma.

Si consiglia di eseguire l'aggiornamento alla versione più recente della piattaforma Windows per utilizzare la versione base AMIs con EC2 Fast Launch abilitato automaticamente. Tuttavia, se è necessario continuare a utilizzare la versione della piattaforma esistente, è possibile abilitare manualmente EC2 Fast Launch sull'AMI di base del proprio ambiente. Per istruzioni, consulta Configurazione manuale di EC2 Fast Launch.

Configurazione manuale di EC2 Fast Launch

> **Note**
>
> L'attivazione manuale di EC2 Fast Launch può comportare costi aggiuntivi rispetto all'utilizzo di versioni della piattaforma con EC2 Fast Launch abilitato automaticamente. Per ulteriori informazioni sui costi di EC2 Fast Launch, consulta la pagina delle risorse sottostanti per Manage costs for EC2 Fast Launch nella Amazon EC2 User Guide.

Segui questi passaggi per abilitare EC2 Fast Launch su un'AMI di base Windows utilizzata dall'ambiente Elastic Beanstalk:

Per abilitare manualmente EC2 Fast Launch per il tuo ambiente Elastic Beanstalk

1. Identifica l'AMI di base del tuo ambiente:

   Segui i passaggi descritti in Creazione di un'AMI personalizzata per identificare l'ID AMI di base del tuo ambiente. Tieni presente che non è necessario creare un AMI personalizzato: devi solo seguire i passaggi per individuare l'ID AMI di base corrente.
2. Abilita l'avvio EC2 rapido sull'AMI:

   Utilizza le istruzioni in Enable EC2 Fast Launch nella Amazon EC2 User Guide per configurare EC2 Fast Launch per la tua AMI.

Aggiunta di un'istanza database Amazon RDS all'ambiente applicativo .NET

Questo argomento fornisce istruzioni per creare un Amazon RDS utilizzando la console Elastic Beanstalk. Puoi utilizzare un'istanza database Amazon Relational Database Service (Amazon RDS) per archiviare i dati raccolti e modificati dall'applicazione. Il database può essere accoppiato all'ambiente e gestito da Elastic Beanstalk oppure può essere creato e gestito esternamente da un altro servizio. In queste istruzioni il database è accoppiato all'ambiente e gestito da Elastic Beanstalk. Per ulteriori informazioni sull'integrazione di un Amazon RDS con Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Sections

- Aggiunta di un'istanza database all'ambiente
- Download di un driver
- Connessione a un database

Aggiunta di un'istanza database all'ambiente

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).

4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'aggiunta di un'istanza database richiede circa 10 minuti. Quando l'aggiornamento dell'ambiente è completo, il nome host dell'istanza database e altre informazioni di connessione sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente:

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Per ulteriori informazioni sulla configurazione di un'istanza di database accoppiata a un ambiente Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Download di un driver

Scarica e installa il pacchetto `EntityFramework` e un driver di database per l'ambiente di sviluppo con `NuGet`.

Provider di database Entity Framework comuni per. NET

- SQL Server – `Microsoft.EntityFrameworkCore.SqlServer`
- MySQL – `Pomelo.EntityFrameworkCore.MySql`
- PostgreSQL – `Npgsql.EntityFrameworkCore.PostgreSQL`

Connessione a un database

Elastic Beanstalk fornisce informazioni di connessione per le istanze database collegate nelle proprietà dell'ambiente. Utilizza `ConfigurationManager.AppSettings` per leggere le proprietà e configurare una connessione di database.

Example Helpers.cs: metodo della stringa di connessione

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;

namespace MVC5App.Models
{
  public class Helpers
  {
    public static string GetRDSConnectionString()
    {
      var appConfig = ConfigurationManager.AppSettings;

      string dbname = appConfig["RDS_DB_NAME"];

      if (string.IsNullOrEmpty(dbname)) return null;

      string username = appConfig["RDS_USERNAME"];
      string password = appConfig["RDS_PASSWORD"];
      string hostname = appConfig["RDS_HOSTNAME"];
```

```
      string port = appConfig["RDS_PORT"];

      return "Data Source=" + hostname + ";Initial Catalog=" + dbname + ";User ID=" +
 username + ";Password=" + password + ";";
    }
  }
}
```

Utilizza la stringa di connessione per inizializzare il contesto del database.

Example DBContext.cs

```
using System.Data.Entity;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace MVC5App.Models
{
  public class RDSContext : DbContext
  {
    public RDSContext()
      : base(GetRDSConnectionString())
    {
    }

    public static RDSContext Create()
    {
      return new RDSContext();
    }
  }
}
```

La AWS Toolkit for Visual Studio

Visual Studio fornisce modelli per differenti tipi di applicazioni e linguaggi di programmazione. Puoi iniziare con uno qualsiasi di questi modelli. Il AWS Toolkit for Visual Studio fornisce anche tre modelli di progetto che avviano lo sviluppo dell'applicazione AWS : Console Project AWS , Web Project AWS ed Empty Project. In questo esempio, creerai una nuova applicazione Web ASP.NET.

Creazione di un nuovo progetto di applicazione Web ASP.NET

1. In Visual Studio, nel menu File, fai clic su Nuovo, quindi su Progetto.
2. Nella finestra di dialogo Nuovo progetto, fai clic su Modelli installati, su Visual C#, quindi su Web. Fai clic su Applicazione Web ASP.NET vuota, digita un nome di progetto, quindi fai clic su OK.

Esecuzione di un progetto

Scegliere una delle seguenti operazioni:

1. Premi F5.
2. Seleziona Avvia debug dal menu Debug.

Test in locale

Visual Studio semplifica il test in locale delle applicazioni. Per testare o eseguire le applicazioni Web ASP.NET, devi disporre di un server Web. Visual Studio offre varie opzioni, ad esempio Internet Information Services (IIS), IIS Express o il server di sviluppo Visual Studio integrato. Per informazioni su ognuna di queste opzioni e per stabilire qual è la più appropriata per le tue esigenze, consulta Server Web in Visual Studio per progetti Web ASP.NET.

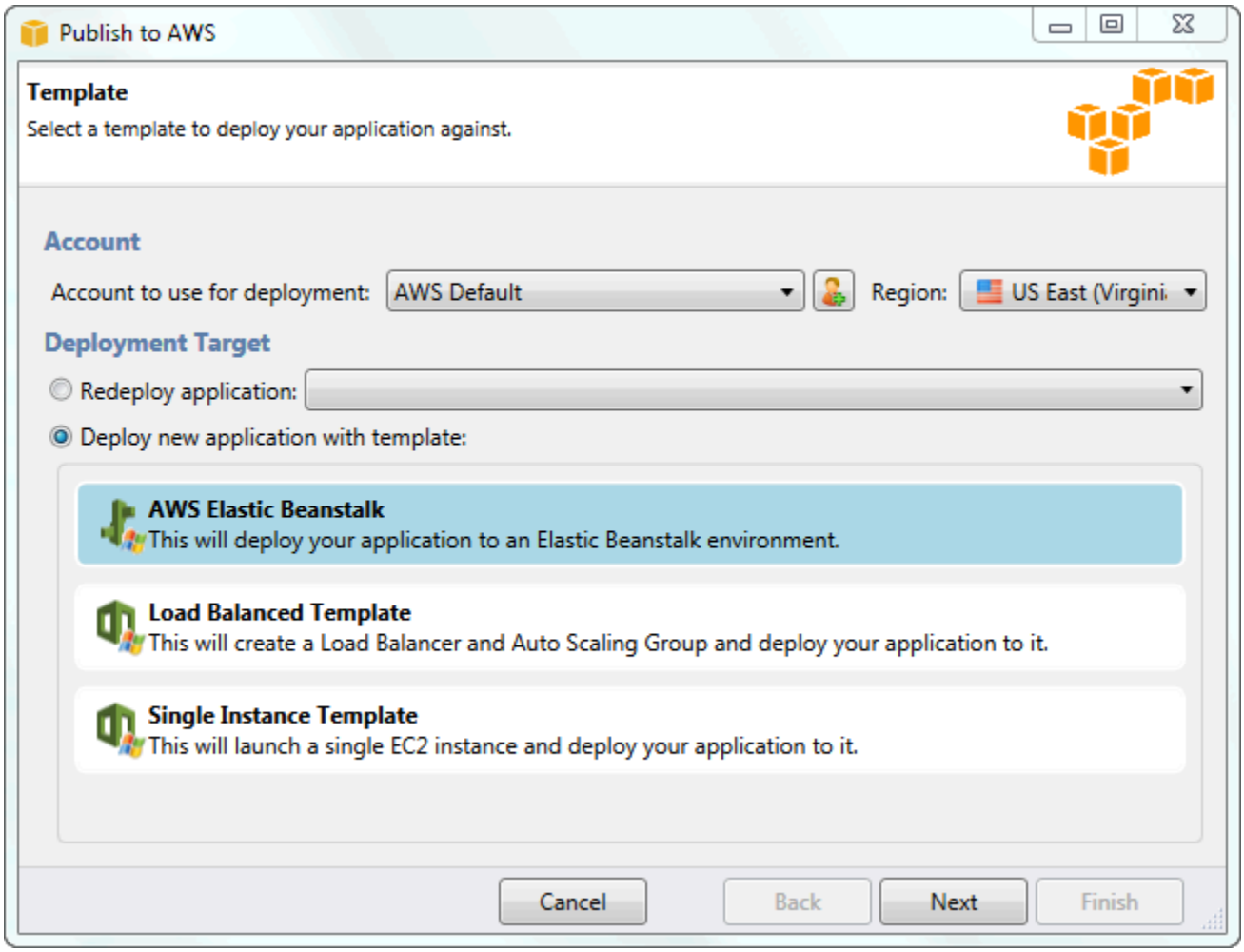
Creazione di un ambiente Elastic Beanstalk

Dopo aver testato l'applicazione, puoi distribuirla su Elastic Beanstalk.

> **Note**
>
> Il file di configurazione deve far parte del progetto per essere incluso nell'archivio. In alternativa, anziché includere i file di configurazione nel progetto, puoi utilizzare Visual Studio per distribuire tutti i file nella cartella di progetto. In Esplora soluzioni, fai cli con il pulsante destro del mouse sul nome del progetto e scegli Proprietà. Fai clic sulla scheda Pubblicazione/creazione pacchetto Web. Nella sezione Elementi da distribuire, seleziona Tutti i file nella cartella del progetto dall'elenco a discesa.

Per distribuire l'applicazione su Elastic AWS Beanstalk utilizzando il toolkit per Visual Studio

1. In Solution Explorer, fai clic con il pulsante destro del mouse sull'applicazione e seleziona Pubblica su. AWS
2. Nella procedura guidata Publish to (Pubblica su) AWS, inserisci le informazioni sul tuo account.
   a. Per AWS account to use for deployment (account da utilizzare per l'implementazione), seleziona il tuo account oppure Other (Altro) per inserire le informazioni su un altro account.
   b. In Region (Regione), seleziona la regione in cui vuoi distribuire l'applicazione. Per informazioni sulle AWS regioni disponibili, consulta AWS Elastic Beanstalk Endpoints and Quotas in. Riferimenti generali di AWS Se selezioni una regione non supportata da Elastic Beanstalk, l'opzione per la distribuzione in Elastic Beanstalk non sarà disponibile.
   c. Fai clic su Deploy new application with template (Distribuisci nuova applicazione con modello) e seleziona Elastic Beanstalk. Fai clic su Next (Successivo).



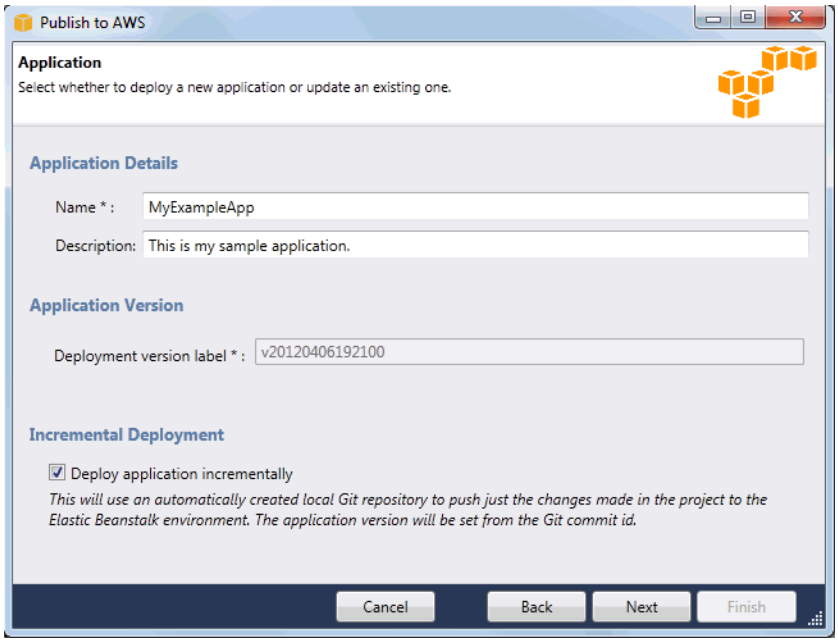
3. Nella pagina Application (Applicazione), immetti i dettagli sull'applicazione.

a. In Name (Nome), digita il nome dell'applicazione.

b. In Description (Descrizione), digita una descrizione dell'applicazione. Questa operazione è facoltativa.

c. L'etichetta versione dell'applicazione viene visualizzata automaticamente in Deployment version label (Etichetta versione distribuzione).

d. Seleziona Deploy application incrementally (Distribuisci applicazione in modo incrementale) per distribuire solo i file modificati. Una distribuzione incrementale è più veloce in quanto vengono aggiornati solo i file modificati e non tutti i file. Se scegli questa opzione, una versione dell'applicazione verrà definita a partire dall'ID di commit Git. Se scegli di non distribuire l'applicazione in modo incrementale, puoi aggiornare l'etichetta versione nella casella Deployment version label (Etichetta versione distribuzione).



e. Fare clic su Next (Successivo).

4. Nella pagina Environment (Ambiente), digita i dettagli sull'ambiente.

a. Seleziona Create a new environment for this application (Crea nuovo ambiente per questa applicazione).

b. In Name (Nome), digita un nome per il tuo ambiente.

c. In Description (Descrizione), digita una descrizione dell'ambiente. Questa operazione è facoltativa.

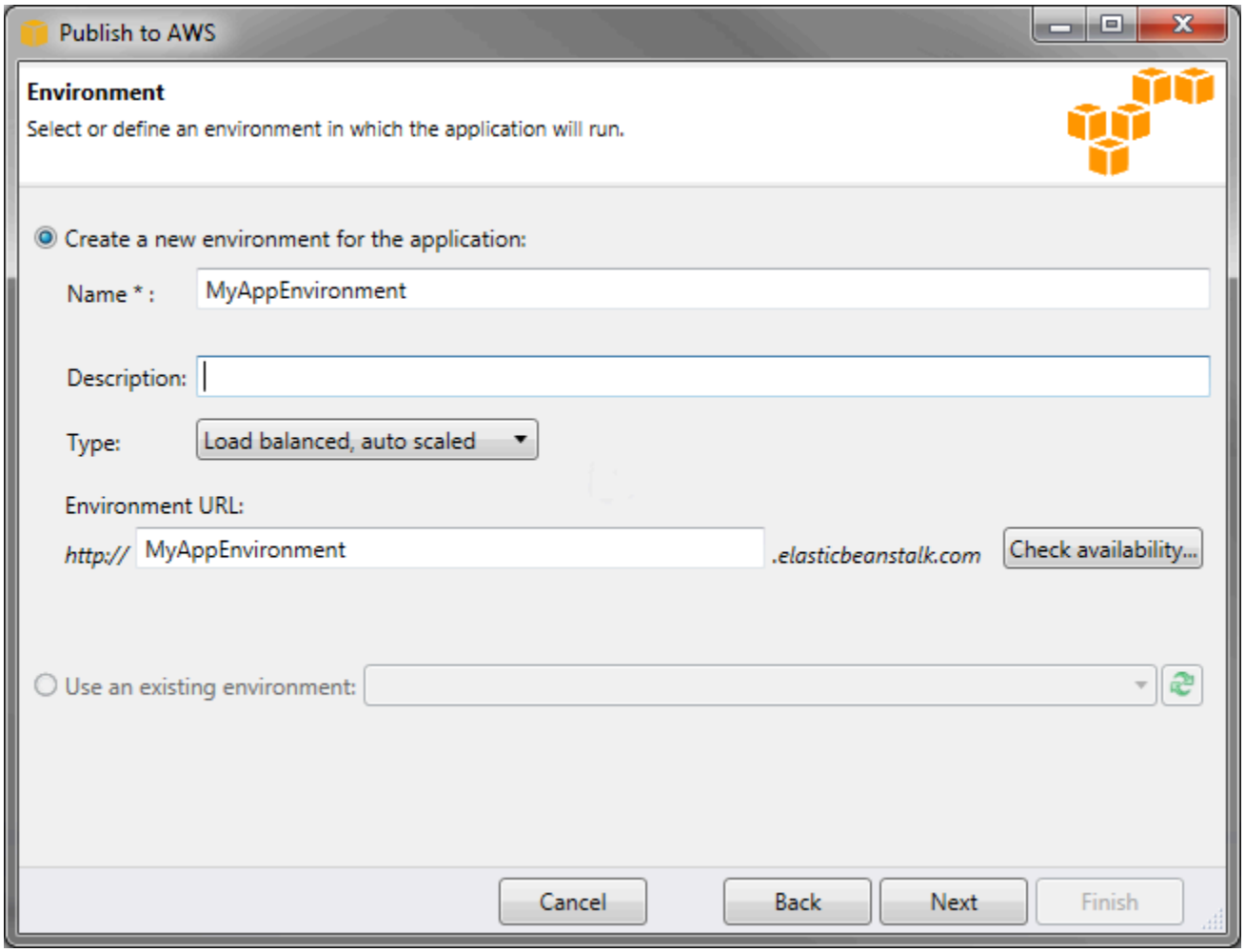
d. In Type (Tipo), seleziona il tipo di ambiente.

Puoi selezionare un ambiente Load balanced, auto scaled (Carico bilanciato, dimensionamento automatico) oppure Single instance (Singola istanza). Per ulteriori informazioni, consulta Tipi di ambiente.

> **Note**
>
> Per gli ambienti con una singola istanza, le impostazioni relative a bilanciamento del carico, Auto Scaling e URL per il controllo dello stato non sono disponibili.

e. L'URL dell'ambiente viene visualizzato automaticamente in Environment URL (URL ambiente) quando sposti il cursore su quella casella.

f. Fai clic su Check availability (Verifica disponibilità) per verificare che l'URL dell'ambiente sia disponibile.



g. Fare clic su Next (Successivo).

5. Nella pagina AWS Options (Opzioni), configura ulteriori opzioni e informazioni per l'implementazione.

   a. In Container Type (Tipo contenitore), seleziona 64bit Windows Server 2012 running IIS 8 (Windows Server 2012 a 64 bit con IIS 8) o 64bit Windows Server 2008 running IIS 7.5 (Windows Server 2008 a 64 bit con IIS 7.5).

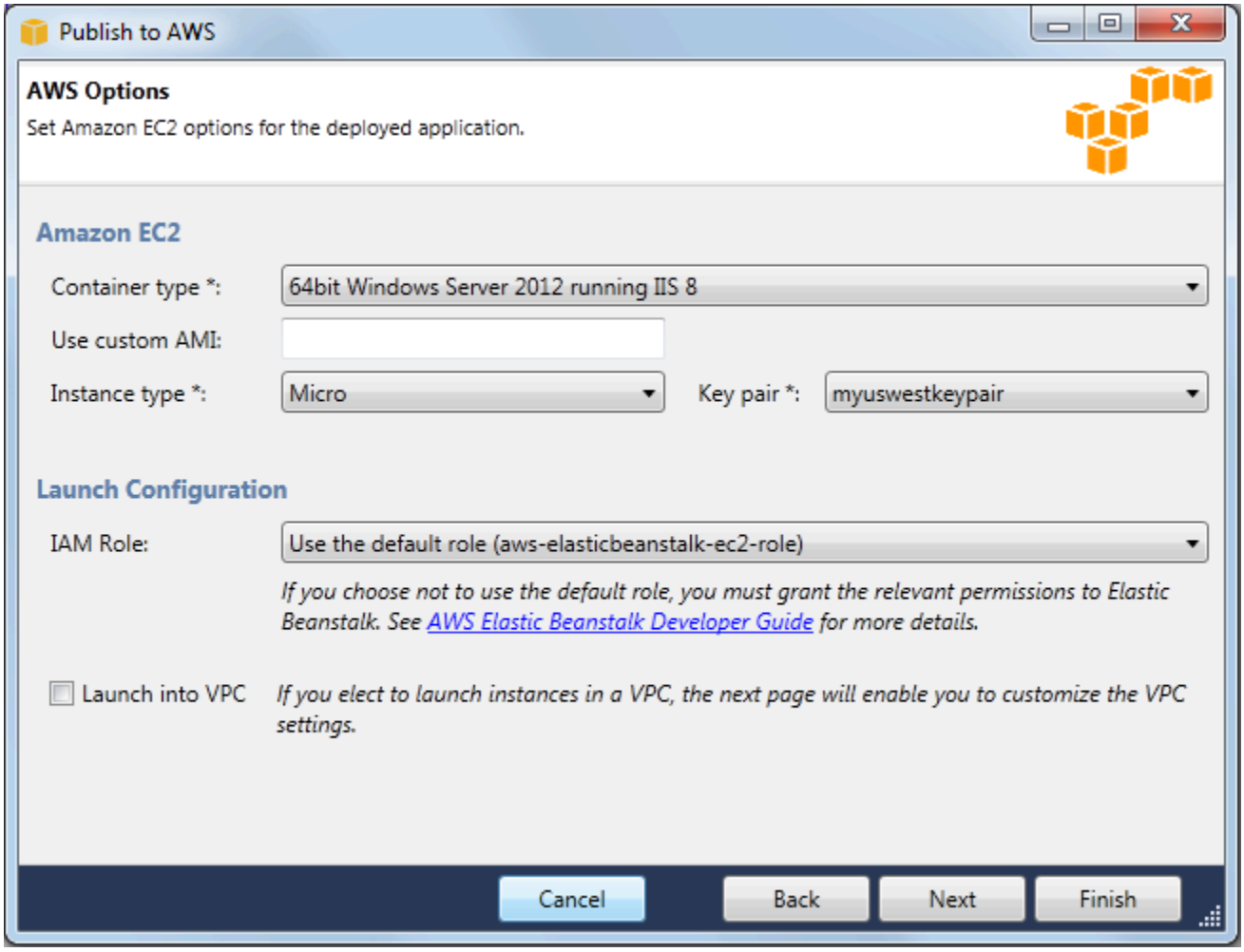
   b. In Instance Type (Tipo di istanza), seleziona Micro.

   c. In Key pair (Coppia di chiavi), seleziona Create new key pair (Crea nuova coppia di chiavi). Digita un nome per la nuova coppia di chiavi, in questo esempio **`myuswestkeypair`**, quindi fai clic su OK. Una coppia di key pair consente l'accesso da desktop remoto alle tue istanze Amazon EC2 . Per ulteriori informazioni sulle coppie di EC2 chiavi Amazon, consulta Using Credentials nella Amazon Elastic Compute Cloud User Guide.

   d. Seleziona un profilo dell'istanza.

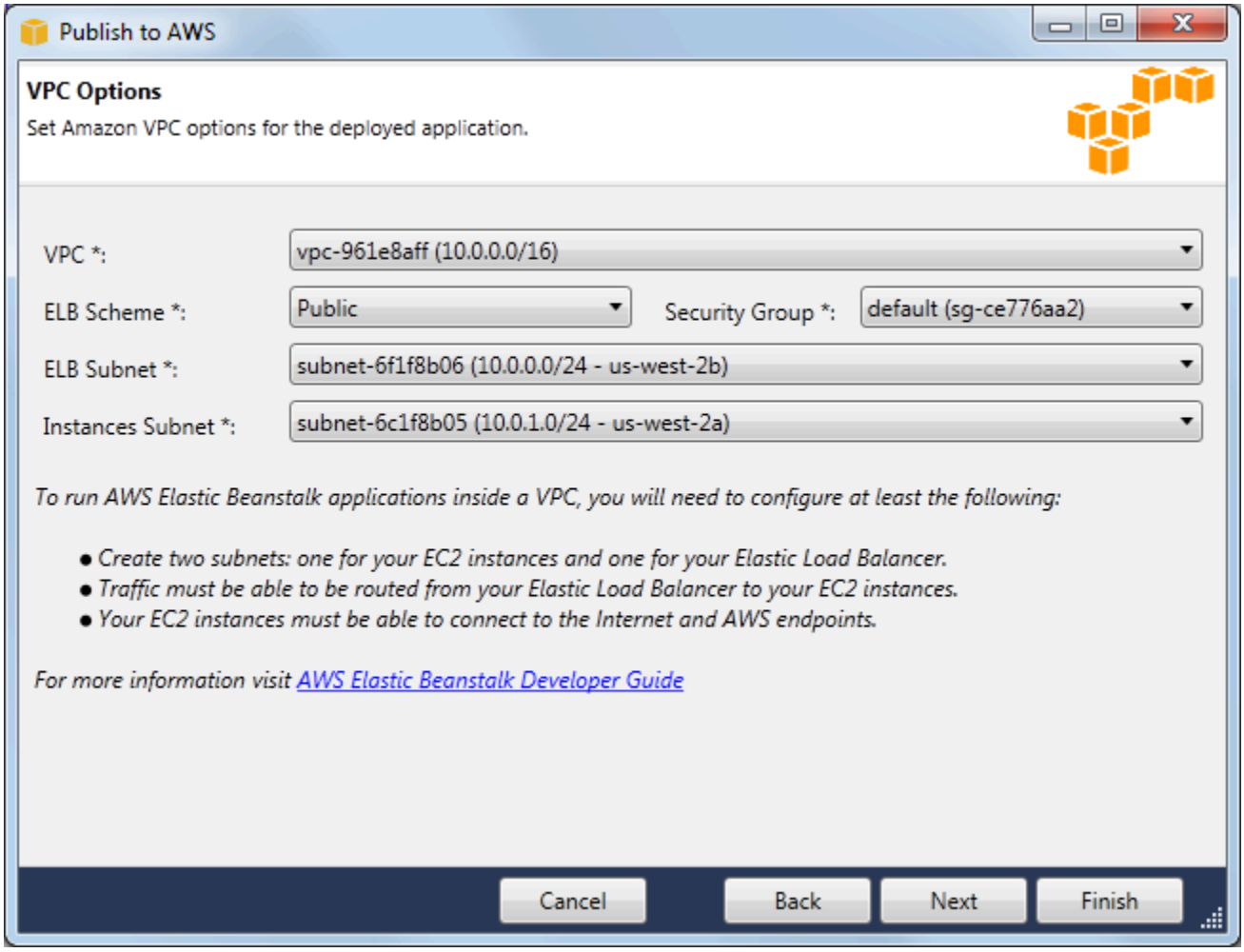
      Se non disponi di un profilo dell'istanza, seleziona Create a default instance profile (Crea profilo dell'istanza di default). Per informazioni sull'utilizzo dei profili dell'istanza con Elastic Beanstalk, consulta Gestione dei profili dell'istanza Elastic Beanstalk.

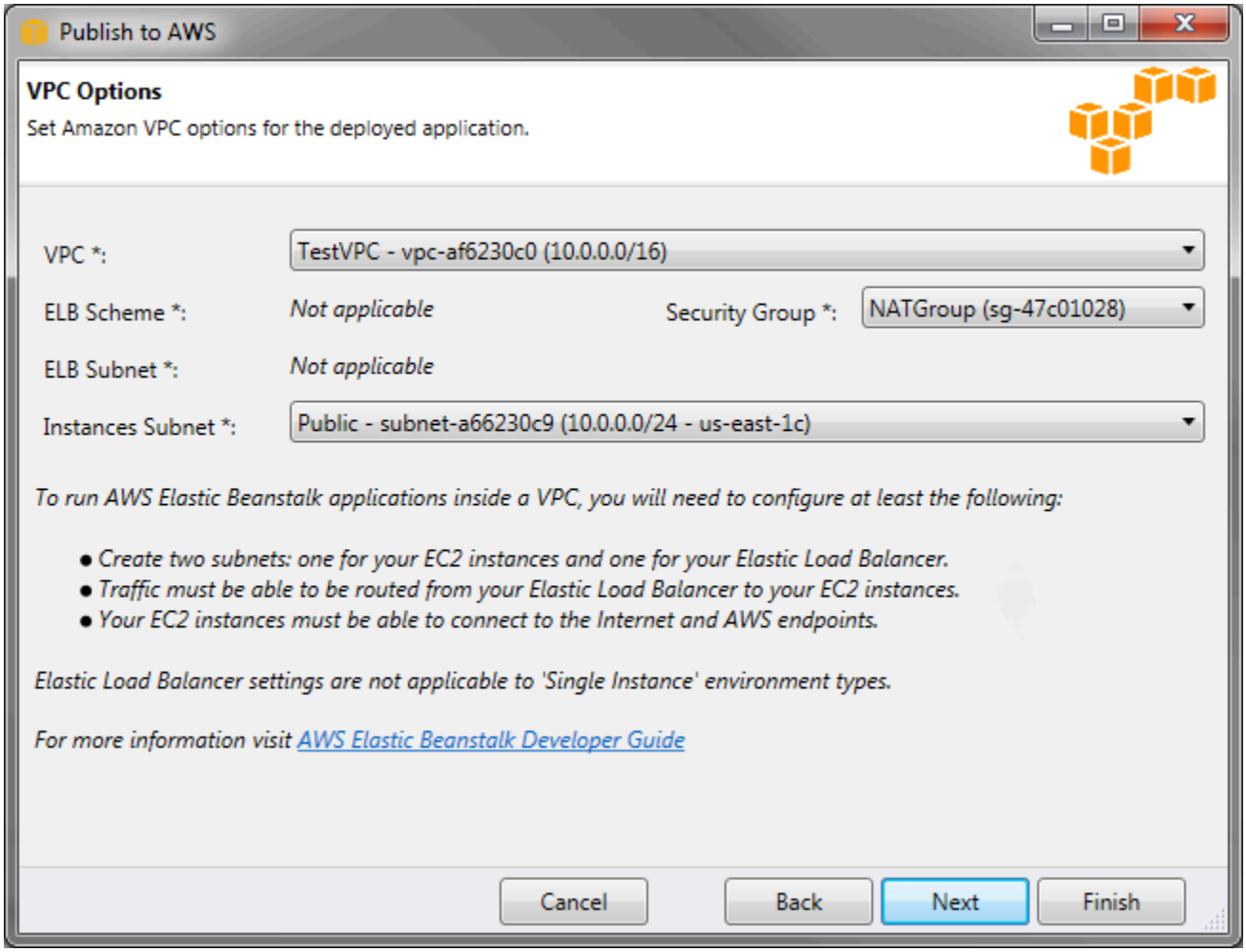
   e. Se hai un VPC personalizzato che desideri utilizzare con l'ambiente, fai clic su Launch into VPC (Avvia in VPC). Puoi configurare le informazioni sul VPC nella pagina successiva. Per ulteriori informazioni su Amazon VPC, consulta Amazon Virtual Private Cloud (Amazon VPC). Per un elenco di tipi di container non legacy supportati, consultare the section called “Perché alcune versioni di piattaforma sono contrassegnate come legacy?”



f. Fare clic su Next (Successivo).

6. Se hai scelto di avviare l'ambiente in un VPC, viene visualizzata la pagina VPC Options (Opzioni VPC); in caso contrario, viene visualizzata la pagina Additional Options (Opzioni aggiuntive). In questa pagina, puoi configurare le opzioni relative al VPC.


Publish to AWS
VPC Options
Set Amazon VPC options for the deployed application.
VPC *:
vpc-961e8aff (10.0.0.0/16)
ELB Scheme *:
Public
Security Group *:
default (sg-ce776aa2)
ELB Subnet *:
subnet-6f1f8b06 (10.0.0.0/24 - us-west-2b)
Instances Subnet *:
subnet-6c1f8b05 (10.0.1.0/24 - us-west-2a)
To run AWS Elastic Beanstalk applications inside a VPC, you will need to configure at least the following:
Create two subnets: one for your EC2 instances and one for your Elastic Load Balancer.
Traffic must be able to be routed from your Elastic Load Balancer to your EC2 instances.
Your EC2 instances must be able to connect to the Internet and AWS endpoints.
For more information visit AWS Elastic Beanstalk Developer Guide
Cancel
Back
Next
Finish



a. Seleziona l'ID del VPC in cui desideri avviare l'ambiente.

b. Per un ambiente con bilanciamento del carico e dimensionabile, seleziona private (privato) per ELB Scheme (Schema ELB) se non vuoi che Elastic Load Balancer sia disponibile su Internet.

   Per un ambiente con una singola istanza, questa opzione non è applicabile in quanto l'ambiente non dispone di un sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Tipi di ambiente.

c. Per un ambiente scalabile e con bilanciamento del carico, seleziona le sottoreti per il bilanciamento del carico elastico e le istanze. EC2 Se hai creato sottoreti pubbliche e private, assicurati che il bilanciamento del carico elastico e le istanze siano associati alla sottorete corretta. EC2 Per impostazione predefinita, Amazon VPC crea una sottorete pubblica di default utilizzando 10.0.0.0/24 e una sottorete privata utilizzando 10.0.1.0/24. Puoi visualizzare le sottoreti esistenti nella console Amazon VPC all'indirizzo. https://console.aws.amazon.com/vpc/

Per un ambiente con una singola istanza, il VPC richiede solo una sottorete pubblica per l'istanza. La selezione di una sottorete per il sistema di bilanciamento del carico non è possibile in quanto l'ambiente non dispone di un sistema di bilanciamento del carico. Per ulteriori informazioni, consulta Tipi di ambiente.

d. Per un ambiente con bilanciamento del carico e dimensionabile, seleziona il gruppo di sicurezza creato per le istanze, se applicabile.

   Per un ambiente con una singola istanza, non è necessario specificare un dispositivo NAT. Seleziona il gruppo di sicurezza di default. Elastic Beanstalk assegna un indirizzo IP elastico all'istanza per l'accesso a Internet.

e. Fare clic su Next (Successivo).

7. Nella pagina Application Options (Opzioni applicazione), configura le opzioni dell'applicazione.

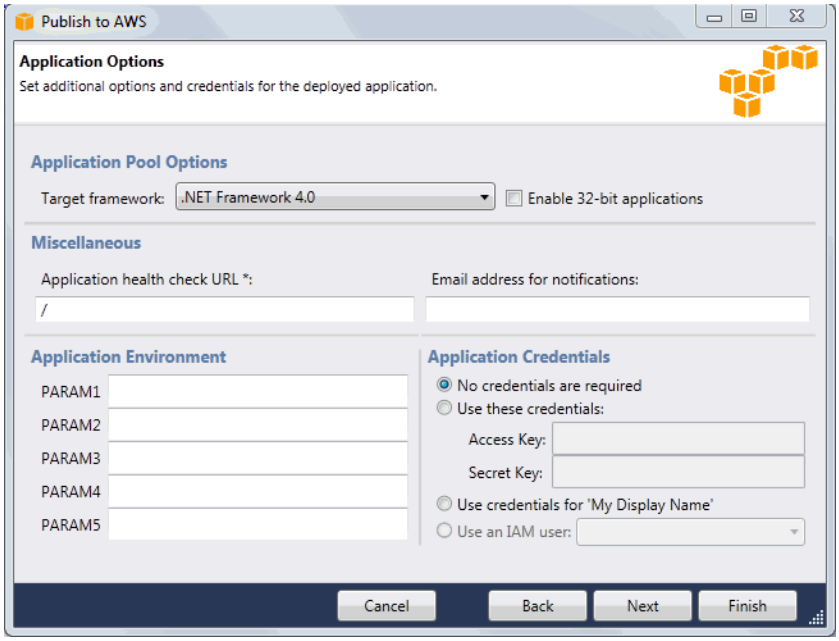
   a. In Target framework, seleziona .NET Framework 4.0.

   b. Elastic Load Balancing utilizza un controllo dello stato per determinare se le EC2 istanze Amazon che eseguono l'applicazione sono integre. Il controllo dello stato determina lo stato dell'istanza sondando un URL specificato secondo l'intervallo definito. Puoi sostituire l'URL di default con un URL di una risorsa esistente nella tua applicazione (ad esempio, `/myapp/index.aspx`) immettendolo nella casella Application health check URL (URL controllo dello stato applicazione). Per ulteriori informazioni sui controlli dello stato per le applicazioni, consulta Controllo dell'integrità.

   c. Digita un indirizzo e-mail per ricevere le notifiche di Amazon Simple Notification Service (Amazon SNS) relative a importanti eventi che riguardano la tua applicazione.

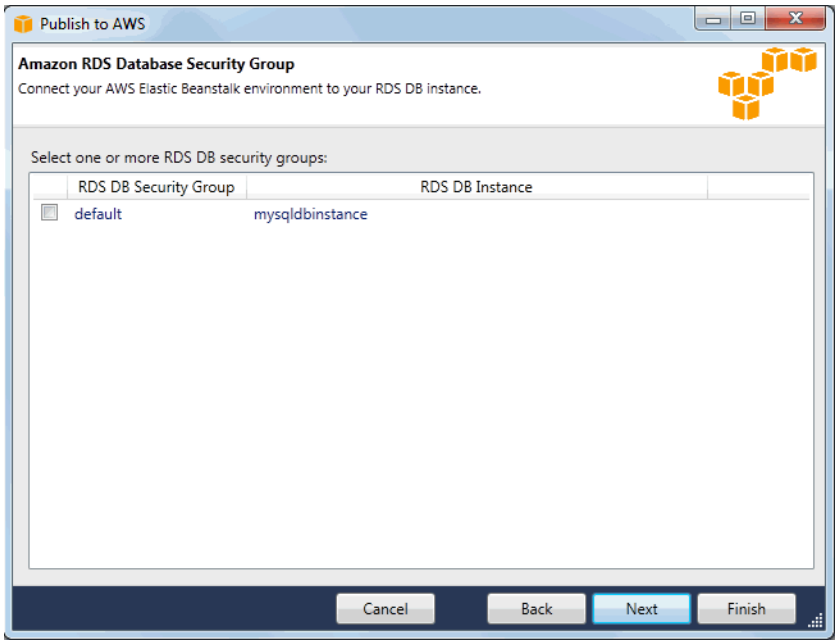
   d. La sezione Ambiente applicativo consente di specificare le variabili di ambiente sulle EC2 istanze Amazon su cui è in esecuzione l'applicazione. Questa impostazione fornisce una maggiore portabilità eliminando la necessità di ricompilare il codice sorgente quando si passa da un ambiente all'altro.

   e. Seleziona l'opzione relativa alle credenziali applicazione che desideri utilizzare per distribuire l'applicazione.

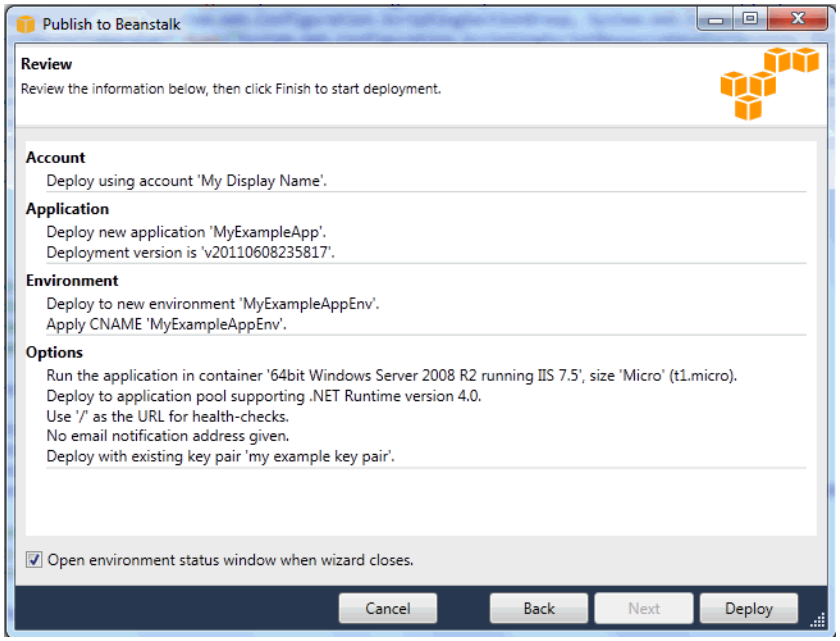


f. Fare clic su Next (Successivo).

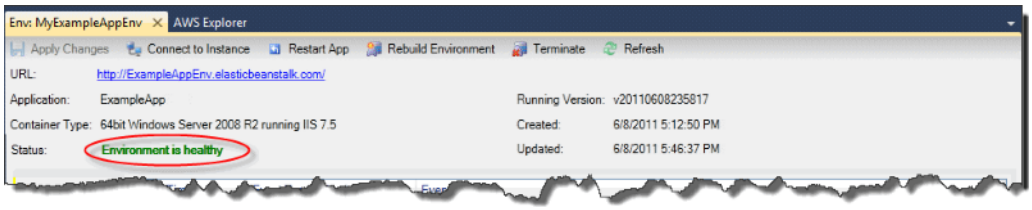
8. Se hai già configurato un database Amazon RDS, si apre la pagina Amazon RDS DB Security Group (Gruppo di sicurezza DB Amazon RDS). Se desideri collegare il tuo ambiente Elastic Beanstalk all'istanza database di Amazon RDS, seleziona uno o più gruppi di sicurezza. In caso contrario, passa alla fase successiva. Quando sei pronto, fai clic su Next (Successivo).



9. Verifica le opzioni di distribuzione. Se non devi apportare alcuna modifica, fai clic su Deploy (Distribuisci).



Il progetto ASP.NET verrà esportato come file di distribuzione Web, caricato in Amazon S3 e registrato come una nuova versione dell'applicazione con Elastic Beanstalk. La funzione di distribuzione di Elastic Beanstalk monitorerà il tuo ambiente esistente finché non sarà disponibile con il codice appena distribuito. Nella scheda env:<environment name> (amb:<nome ambiente>), viene indicato lo stato dell'ambiente.



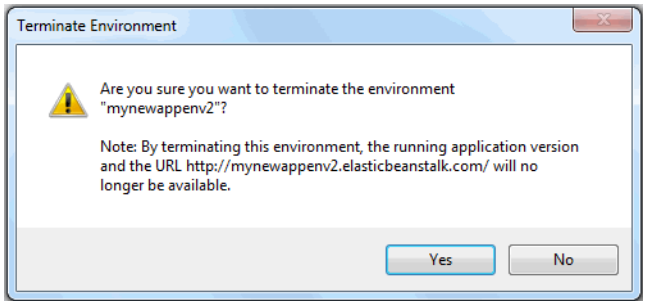
Terminazione di un ambiente

Per evitare di incorrere in addebiti per AWS le risorse non utilizzate, puoi terminare un ambiente in esecuzione utilizzando Toolkit for AWS Visual Studio.

> **Note**
>
> Puoi sempre avviare un nuovo ambiente utilizzando la stessa versione in un secondo momento.

Per terminare un ambiente

1. Espandi il nodo Elastic Beanstalk e il nodo applicazione in AWS Explorer. Fai clic con il pulsante destro del mouse sull'ambiente dell'applicazione e seleziona Terminate Environment (Termina ambiente).
2. Quando richiesto, fai clic su Yes (Sì) per confermare che intendi terminare l'ambiente. Elastic Beanstalk impiegherà alcuni minuti per AWS terminare le risorse in esecuzione nell'ambiente.



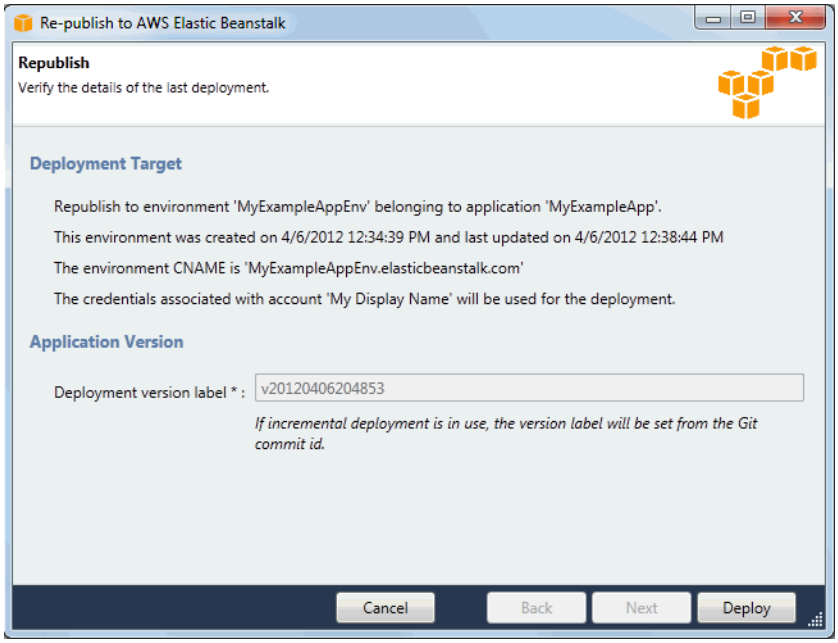
> **Note**
>
> Quando termini l'ambiente, il CNAME associato all'ambiente terminato diventa disponibile e può essere utilizzato da un altro utente.

Distribuzione nell'ambiente

Ora che hai testato la tua applicazione, è facile modificare e ridistribuire l'applicazione e visualizzare i risultati in pochi istanti.

Per modificare e ridistribuire l'applicazione Web ASP.NET

1. In Solution Explorer, fate clic con il pulsante destro del mouse sull'applicazione, quindi fate clic su Ripubblica in ambiente < ***your environment name*** >. Viene visualizzata la procedura guidata Re-publish to (Ripubblica su) AWS Elastic Beanstalk.



2. Consulta i dettagli relativi alla distribuzione, quindi fai clic su Deploy (Distribuisci).

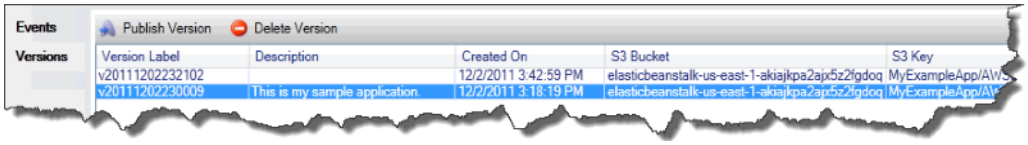
> **Note**
>
> Se desideri modificare le impostazioni, fai clic su Cancel (Annulla) e utilizza la procedura guidata Publish to (Pubblica su) AWS. Per istruzioni, consulta Creazione di un ambiente Elastic Beanstalk.

Il progetto Web aggiornato ASP.NET verrà esportato come file di distribuzione Web con una nuova etichetta versione, caricato in Amazon S3 e registrato come una nuova versione dell'applicazione con Elastic Beanstalk. La funzione di distribuzione di Elastic Beanstalk monitora il tuo ambiente esistente finché non sarà disponibile con il codice appena distribuito. Nella scheda env: < ***environment name*** >, vedrai lo stato del tuo ambiente.

Puoi anche distribuire un'applicazione esistente in un ambiente esistente se, ad esempio, è necessario eseguire il rollback a una versione precedente dell'applicazione.

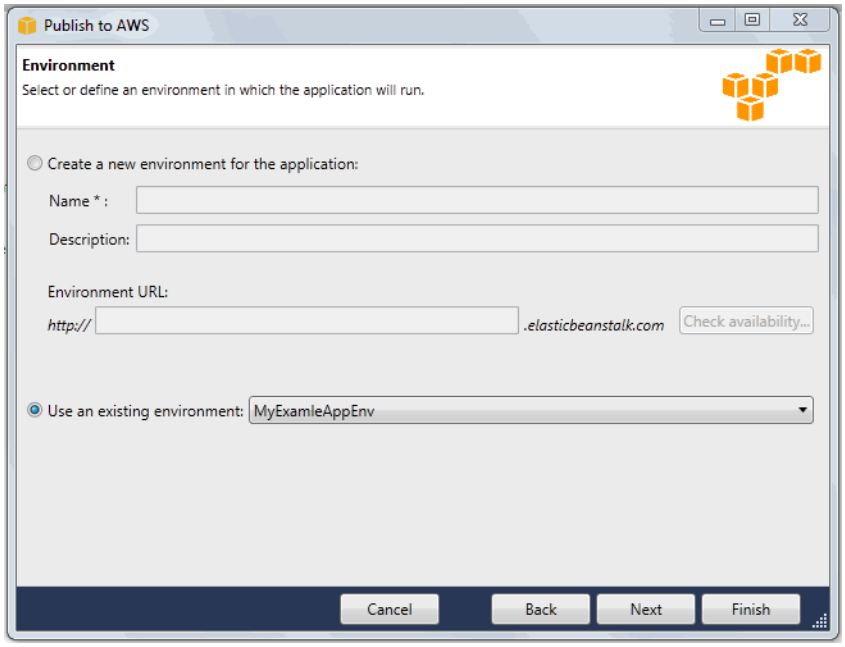
Per distribuire una versione dell'applicazione in un ambiente esistente

1. Fai clic con il pulsante destro del mouse sull'applicazione Elastic Beanstalk espandendo il nodo Elastic Beanstalk in AWS Explorer. Seleziona View Status (Visualizza stato).
2. Nella scheda App: < ***application name*** >, fai clic su Versioni.

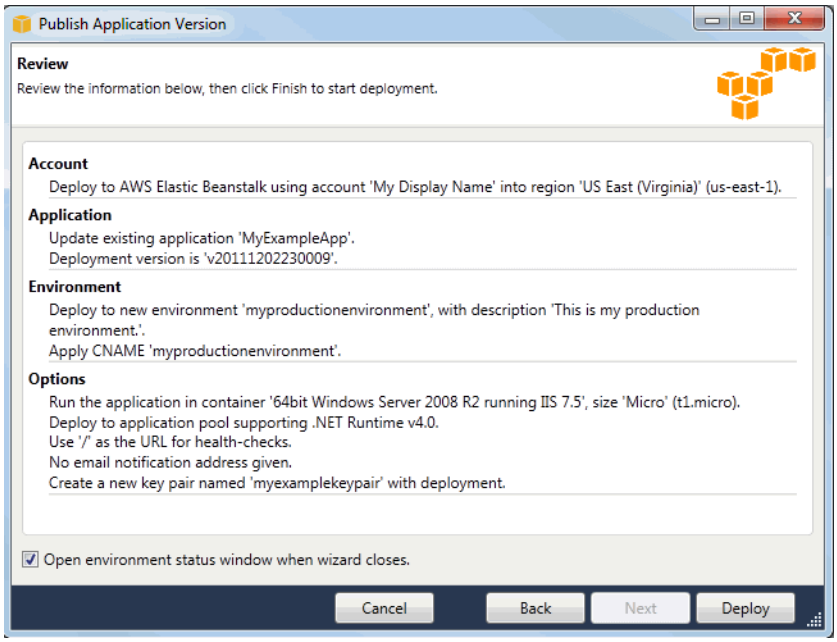


3. Seleziona la versione dell'applicazione che desideri distribuire e fai clic su Publish Version (Pubblica versione).

4. Nella procedura guidata Publish Application Version (Pubblica versione applicazione), fai clic su Next (Successivo).



5. Rivedi le opzioni di distribuzione, quindi fai clic su Deploy (Distribuisci).



Il tuo progetto ASP.NET verrà esportato come un file di distribuzione Web e caricato su Amazon S3. La funzione di distribuzione di Elastic Beanstalk monitorerà il tuo ambiente esistente finché non sarà disponibile con il codice appena distribuito. Nella scheda env: **<** ***environment name*** **>**, vedrai lo stato del tuo ambiente.

Gestione degli ambienti applicativi Elastic Beanstalk

Con AWS Toolkit for Visual Studio e AWS Management Console, puoi modificare il provisioning e la configurazione delle AWS risorse utilizzate dagli ambienti applicativi. Per informazioni su come gestire gli ambienti applicativi utilizzando la Console di AWS gestione, consulta. Gestione degli ambienti Elastic Beanstalk Questa sezione illustra le impostazioni specifiche del servizio che puoi modificare in AWS Toolkit for Visual Studio come parte della configurazione dell'ambiente applicativo.

Modifica delle impostazioni di configurazione dell'ambiente

Quando distribuisci la tua applicazione, Elastic Beanstalk configura una serie di servizi di cloud computing. AWS È possibile controllare la configurazione di questi singoli servizi utilizzando AWS Toolkit for Visual Studio.

Per modificare le impostazioni dell'ambiente di un'applicazione

- Espandi il nodo Elastic Beanstalk e il nodo dell'applicazione. In AWS Explorer, fai clic con il pulsante destro del mouse sull'ambiente Elastic Beanstalk. Seleziona View Status (Visualizza stato).

  È ora possibile configurare le impostazioni per i componenti seguenti:

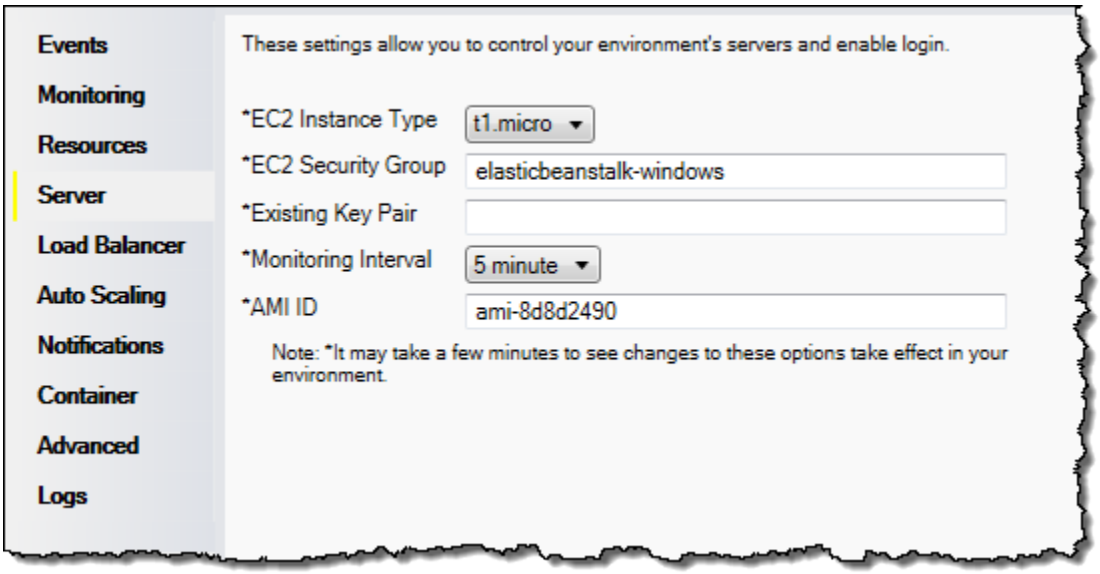
  - Server
  - Bilanciamento del carico
  - Auto Scaling
  - Notifiche
  - Proprietà dell'ambiente

Configurazione delle istanze EC2 del server utilizzando il AWS toolkit per Visual Studio

Amazon Elastic Compute Cloud (Amazon EC2) è un servizio Web che usi per avviare e gestire istanze di server nei data center di Amazon. Puoi utilizzare le istanze EC2 del server Amazon in

qualsiasi momento, per tutto il tempo necessario e per qualsiasi scopo legale. Le istanze sono disponibili in diverse dimensioni e configurazioni. Per ulteriori informazioni, vai su Amazon EC2.

Puoi modificare la configurazione dell'istanza Amazon dell'ambiente Elastic Beanstalk con EC2 la scheda Server all'interno della AWS scheda Ambiente applicativo in Toolkit for Visual Studio.



Tipi di EC2 istanze Amazon

L'opzione Tipo di istanza visualizza i tipi di istanza disponibili per l'applicazione Elastic Beanstalk. Cambia il tipo di istanza per selezionare un server con le caratteristiche (come quantità di memoria e potenza della CPU) più adatte alla tua applicazione. Ad esempio, le applicazioni con operazioni intensive e di lunga durata possono richiedere più memoria o CPU.

Per ulteriori informazioni sui tipi di EC2 istanze Amazon disponibili per la tua applicazione Elastic Beanstalk, consulta Instance Types nella Amazon Elastic Compute Cloud User Guide.

Gruppi EC2 di sicurezza Amazon

Puoi controllare l'accesso alla tua applicazione Elastic Beanstalk utilizzando EC2 un Amazon Security Group. Un gruppo di sicurezza definisce le regole del firewall per le proprie istanze. Tali regole specificano quale traffico di rete in entrata deve essere distribuito sulla tua istanza. Tutto il restante traffico in entrata verrà eliminato. Puoi modificare le regole per un gruppo in qualsiasi momento. Le nuove regole vengono automaticamente applicate per tutte le istanze in esecuzione e le istanze avviate in futuro.

Puoi configurare i gruppi di EC2 sicurezza Amazon utilizzando la Console di AWS gestione o il AWS Toolkit for Visual Studio. Puoi specificare quali gruppi di EC2 sicurezza Amazon controllano l'accesso

alla tua applicazione Elastic Beanstalk inserendo i nomi di uno o più nomi di gruppi di sicurezza EC2 Amazon (delimitati da virgole) nella casella di testo Security Groups. EC2

> **Note**
>
> Verifica che la porta 80 (HTTP) sia accessibile da 0.0.0.0/0 come intervallo CIDR sorgente se desideri abilitare i controlli dello stato per l'applicazione. Per altre informazioni sui controlli dello stato, consulta Controlli dello stato.

Per creare un gruppo di sicurezza utilizzando il toolkit per Visual Studio AWS

1. In Visual Studio, in AWS Explorer, espandi il EC2 nodo Amazon, quindi fai doppio clic su Gruppi di sicurezza.
2. Fai clic su Create Security Group (Crea gruppi di sicurezza), quindi immetti un nome e una descrizione per il gruppo di sicurezza.
3. Fare clic su OK.

Per ulteriori informazioni su Amazon EC2 Security Groups, consulta Using Security Groups nella Amazon Elastic Compute Cloud User Guide.

Coppie di EC2 chiavi Amazon

Puoi accedere in modo sicuro alle EC2 istanze Amazon fornite per la tua applicazione Elastic Beanstalk con una coppia di chiavi Amazon. EC2

> **Important**
>
> Devi creare una coppia di EC2 chiavi Amazon e configurare le istanze Amazon con provisioning di Elastic Beanstalk per utilizzare la EC2 coppia di chiavi Amazon prima di poter accedere alle EC2 istanze Amazon fornite da Elastic Beanstalk. EC2 Puoi creare la tua coppia di chiavi utilizzando la AWS procedura guidata Publish to all'interno di AWS Toolkit for Visual Studio quando distribuisci l'applicazione su Elastic Beanstalk. Per creare ulteriori coppie di chiavi utilizzando il Toolkit, segui le fasi descritte qui di seguito. In alternativa, puoi configurare le tue coppie di EC2 chiavi Amazon utilizzando la console di AWS gestione. Per istruzioni sulla creazione di una coppia di key pair per Amazon EC2, consulta la Amazon Elastic Compute Cloud Getting Started Guide.

La casella di testo Existing Key Pair ti consente di specificare il nome di una coppia di EC2 chiavi Amazon che puoi utilizzare per accedere in modo sicuro alle EC2 istanze Amazon che eseguono la tua applicazione Elastic Beanstalk.

Per specificare il nome di una coppia di EC2 chiavi Amazon

1. Espandi il EC2 nodo Amazon e fai doppio clic su Key Pairs.
2. Fai clic su Create Key Pair (Crea coppia di chiavi) e inserisci il nome della coppia di chiavi.
3. Fare clic su OK.

Per ulteriori informazioni sulle coppie di EC2 chiavi Amazon, consulta Using Amazon EC2 Credentials nella Amazon Elastic Compute Cloud User Guide. Per ulteriori informazioni sulla connessione alle EC2 istanze Amazon, consultaCreazione di elenchi e connessioni alle istanze dei server.

Intervallo di monitoraggio

Per impostazione predefinita, sono abilitate solo le CloudWatch metriche Amazon di base. che restituiscono dati in periodi di cinque minuti. Puoi abilitare CloudWatch metriche più granulari di un minuto selezionando 1 minuto per l'intervallo di monitoraggio nella sezione Server della scheda Configurazione per il tuo ambiente in. AWS Toolkit for Eclipse

> **Note**
>
> I costi CloudWatch del servizio Amazon possono essere applicati per metriche a intervalli di un minuto. CloudWatchPer ulteriori informazioni, consulta Amazon.

ID AMI personalizzata

Puoi sostituire l'AMI predefinita utilizzata per le tue EC2 istanze Amazon con la tua AMI personalizzata inserendo l'identificatore dell'AMI personalizzato nella casella ID AMI personalizzato nella sezione Server della scheda Configurazione per il tuo ambiente in. AWS Toolkit for Eclipse

> **Important**
>
> L'utilizzo di un'AMI personalizzata è un'attività avanzata che devi completare con estrema attenzione. Se hai bisogno di un'AMI personalizzata, ti consigliamo di iniziare con l'AMI predefinita di Elastic Beanstalk e di modificarla. Per essere considerate affidabili, Elastic Beanstalk EC2 si aspetta che le istanze Amazon soddisfino una serie di requisiti, tra cui

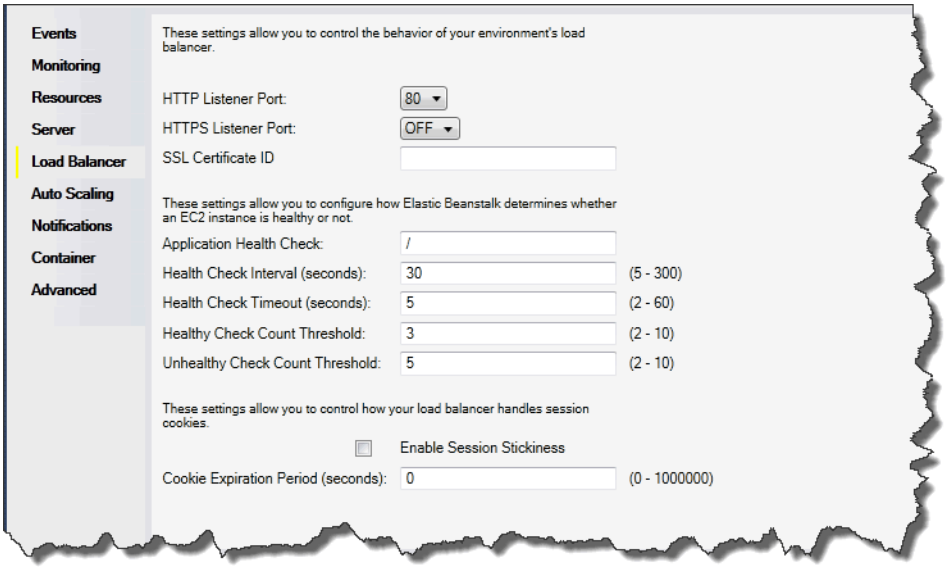
un host manager funzionante. Se questi requisiti non vengono soddisfatti, il tuo ambiente potrebbe non funzionare in modo corretto.

Configurazione di Elastic Load Balancing utilizzando AWS il toolkit per Visual Studio

Elastic Load Balancing è un servizio Web Amazon che aiuta a migliorare la disponibilità e la scalabilità della tua applicazione. Questo servizio semplifica la distribuzione dei carichi di applicazioni tra due o più EC2 istanze Amazon. Elastic Load Balancing consente la disponibilità attraverso la ridondanza e supporta l'aumento di traffico per la tua applicazione.

Elastic Load Balancing distribuisce e bilancia automaticamente il traffico di applicazioni in ingresso tra tutte le istanze in esecuzione. Il servizio, inoltre, consente di aggiungere nuove istanze per aumentare la capacità di un'applicazione.

Elastic Beanstalk effettua automaticamente il provisioning di Elastic Load Balancing quando distribuisci un'applicazione. Puoi modificare la configurazione dell'istanza Amazon dell'ambiente EC2 Elastic Beanstalk con la scheda Load Balancer all'interno della scheda dell'ambiente applicativo in Toolkit for AWS Visual Studio.



Le seguenti sezioni descrivono i parametri di Elastic Load Balancing che puoi configurare per la tua applicazione.

Porte

Il sistema di bilanciamento del carico predisposto per gestire le richieste per l'applicazione Elastic Beanstalk invia le richieste alle istanze EC2 Amazon che eseguono l'applicazione. Il sistema di

bilanciamento del carico fornito può ascoltare le richieste sulle porte HTTP e HTTPS e instradare le richieste verso le EC2 istanze Amazon dell'applicazione. AWS Elastic Beanstalk Per impostazione predefinita, il sistema di bilanciamento del carico gestisce le richieste sulla porta HTTP. Almeno una delle porte (HTTP o HTTPS) deve essere attiva.

> **Important**
>
> Verifica che la porta specificata non sia bloccata; in caso contrario, gli utenti non saranno in grado di connettersi alla tua applicazione Elastic Beanstalk.

Controllo della porta HTTP

Per disattivare la porta HTTP, seleziona OFF per HTTP Listener Port (Porta HTTP del listener). Per attivare la porta HTTP, seleziona una porta HTTP (ad esempio, 80) dall'elenco.

> **Note**
>
> Per accedere al tuo ambiente utilizzando una porta diversa dalla porta 80 predefinita, come la porta 8080, aggiungi un listener al sistema di bilanciamento del carico esistente e configura il nuovo listener per l'ascolto su quella porta.
> Ad esempio, utilizzando il comando AWS CLI for Classic load balancer, digita il seguente comando, sostituendolo *LOAD_BALANCER_NAME* con il nome del tuo load balancer per Elastic Beanstalk.
>
> ```
> aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
> --listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
> InstancePort=80"
> ```
>
> Ad esempio, utilizzando AWS CLI for Application Load Balancers, digita il comando seguente, sostituendolo *LOAD_BALANCER_ARN* con l'ARN del tuo load balancer per Elastic Beanstalk.
>
> ```
> aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
> --port 8080
> ```

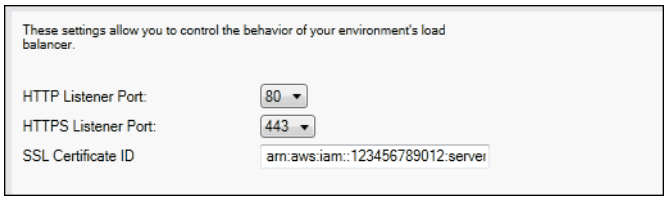
Se desideri che Elastic Beanstalk monitori il tuo ambiente, non rimuovere il listener sulla porta 80.

Controllo della porta HTTPS

Elastic Load Balancing supporta il HTTPS/TLS protocollo per abilitare la crittografia del traffico per le connessioni dei client al sistema di bilanciamento del carico. Le connessioni dal load balancer alle EC2 istanze utilizzano la crittografia in chiaro. Per impostazione predefinita, la porta HTTPS non è attiva.

Per attivare la porta HTTPS

1. Crea un nuovo certificato utilizzando AWS Certificate Manager (ACM) o carica un certificato e una chiave su (IAM). AWS Identity and Access Management Per ulteriori informazioni sulla richiesta di un certificato ACM, consulta Richiesta di un certificato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di certificati nella Guida per l'utente di AWS Certificate Manager . Se ACM non è disponibile nella tua regione, utilizza AWS Identity and Access Management (IAM) per caricare un certificato di terze parti. I servizi ACM e IAM archiviano il certificato e forniscono un Amazon Resource Name (ARN) per il certificato SSL. Per ulteriori informazioni sulla creazione e sul caricamento dei certificati in IAM, consulta l'argomento relativo all'utilizzo dei certificati server nella Guida per l'utente di IAM.
2. Specifica la porta HTTPS selezionando una porta per HTTPS Listener Port (Porta HTTPS del listener).



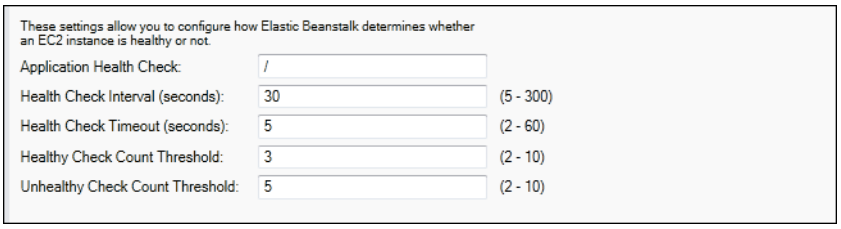
3. In SSL Certificate ID (ID certificato SSL) immetti l'Amazon Resources Name (ARN) del certificato SSL. Ad esempio, **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** o **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**. Usa il certificato SSL creato o caricato nella fase 1.

Per disattivare la porta HTTPS, seleziona OFF per HTTPS Listener Port (Porta HTTPS del listener).

Controlli dello stato

La definizione di controllo dello stato include un URL a cui inviare una query per lo stato dell'istanza. Per impostazione predefinita, Elastic Beanstalk usa TCP: 80 per container non legacy e HTTP: 80 per container legacy. Puoi sostituire l'URL di default con un URL di una risorsa esistente nella tua applicazione (ad esempio, `/myapp/default.aspx`) immettendolo nella casella Application Health Check URL (URL controllo dello stato applicazione). Se ignori l'URL predefinito, Elastic Beanstalk utilizza HTTP per eseguire le query sulla risorsa. Per verificare se stai utilizzando un tipo di container legacy, consulta the section called “Perché alcune versioni di piattaforma sono contrassegnate come legacy?”

Puoi controllare le impostazioni per il controllo dello stato utilizzando la sezione EC2 Instance Health Check del pannello Load Balancing.



La definizione di controllo dello stato include un URL a cui inviare una query per lo stato dell'istanza. Sostituisci l'URL di default con un URL di una risorsa esistente nella tua applicazione (ad esempio, `/myapp/index.jsp`) immettendolo nella casella Application Health Check URL (URL controllo dello stato applicazione).

Di seguito sono descritti i parametri di controllo dello stato che puoi impostare per l'applicazione.

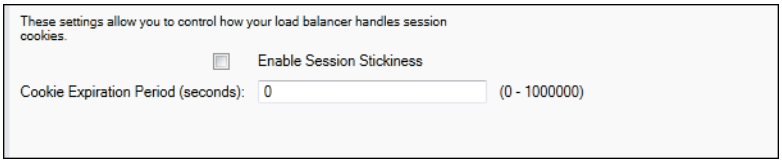
- Per Health Check Interval (secondi), inserisci il numero di secondi di attesa di Elastic Load Balancing tra i controlli di integrità per le istanze Amazon dell'applicazione. EC2
- In Health Check Timeout (Timeout del controllo dello stato (secondi)), specifica il numero di secondi che Elastic Load Balancing deve aspettare per avere una risposta prima di ritenere che l'istanza non risponda.
- In Healthy Check Count Threshold (Soglia di conteggio dei controlli integri) e Unhealthy Check Count Threshold (Soglia di conteggio dei controlli non integri), specifica il numero di sondaggi URL consecutivi riusciti o non riusciti prima che Elastic Load Balancing modifichi lo stato di integrità dell'istanza. Ad esempio, specificando **5** nella casella Unhealthy Check Count Threshold (Soglia di conteggio dei controlli non integri), l'URL restituisce un messaggio di errore o di timeout per cinque volte consecutive prima che Elastic Load Balancing consideri il controllo dello stato come non riuscito.

Sessioni

Per impostazione predefinita, un sistema di bilanciamento del carico prevede l'instradamento di ogni richiesta in modo indipendente all'istanza del server con il carico minore. In confronto, una sticky session associa una sessione utente a una determinata istanza del server, in modo che tutte le richieste provenienti dall'utente durante la sessione siano inviate alla stessa istanza del server.

Elastic Beanstalk utilizza i cookie HTTP generati dal sistema di bilanciamento del carico quando le sticky session sono attivate per un'applicazione. Il sistema di bilanciamento del carico utilizza un cookie speciale generato dal sistema di bilanciamento del carico per monitorare l'istanza dell'applicazione per ogni richiesta. Quando il sistema di bilanciamento del carico riceve una richiesta, verifica innanzitutto se questo cookie è presente nella richiesta. In questo caso, la richiesta viene inviata all'istanza dell'applicazione specificata nel cookie. Se non vengono trovati cookie, il sistema di bilanciamento del carico sceglie un'istanza delle applicazioni basata su algoritmo di bilanciamento del carico esistente. Un cookie è inserito nella risposta per le richieste successive vincolanti dallo stesso utente per quell'istanza dell'applicazione. La configurazione della policy definisce la scadenza di un cookie che stabilisce la durata di validità per ogni cookie.

Puoi utilizzare la sezione Sessions (Sessioni) nella scheda Load Balancer (Sistema di bilanciamento del carico) per specificare se il sistema di bilanciamento del carico della tua applicazione consente l'adesione della sessione.



Per ulteriori informazioni su Elastic Load Balancing, consulta Guida per sviluppatori di Elastic Load Balancing.

Configurazione dell'Auto Scaling utilizzando AWS il toolkit per Visual Studio

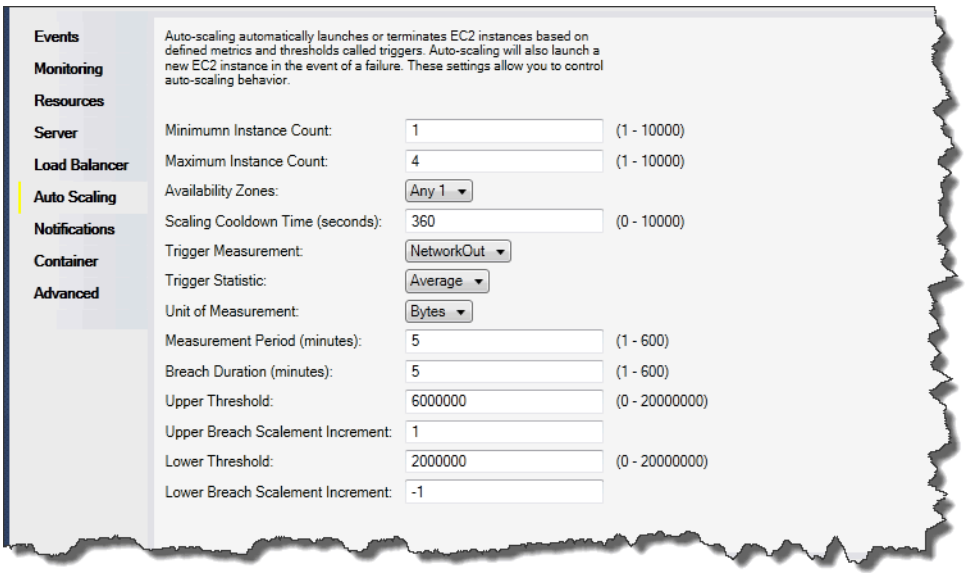
Amazon EC2 Auto Scaling è un servizio web Amazon progettato per avviare o terminare automaticamente le istanze EC2 Amazon in base a trigger definiti dall'utente. Gli utenti possono configurare gruppi Auto Scaling e associare trigger a questi gruppi per scalare automaticamente le risorse di calcolo basate su parametri come l'utilizzo della banda o della CPU. Amazon EC2 Auto Scaling collabora con Amazon CloudWatch per recuperare i parametri per le istanze del server che eseguono la tua applicazione.

Amazon EC2 Auto Scaling ti consente di prendere un gruppo di EC2 istanze Amazon e impostare vari parametri per aumentare o diminuire automaticamente il numero di questo gruppo. Amazon EC2

Auto Scaling può aggiungere o rimuovere EC2 istanze Amazon da quel gruppo per aiutarti a gestire senza problemi le variazioni del traffico della tua applicazione.

Amazon EC2 Auto Scaling monitora anche lo stato di ogni EC2 istanza Amazon che avvia. Se un'istanza si interrompe in modo imprevisto, Amazon EC2 Auto Scaling rileva la chiusura e avvia un'istanza sostitutiva. Questa funzionalità consente di mantenere automaticamente un numero fisso e desiderato di EC2 istanze Amazon.

Elastic Beanstalk fornisce EC2 Amazon Auto Scaling per la tua applicazione. Puoi modificare la configurazione dell'istanza Amazon dell'ambiente EC2 Elastic Beanstalk con la scheda Auto Scaling all'interno della scheda Ambiente applicativo in Toolkit for AWS Visual Studio.

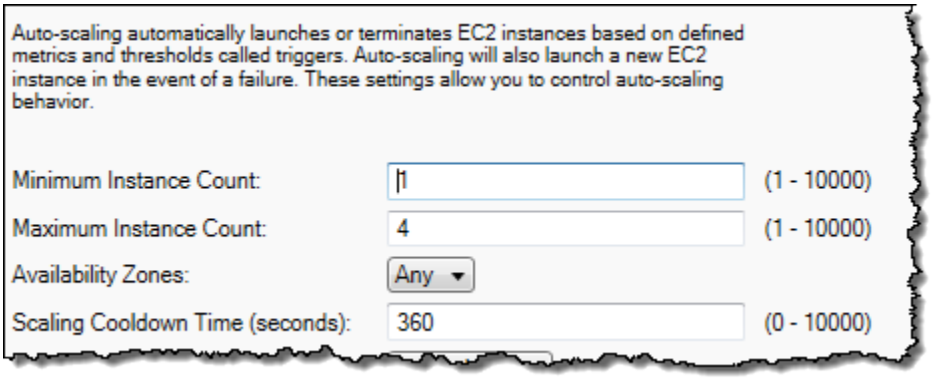


La seziona seguente illustra come configurare i parametri di Auto Scaling per la tua applicazione.

Avvio della configurazione

Puoi modificare la configurazione di avvio per controllare il modo in cui l'applicazione Elastic Beanstalk fornisce le risorse EC2 Amazon Auto Scaling.

Le caselle Minimum Instance Count (Conteggio istanze minimo) e Maximum Instance Count (Conteggio istanze massimo) consentono di specificare la dimensione minima e massima del gruppo Auto Scaling usato dall'applicazione Elastic Beanstalk.



> **Note**
>
> Per mantenere un numero fisso di EC2 istanze Amazon, imposta Numero minimo di istanze e Numero massimo di istanze sullo stesso valore.

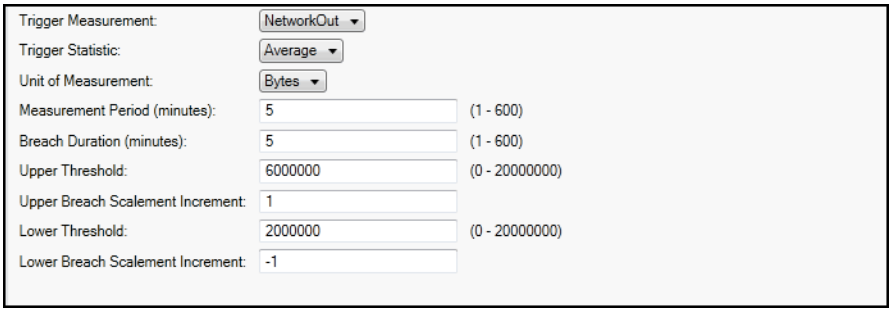
La casella Zone di disponibilità ti consente di specificare il numero di zone di disponibilità in cui desideri che si trovino le tue EC2 istanze Amazon. È importante impostare questo numero se desideri creare applicazioni con tolleranza ai guasti. Se una zona di disponibilità non è raggiungibile, le istanze saranno ancora in esecuzione in altre zone di disponibilità.

> **Note**
>
> Al momento, non è possibile specificare in quale zona di disponibilità si trova l'istanza.

Triggers

Un trigger è un meccanismo di Amazon EC2 Auto Scaling impostato per indicare al sistema quando aumentare (scalare in orizzontale) il numero di istanze e quando diminuire (scalare in avanti) il numero di istanze. Puoi configurare i trigger in modo che si attivino su qualsiasi metrica pubblicata su Amazon CloudWatch, come l'utilizzo della CPU, e determinare se le condizioni specificate sono state soddisfatte. Quando le soglie superiori o inferiori delle condizioni specificate per il parametro sono state violate per il periodo di tempo specificato, il trigger avvia un processo di lunga durata denominato attività di dimensionamento.

Puoi definire un trigger di scalabilità per la tua applicazione Elastic Beanstalk utilizzando Toolkit AWS for Visual Studio.



I trigger di Amazon EC2 Auto Scaling funzionano osservando una CloudWatch metrica Amazon specifica per un'istanza. I trigger includono l'utilizzo della CPU, il traffico di rete e l'attività del disco. Utilizza l'impostazione Trigger Measurement (Misurazione del trigger) per selezionare un parametro per il trigger.

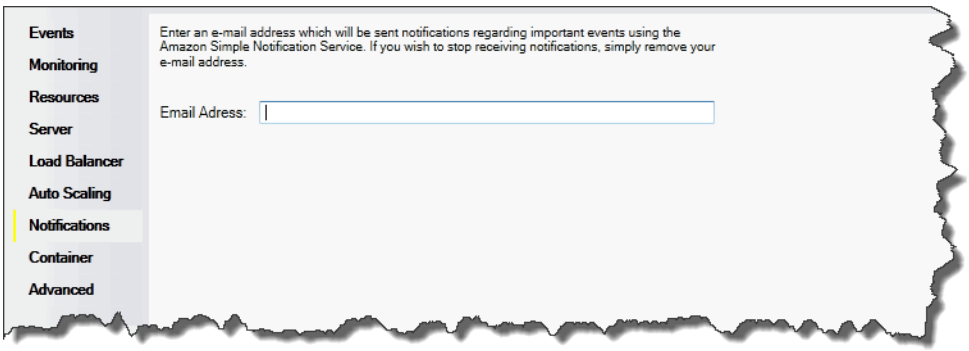
L'elenco seguente descrive i parametri di attivazione che puoi configurare utilizzando la AWS Console di gestione.

- Puoi specificare le statistiche che il trigger deve utilizzare. Puoi selezionare Minimum (Minimo), Maximum (Massimo), Sum (Somma) o Average (Media) per Trigger Statistic (Statistica trigger).
- In Unit of Measurement (Unità di misura), specifica l'unità per la misurazione del trigger.
- Il valore nella casella Periodo di misurazione specifica la frequenza con cui Amazon CloudWatch misura le metriche relative al trigger. Breach Duration (Durata della violazione) specifica la quantità di tempo in cui un parametro può essere oltre il limite definito (specificato in Upper Treshold (Soglia superiore) e in Lower Threshold (Soglia inferiore)) prima che il trigger si attivi.
- Per Upper Breach Scale Increment e Lower Breach Scale Increment, specifica quante EC2 istanze Amazon aggiungere o rimuovere durante l'esecuzione di un'attività di scalabilità.

Per ulteriori informazioni su Amazon EC2 Auto Scaling, consulta la sezione Amazon Auto EC2 Scaling nella documentazione di Amazon Elastic Compute Cloud.

Configurazione delle notifiche utilizzando AWS il toolkit per Visual Studio

Elastic Beanstalk utilizza Amazon Simple Notification Service (Amazon SNS) per la notifica di eventi importanti che interessano l'applicazione. Per abilitare le notifiche di Amazon SNS, è sufficiente inserire il tuo indirizzo e-mail nella casella Email Address (Indirizzo e-mail). Per disabilitare queste notifiche, elimina il tuo indirizzo e-mail dalla casella di testo.



Configurazione dei contenitori.NET utilizzando il AWS toolkit per Visual Studio

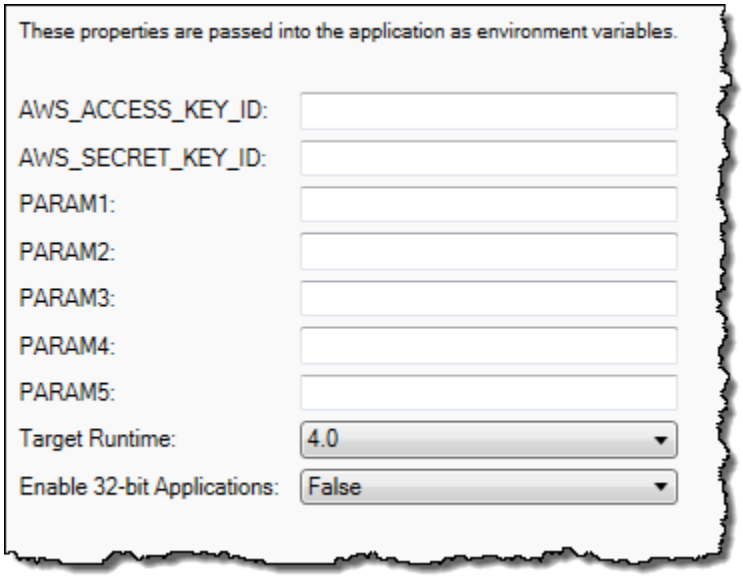
Il pannello Container/.NET Options consente di ottimizzare il comportamento delle EC2 istanze Amazon e abilitare o disabilitare la rotazione dei log di Amazon S3. Puoi usare AWS Toolkit for Visual Studio per configurare le informazioni sul contenitore.

> **Note**
>
> È possibile modificare le impostazioni di configurazione senza tempi di inattività scambiando il CNAME per i tuoi ambienti. Per ulteriori informazioni, consulta Distribuzioni blu/verde con Elastic Beanstalk.

Se desideri, è possibile estendere il numero di parametri. Per informazioni sull'estensione dei parametri consulta Impostazioni opzioni.

Per accedere al pannello delle opzioni Container/.NET dell'applicazione Elastic Beanstalk

1. In AWS Toolkit for Visual Studio, espandi il nodo Elastic Beanstalk e il nodo dell'applicazione.
2. In AWS Explorer, fai doppio clic sull'ambiente Elastic Beanstalk.
3. Nella parte inferiore del riquadro Overview (Panoramica), fai clic sulla scheda Configuration (Configurazione).
4. In Container, puoi configurare le opzioni relative al container.

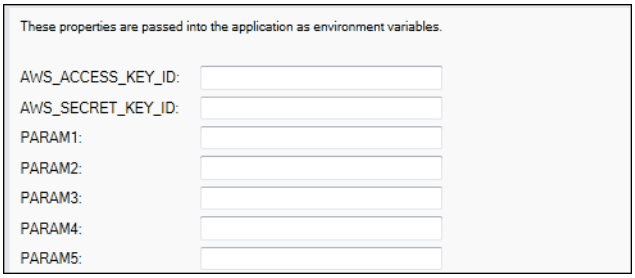


Opzioni del container .NET

È possibile scegliere la versione di .NET Framework per la tua applicazione. Scegli 2.0 o 4.0 per Target runtime (Runtime target). Seleziona Enable 32-bit Applications (Abilita le applicazioni a 32 bit) per abilitare le applicazioni a 32 bit.

Impostazioni applicazioni

La sezione Application Settings (Impostazioni applicazione) consente di specificare variabili di ambiente che puoi leggere dal codice dell'applicazione.



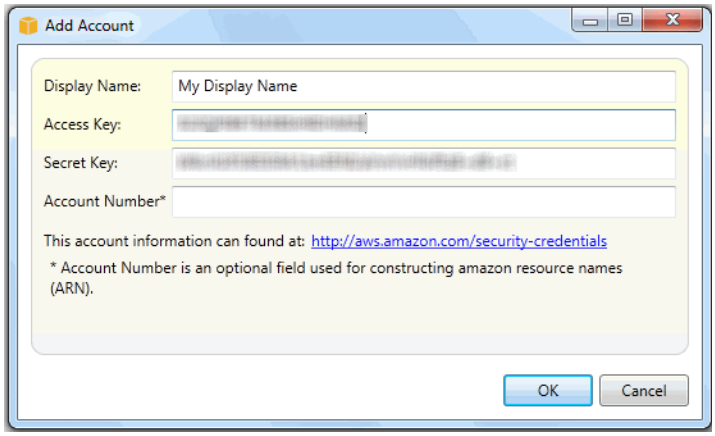
Gestione degli account

Se desideri configurare AWS account diversi per eseguire attività diverse, come test, gestione temporanea e produzione, puoi aggiungere, modificare ed eliminare account utilizzando AWS Toolkit for Visual Studio.

Gestione di più account

1. In Visual Studio, nel menu View (Visualizza), fai clic su AWS Explorer.
2. Accanto all'elenco Account, fai clic sul pulsante Add account (Aggiungi account).

Si apre la finestra di dialogo Add account (Aggiungi account).



3. Inserisci le informazioni richieste.
4. Le informazioni sul tuo account ora appaiono nella scheda AWS Explorer. Quando pubblichi su Elastic Beanstalk, è possibile selezionare l'account da utilizzare.

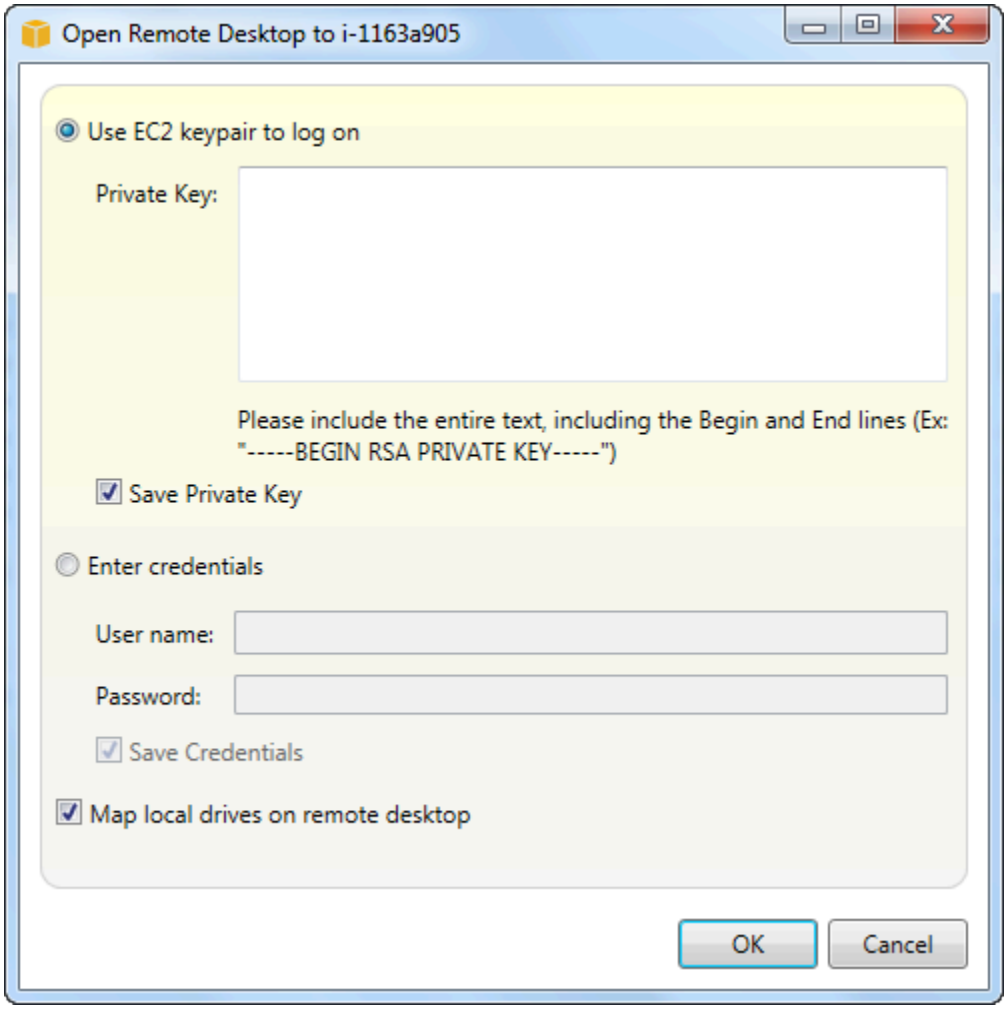
Creazione di elenchi e connessioni alle istanze dei server

Puoi visualizzare un elenco di EC2 istanze Amazon che eseguono il tuo ambiente applicativo Elastic Beanstalk tramite Toolkit AWS for Visual Studio o dalla console di gestione. AWS Puoi connetterti a queste istanze utilizzando la connessione al desktop remoto. Per informazioni su come elencare e connettersi alle istanze del server utilizzando la console di gestione, consulta. AWS Creazione di elenchi e connessioni alle istanze dei server La sezione seguente illustra come visualizzare e connettere l'utente alle istanze del server utilizzando AWS Toolkit for Visual Studio.

Per visualizzare e connettersi alle EC2 istanze Amazon per un ambiente

1. In Visual Studio, in AWS Explorer, espandi il EC2 nodo Amazon e fai doppio clic su Istanze.

2. Fai clic con il pulsante destro del mouse sull'ID dell' EC2 istanza Amazon in esecuzione nel sistema di bilanciamento del carico dell'applicazione nella colonna Istanza e seleziona Open Remote Desktop dal menu contestuale.



3. Seleziona Usa EC2 coppia di chiavi per accedere e incolla il contenuto del file di chiave privata che hai usato per distribuire l'applicazione nella casella Chiave privata. In alternativa, immetti il tuo nome utente e la password nelle caselle di testo User name (Nome utente) e Password.

> **Note**
>
> Se la coppia di chiavi viene archiviata all'interno del Toolkit, la casella di testo non viene visualizzata.

4. Fare clic su OK.

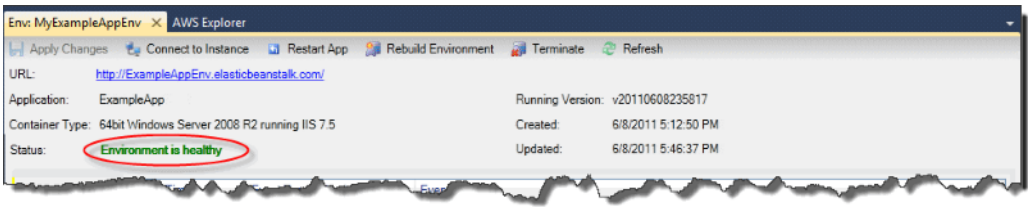
Monitoraggio dello stato dell'applicazione

Quando esegui un sito Web di produzione, è importante sapere che la tua applicazione è disponibile e risponde alle richieste. Per facilitare il monitoraggio della reattività dell'applicazione, Elastic Beanstalk fornisce caratteristiche che consentono di controllare le statistiche sull'applicazione e creare avvisi che vengono attivati al superamento delle soglie.

Per informazioni sul monitoraggio dello stato fornito da Elastic Beanstalk, consulta Report dello stato di base.

È possibile accedere alle informazioni operative sull'applicazione utilizzando il AWS Toolkit for Visual Studio o AWS la Console di gestione.

Il kit di strumenti mostra lo stato dell'ambiente e delle applicazioni nel campo Status (Stato).



Per monitorare lo stato dell'applicazione

1. In AWS Toolkit for Visual Studio, AWS in Explorer, espandi il nodo Elastic Beanstalk, quindi espandi il nodo dell'applicazione.
2. Fai clic con il pulsante destro del mouse sull'ambiente Elastic Beanstalk, quindi fai clic su View Status (Visualizza stato).
3. Nella scheda dell'ambiente dell'applicazione, fai clic su Monitoring (Monitoraggio).

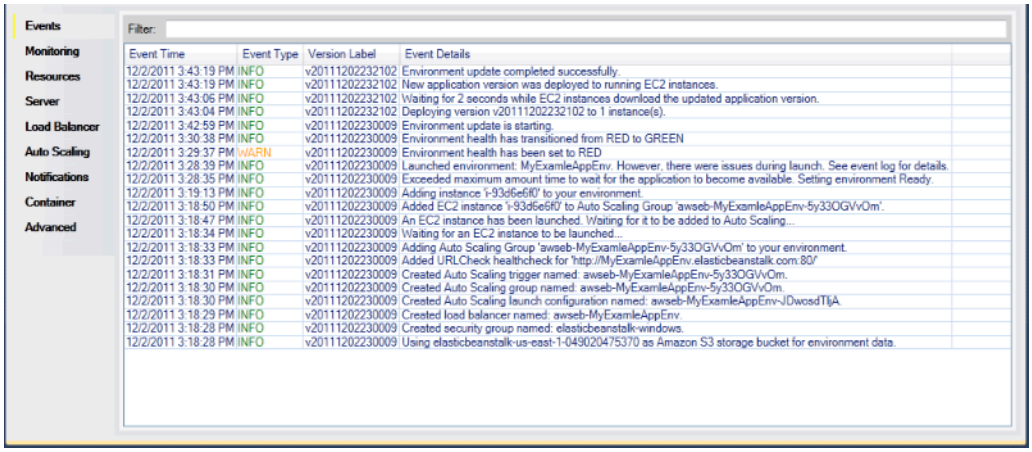
   Il pannello Monitoring (Monitoraggio) include un set di grafici che mostrano l'utilizzo delle risorse per un ambiente applicativo specifico.

> **Note**
>
> Per impostazione predefinita, l'intervallo di tempo è impostato sull'ultima ora. Per modificare questa impostazione, nell'elenco Time Range (Intervallo tempo) fai clic su un altro intervallo temporale.

È possibile utilizzare AWS Toolkit for Visual Studio o AWS la Console di gestione per visualizzare gli eventi associati all'applicazione.

Per visualizzare gli eventi dell'applicazione

1. Nel AWS Toolkit for Visual Studio, AWS in Explorer, espandi il nodo Elastic Beanstalk e il nodo dell'applicazione.
2. Fai clic con il pulsante destro del mouse sull'ambiente Elastic Beanstalk in AWS Explorer, quindi fai clic su View Status (Visualizza stato).
3. Nella scheda dell'ambiente dell'applicazione, fai clic su Events (Eventi).



Distribuzione di applicazioni Elastic Beanstalk in.NET utilizzando strumenti di distribuzione AWS

AWS offre due strumenti da riga di comando per distribuire applicazioni.NET Core e ASP.NET Core su Elastic Beanstalk:

- AWS Strumento di distribuzione.NET: un moderno strumento a riga di comando che semplifica la distribuzione di applicazioni.NET nei AWS servizi, incluso Elastic Beanstalk. Questo strumento offre un'esperienza interattiva ed è in grado di rilevare automaticamente il servizio migliore AWS per l'applicazione. Per ulteriori informazioni, consulta lo strumento di AWS distribuzione.NET.
- AWS Estensioni per .NET CLI: un insieme di estensioni CLI.NET che forniscono comandi per la distribuzione di applicazioni.NET Core su Elastic Beanstalk e altri servizi. AWS Queste estensioni si integrano con il flusso di lavoro standard dell'interfaccia CLI.NET. Per ulteriori informazioni, consulta AWS Extensions for .NET CLI.

Migrazione dell'applicazione.NET in locale a Elastic Beanstalk

AWS Elastic Beanstalk fornisce un percorso di migrazione semplificato per le applicazioni Windows in esecuzione su Internet Information Services (IIS) tramite l'Elastic Beanstalk Command Line Interface (EB CLI). Il eb migrate comando rileva automaticamente i siti, le applicazioni e le directory virtuali IIS, ne preserva le configurazioni e li distribuisce nel cloud. AWS

Questa funzionalità di migrazione integrata offre un approccio più semplice che riduce la complessità e il tempo tipicamente associati alle migrazioni cloud. Il processo di migrazione aiuta a mantenere la funzionalità dell'applicazione e l'integrità della configurazione durante la transizione a. AWS

Per istruzioni complete e dettagliate sulla migrazione delle applicazioni IIS a AWS Elastic Beanstalk, consulta il Migrazione delle applicazioni IIS capitolo di questa guida.

Consigli per i componenti ritirati di Windows Server su Elastic Beanstalk

Questo argomento fornisce consigli se le applicazioni sono attualmente in esecuzione sui rami della piattaforma Windows Server 2012 R2 ritirati. Riguarda anche il supporto obsoleto per le versioni dei protocolli TLS 1.0 e 1.1 sugli endpoint delle nostre API di AWS servizio e sui rami della piattaforma interessati.

Ramificazioni della piattaforma Windows Server 2012 R2 ritirate

Elastic Beanstalk ha ritirato le filiali della piattaforma Windows Server 2012 R2 il 4 dicembre 2023 e AMIs ha reso private quelle associate a tali piattaforme il 10 aprile 2024. Questa azione impedisce l'avvio di istanze negli ambienti Windows Server 2012 che utilizzano l'AMI Beanstalk predefinita.

Se disponi di ambienti in esecuzione su rami della piattaforma Windows ritirati, ti consigliamo di migrarli su una delle seguenti piattaforme Windows Server, che sono attuali e completamente supportate:

- Windows Server 2022 con IIS 10.0 versione 2.x
- Windows Server 2019 con IIS 10.0 versione 2.x

Per considerazioni complete sulla migrazione, consulta Migrazione da versioni principali della piattaforma Windows Server.

Per informazioni sulla dichiarazione della piattaforma come obsolete, consulta Policy di supporto per la piattaforma Elastic Beanstalk.

> **Note**
>
> Se non riesci a migrare verso queste piattaforme completamente supportate, ti consigliamo di utilizzare un'immagine personalizzata AMIs creata con Windows Server 2012 R2 o Windows Server 2012 R2 Core AMIs come immagine di base, se non l'hai già fatto. Per istruzioni dettagliate, vedi Preservare l'accesso a una Amazon Machine Image (AMI) per una piattaforma ritirata. Rivolgiti al AWS Support Center se hai bisogno di accedere temporaneamente a un'AMI mentre esegui una di queste fasi di migrazione.

Compatibilità con TLS 1.2

A partire dal 31 dicembre 2023, AWS ha iniziato a applicare completamente TLS 1.2 su tutti gli endpoint AWS API. Questa azione ha rimosso la possibilità di utilizzare le versioni TLS 1.0 e 1.1 con tutti. AWS APIs Queste informazioni sono state originariamente comunicate il 28 giugno 2022. Per evitare il rischio di un impatto sulla disponibilità, aggiorna tutti gli ambienti che eseguono le versioni della piattaforma qui identificate a una versione più recente il prima possibile, se non l'hai già fatto.

Impatto potenziale

Le versioni delle piattaforme Elastic Beanstalk che eseguono TLS v1.1 o precedenti ne risentono. Questa modifica influisce sulle azioni ambientali che includono, a titolo esemplificativo ma non esaustivo: distribuzioni di configurazione, distribuzione di applicazioni, scalabilità automatica, lancio di un nuovo ambiente, rotazione dei log, report sullo stato migliorati e pubblicazione dei log delle applicazioni nel bucket Amazon S3 associato alle tue applicazioni.

Versioni della piattaforma Windows interessate

Si consiglia ai clienti con ambienti Elastic Beanstalk sulle versioni della piattaforma riportate di seguito di aggiornare ciascuno degli ambienti corrispondenti alla versione della piattaforma Windows 2.8.3 o successiva, rilasciata il 18 febbraio 2022.

- Windows Server 2019: versione della piattaforma 2.8.2 o versioni precedenti

Si consiglia ai clienti con ambienti Elastic Beanstalk sulle versioni della piattaforma riportate di seguito di aggiornare ciascuno degli ambienti corrispondenti alla versione della piattaforma Windows 2.10.7 o successiva, rilasciata il 28 dicembre 2022.

- Windows Server 2016: versione della piattaforma 2.10.6 o versioni precedenti
- Windows Server 2012: tutte le versioni della piattaforma; questa piattaforma è stata ritirata il 4 dicembre 2023
- Windows Server 2008: tutte le versioni della piattaforma; questa piattaforma è stata ritirata il 28 ottobre 2019

Per un elenco delle versioni della piattaforma di Windows Server più recenti e supportate, consulta la pagina Piattaforme supportate nella guida Piattaforme AWS Elastic Beanstalk .

Per dettagli e procedure consigliate sull'aggiornamento dell'ambiente, consulta Aggiornamento della versione della piattaforma dell'ambiente Elastic Beanstalk.

Distribuzione di applicazioni.NET core (Linux) con Elastic Beanstalk

> **Dai un'occhiata a .NET on AWS Developer Center**
>
> Hai visitato il nostro.Net Developer Center? È il nostro sportello unico per tutto ciò che riguarda .NET AWS.
> Per ulteriori informazioni, consulta .NET on AWS Developer Center.

Questo capitolo fornisce istruzioni per configurare e distribuire l'applicazione web.NET core (Linux) su. AWS Elastic Beanstalk Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni web.NET core (Linux) utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, puoi continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure puoi utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Segui i passaggi indicati nelle step-by-step istruzioni QuickStart per.NET Core su Linux per creare e distribuire un'applicazione Web ASP.NET Core Hello World con l'EB CLI.

Argomenti

- QuickStart: distribuisci un'applicazione.NET Core su Linux su Elastic Beanstalk
- Configurazione del core.NET nell'ambiente di sviluppo Linux per Elastic Beanstalk
- Utilizzo del core Elastic Beanstalk.NET su piattaforma Linux
- Il AWS Toolkit for Visual Studio - Lavorare con.Net Core su Elastic Beanstalk
- Migrazione da.NET su piattaforma Windows Server a .NET Core su piattaforma Linux su Elastic Beanstalk

QuickStart: distribuisci un'applicazione.NET Core su Linux su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione.NET Core su Linux e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Passaggio 1: Creare un'applicazione.NET Core su Linux
- Passaggio 2: Esegui l'applicazione localmente
- Fase 3: Distribuire l'applicazione.NET Core su Linux con l'EB CLI
- Fase 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

.NET Core su Linux

Se non hai installato il.NET SDK sul tuo computer locale, puoi installarlo seguendo il link Download.NET sul sito Web della documentazione .NET.

Verifica l'installazione di.NET SDK eseguendo il comando seguente.

```
~$ dotnet --info
```

Passaggio 1: Creare un'applicazione.NET Core su Linux

Crea una directory di progetto.

```
~$ mkdir eb-dotnetcore
~$ cd eb-dotnetcore
```

Quindi, crea un'applicazione Hello World di esempio eseguendo i seguenti comandi.

```
~/eb-dotnetcore$ dotnet new web --name HelloElasticBeanstalk
~/eb-dotnetcore$ cd HelloElasticBeanstalk
```

Passaggio 2: Esegui l'applicazione localmente

Esegui il comando seguente per eseguire l'applicazione localmente.

```
~/eb-dotnetcore/HelloElasticBeasntalk$ dotnet run
```

L'output dovrebbe avere un aspetto simile al testo seguente.

```
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7294
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5052
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
```

> **Note**
>
> Il `dotnet` comando seleziona una porta a caso quando si esegue l'applicazione localmente. In questo esempio la porta è 5052. Quando distribuisci l'applicazione nell'ambiente Elastic Beanstalk, l'applicazione verrà eseguita sulla porta 5000.

Inserisci l'indirizzo URL `http://localhost:`*`port`* nel tuo browser web. Per questo esempio specifico, il comando è`http://localhost:5052`. Il browser web dovrebbe visualizzare «Hello World!».

Fase 3: Distribuire l'applicazione.NET Core su Linux con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire l'applicazione.NET Core su Linux

1. Compila e pubblica l'applicazione in una cartella per la distribuzione nell'ambiente Elastic Beanstalk che stai per creare.

   ```
   ~$ cd eb-dotnetcore/HelloElasticBeanstalk
   ```

```
~/eb-dotnetcore/HelloElasticBeanstalk$ dotnet publish -o site
```

2. Vai alla directory del sito in cui hai appena pubblicato l'app.

```
~/eb-dotnetcore/HelloElasticBeanstalk$ cd site
```

3. Inizializza il repository della CLI EB con il comando eb init.

   Tieni presente i seguenti dettagli relativi alla versione del ramo della piattaforma specificata nel comando:

   - Sostituire *x.y.z* nel comando seguente con l'ultima versione del ramo della piattaforma .NET 6 su AL2023.
   - Per individuare la versione più recente del ramo della piattaforma, consulta .NET Core per piattaforme supportate da Linux nella guida AWS Elastic Beanstalk Piattaforme.
   - Un esempio di nome di stack di soluzioni che include il numero di versione è`64bit-amazon-linux-2023-v`**`3.1.1`**`-running-.net-6`. In questo esempio la versione del ramo è 3.1.1.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb init -p 64bit-amazon-linux-2023-
vx.y.z-running-.net-6 dotnetcore-tutorial --region us-east-2
Application dotnetcore-tutorial has been created.
```

   Questo comando crea un'applicazione denominata `dotnetcore-tutorial` e configura l'archivio locale per creare ambienti con la versione della piattaforma .NET Core su Linux specificata nel comando.

4. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

   Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

5. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

   in

   ```
   ~eb-dotnetcore/HelloElasticBeanstalk/site$ eb create dotnet-tutorial
   ```

   Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb open
```

Complimenti! Hai distribuito un'applicazione.NET Core su Linux con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 5: rimozione

Puoi chiudere il tuo ambiente quando finisci di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx

come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.

- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire localmente .NET Core su applicazioni Linux, vediConfigurazione del core.NET nell'ambiente di sviluppo Linux per Elastic Beanstalk.

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Configurazione del core.NET nell'ambiente di sviluppo Linux per Elastic Beanstalk

Questo argomento fornisce istruzioni per configurare un .NET core in un ambiente di sviluppo Linux per testare l'applicazione localmente prima di distribuirla AWS Elastic Beanstalk. Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Sections

- Installazione dell'SDK NET Core
- Installazione di un IDE
- Installazione di AWS Toolkit for Visual Studio

Installazione dell'SDK NET Core

È possibile utilizzare l'SDK .NET Core per sviluppare applicazioni in esecuzione su Linux.

Consulta la pagina di download di .NET per scaricare e installare l'SDK .NET Core.

Installazione di un IDE

Gli ambienti di sviluppo integrati (IDEs) forniscono una serie di funzionalità che facilitano lo sviluppo di applicazioni. Se non hai utilizzato un IDE per lo sviluppo .NET, prova Visual Studio Community per iniziare.

Visita la pagina di Visual Studio Community per scaricare e installare Visual Studio Community.

Installazione di AWS Toolkit for Visual Studio

AWS Toolkit for Visual StudioSi tratta di un plug-in open source per l'IDE di Visual Studio che semplifica lo sviluppo, il debug e la distribuzione di applicazioni.NET utilizzando. AWS Visita la home page di Toolkit for Visual Studio per le istruzioni di installazione.

Utilizzo del core Elastic Beanstalk.NET su piattaforma Linux

Questo argomento descrive come configurare, creare ed eseguire il core.NET su applicazioni Linux su Elastic Beanstalk.

AWS Elastic Beanstalk supporta una serie di rami di piattaforma per diverse versioni del framework.NET Core eseguite sul sistema operativo Linux. Per un elenco completo, consulta .NET core su Linux nella sezione AWS Elastic Beanstalk Piattaforme.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Considerazioni su .NET Core sulla piattaforma Linux

Server proxy

Elastic Beanstalk .NET Core su piattaforma Linux include un proxy inverso che inoltra le richieste alla tua applicazione. Per impostazione predefinita, Elastic Beanstalk utilizza NGINX come server proxy. Puoi scegliere di non usare alcun server proxy e configurare Kestrel come server web. Kestrel è incluso per impostazione predefinita nei modelli di progetto ASP.NET Core.

Struttura di un’applicazione

Puoi pubblicare applicazioni dipendenti da runtime che utilizzano il runtime .NET Core fornito da Elastic Beanstalk. È inoltre possibile pubblicare applicazioni autonome che includono il runtime .NET Core e le dipendenze dell'applicazione nel bundle di origine. Per ulteriori informazioni, consulta the section called “Raggruppamento delle applicazioni”.

Configurazione della piattaforma

Per configurare i processi in esecuzione nelle istanze del server nell'ambiente, includi un oggetto Procfile opzionale nel bundle di origine. Un `Procfile` è necessario se si dispone di più di un'applicazione nel bundle di origine.

Si consiglia di fornire sempre un `Procfile` nel bundle di origine insieme all'applicazione. In questo modo puoi controllare con precisione quali processi vengono eseguiti da Elastic Beanstalk per la tua applicazione.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci

l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Configurazione dell'ambiente .NET Core su Linux

Le impostazioni di NET Core su piattaforma Linux consentono di ottimizzare il comportamento delle istanze Amazon EC2. Puoi modificare la configurazione dell'istanza Amazon EC2 dell'ambiente Elastic Beanstalk utilizzando la console Elastic Beanstalk.

Utilizza la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3 e configurare variabili che l'applicazione è in grado di leggere dall'ambiente.

Per configurare l'ambiente .NET Core su Linux utilizzando la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Enable log file rotation to Amazon S3 (Abilita rotazione file di log in Amazon S3): specifica se i file di log per le istanze Amazon EC2 dell'applicazione debbano essere copiati nel bucket Amazon S3 associato all'applicazione.

Proprietà dell'ambiente

La sezione Environment Properties (Proprietà di ambiente) consente di specificare le impostazioni di configurazione dell'ambiente nelle istanze Amazon EC2 che eseguono l'applicazione. Le proprietà di ambiente sono passate come coppie chiave-valore all'applicazione.

All'interno dell'ambiente .NET Core su Linux in esecuzione in Elastic Beanstalk, le variabili di ambiente sono accessibili tramite la funzione `Environment.GetEnvironmentVariable("variable-name")`. Ad esempio, è possibile leggere una proprietà denominata `API_ENDPOINT` su una variabile con il codice seguente.

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

Per ulteriori informazioni, consultare Variabili di ambiente e altre impostazioni software.

NET Core nello spazio dei nomi di configurazione Linux

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

Oltre alle opzioni supportate per tutti gli ambienti Elastic Beanstalk, la piattaforma .NET Core su Linux supporta quelle incluse negli spazi dei nomi seguenti:

- `aws:elasticbeanstalk:environment:proxy`— Scegli di usare NGINX o nessun server proxy. I valori validi sono `nginx` e `none`.

L'esempio di file di configurazione seguente mostra l'utilizzo delle opzioni di configurazione specifiche di .NET Core su Linux.

Example.ebextensions/proxy-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la

console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Raggruppamento di applicazioni per.NET Core sulla piattaforma Linux Elastic Beanstalk

Puoi eseguire applicazioni .NET Core dipendenti dal runtime e autonome su AWS Elastic Beanstalk.

Un'applicazione dipendente dal runtime utilizza un runtime .NET Core fornito da Elastic Beanstalk per eseguire l'applicazione. Elastic Beanstalk utilizza il file `runtimeconfig.json` nel bundle di origine per determinare il runtime da utilizzare per l'applicazione. Elastic Beanstalk sceglie l'ultimo runtime compatibile disponibile per la tua applicazione.

Un'applicazione autonoma include il runtime .NET Core, l'applicazione e le relative dipendenze. Per utilizzare una versione del runtime di NET Core che Elastic Beanstalk non include nelle piattaforme, fornisci un'applicazione autonoma.

Esempi

È possibile compilare sia applicazioni autonome che dipendenti dal runtime con il comando `dotnet publish`. Per ulteriori informazioni sulla pubblicazione di app .NET Core, vedere Panoramica sulla pubblicazione di applicazioni NET Core nella documentazione di .NET Core.

La struttura di file di esempio seguente definisce una singola applicazione che utilizza un runtime .NET Core fornito da Elastic Beanstalk.

```
### appsettings.Development.json
### appsettings.json
### dotnetcoreapp.deps.json
### dotnetcoreapp.dll
### dotnetcoreapp.pdb
### dotnetcoreapp.runtimeconfig.json
### web.config
### Procfile
### .ebextensions
### .platform
```

È possibile includere più applicazioni nel bundle di origine. Nell'esempio seguente vengono definite due applicazioni da eseguire sullo stesso server Web. Per eseguire più applicazioni, è necessario includere un Procfile nel bundle di origine. Per un esempio completo di applicazione, consulta - apps.zip. dotnet-core-linux-multiple

```
### DotnetMultipleApp1
#   ### Amazon.Extensions.Configuration.SystemsManager.dll
#   ### appsettings.Development.json
#   ### appsettings.json
#   ### AWSSDK.Core.dll
#   ### AWSSDK.Extensions.NETCore.Setup.dll
#   ### AWSSDK.SimpleSystemsManagement.dll
#   ### DotnetMultipleApp1.deps.json
#   ### DotnetMultipleApp1.dll
#   ### DotnetMultipleApp1.pdb
#   ### DotnetMultipleApp1.runtimeconfig.json
#   ### Microsoft.Extensions.PlatformAbstractions.dll
#   ### Newtonsoft.Json.dll
#   ### web.config
### DotnetMultipleApp2
#   ### Amazon.Extensions.Configuration.SystemsManager.dll
#   ### appsettings.Development.json
#   ### appsettings.json
#   ### AWSSDK.Core.dll
#   ### AWSSDK.Extensions.NETCore.Setup.dll
#   ### AWSSDK.SimpleSystemsManagement.dll
#   ### DotnetMultipleApp2.deps.json
#   ### DotnetMultipleApp2.dll
#   ### DotnetMultipleApp2.pdb
#   ### DotnetMultipleApp2.runtimeconfig.json
#   ### Microsoft.Extensions.PlatformAbstractions.dll
#   ### Newtonsoft.Json.dll
#   ### web.config
### Procfile
### .ebextensions
### .platform
```

Utilizzo di un Procfile per configurare l'ambiente.NET Core su Linux Elastic Beanstalk

Per eseguire più applicazioni sullo stesso server Web, è necessario includere un bundle `Procfile` di origine che indichi a Elastic Beanstalk quali applicazioni eseguire.

Si consiglia di fornire sempre un `Procfile` nel bundle di origine insieme all'applicazione. In questo modo si controlla con precisione quali processi vengono eseguiti da Elastic Beanstalk per l'applicazione e quali argomenti ricevono questi processi.

Nell'esempio seguente viene utilizzato un `Procfile` per specificare due applicazioni che Elastic Beanstalk deve eseguire sullo stesso server Web.

Example Procfile

```
web: dotnet ./dotnet-core-app1/dotnetcoreapp1.dll
web2: dotnet ./dotnet-core-app2/dotnetcoreapp2.dll
```

Per informazioni dettagliate sulla scrittura e l'utilizzo di un file`Procfile`, vedereBuildfile e Procfile.

Configurazione del server proxy

AWS Elastic Beanstalk utilizza NGINX come reverse proxy per inoltrare le richieste all'applicazione. Elastic Beanstalk fornisce una configurazione NGINX predefinita che puoi estendere o sovrascrivere completamente con la tua configurazione.

Per impostazione predefinita, Elastic Beanstalk configura il proxy NGINX per inoltrare le richieste all'applicazione sulla porta 5000. È possibile sostituire la porta predefinita impostando la proprietà dell'ambiente `PORT` sulla porta di ascolto dell'applicazione principale.

> **Note**
>
> La porta su cui l'applicazione è in ascolto non influisce sulla porta utilizzata dal server NGINX per ricevere le richieste dal sistema di bilanciamento del carico.

Configurazione del server proxy sulla versione della piattaforma

Tutte le AL2 piattaforme AL2023/supportano una funzionalità di configurazione proxy uniforme. Per ulteriori informazioni sulla configurazione del server proxy sulle versioni della piattaforma che eseguono AL2023/AL2, consultaConfigurazione del proxy inverso.

Il seguente file di configurazione di esempio estende la configurazione NGINX del tuo ambiente. La configurazione indirizza le richieste `/api` a una seconda applicazione Web in ascolto sulla porta 5200 del server Web. Per impostazione predefinita, Elastic Beanstalk inoltra le richieste a una singola applicazione in ascolto sulla porta 5000.

Example **`01_custom.conf`**

```
location /api {
```

```
    proxy_pass          http://127.0.0.1:5200;
    proxy_http_version  1.1;

    proxy_set_header   Upgrade $http_upgrade;
    proxy_set_header   Connection $http_connection;
    proxy_set_header   Host $host;
    proxy_cache_bypass $http_upgrade;
    proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header   X-Forwarded-Proto $scheme;
}
```

Il AWS Toolkit for Visual Studio - Lavorare con.Net Core su Elastic Beanstalk

In questo argomento viene illustrato come eseguire le seguenti attività utilizzando AWS Toolkit for Visual Studio:

- Crea un'applicazione Web ASP.NET Core utilizzando un modello Visual Studio.
- Crea un ambiente Elastic Beanstalk Amazon Linux.
- Distribuzione dell'applicazione Web ASP.NET Core nel nuovo ambiente Amazon Linux.

Questo argomento illustra anche come utilizzare il AWS Toolkit for Visual Studio per gestire gli ambienti applicativi Elastic Beanstalk e monitorare lo stato dell'applicazione.

AWS Toolkit for Visual Studio È un plugin per l'IDE di Visual Studio. Con il toolkit puoi distribuire e gestire applicazioni in Elastic Beanstalk mentre utilizzi l'ambiente Visual Studio.

Sections

- Prerequisiti
- Creazione di un nuovo progetto di applicazione
- Creazione di un ambiente Elastic Beanstalk e distribuzione dell'applicazione
- Terminazione di un ambiente
- Gestione degli ambienti applicativi Elastic Beanstalk
- Monitoraggio dello stato dell'applicazione

Prerequisiti

Prima di iniziare questa esercitazione, è necessario installare AWS Toolkit for Visual Studio. Per istruzioni, consulta Configurazione di AWS Toolkit for Visual Studio.

Se non hai mai usato il toolkit prima, la prima cosa che devi fare dopo averlo installato è registrare le tue AWS credenziali nel toolkit. Per ulteriori informazioni, consulta Fornitura delle credenziali AWS.

Creazione di un nuovo progetto di applicazione

Se non hai un progetto di applicazione .NET Core in Visual Studio puoi crearne facilmente uno utilizzando uno dei modelli di progetto di Visual Studio.

Creazione di un nuovo progetto di applicazione Web ASP.NET Core

1. Nel menu File di Visual Studio scegli New (Nuovo), quindi Project (Progetto).
2. Nella finestra di dialogo Create a new project (Crea un nuovo progetto) seleziona C#, poi Linux, quindi Cloud.
3. Dall'elenco dei modelli di progetto visualizzato seleziona ASP.NET Core Web Application (Applicazione Web ASP.NET Core), e quindi seleziona Next (Avanti).

> **Note**
>
> Se l'opzione ASP.NET Core Web Application (Applicazione Web ASP.NET Core) non è elencata nei modelli di progetto, puoi installarla in Visual Studio.
>
> 1. Scorrere fino alla fine dell'elenco dei modelli e selezionare il collegamento Installa altri strumenti e funzionalità che si trova sotto l'elenco dei modelli.
> 2. Se viene richiesto di consentire all'applicazione Visual Studio di apportare modifiche al dispositivo, seleziona Sì.
> 3. Scegli la scheda Workloads (Carichi di lavoro) quindi seleziona ASP.NET and web development. (ASP.NET e sviluppo Web).
> 4. Seleziona il pulsante Modify (Modifica) . Il programma Visual Studio Installer installerà il modello di progetto.
> 5. Al termine del programma di installazione, esci dal pannello per tornare al punto in cui avevi interrotto con Visual Studio.

4. Nella finestra di dialogo Configure your new project (Configura nuovo progetto) immetti un Project name (Nome del progetto). Per impostazione predefinita, il nome della soluzione è il nome del progetto. Quindi, scegli Create (Crea).
5. Nella finestra di dialogo Create a new ASP.NET Core web application (Crea una nuova applicazione Web ASP.NET Core) seleziona NET Core e quindi seleziona ASP.NET Core 3.1. Dall'elenco dei tipi di applicazione visualizzati seleziona Web Application (Applicazione Web), quindi seleziona il pulsante Create (Crea).

Visual Studio visualizza la finestra di dialogo Creating Project (Creazione progetto) mentre viene creata l'applicazione. Al termine della generazione dell'applicazione, Visual Studio visualizza un pannello con il nome dell'applicazione.

Creazione di un ambiente Elastic Beanstalk e distribuzione dell'applicazione

In questa sezione viene descritto come creare un ambiente Elastic Beanstalk per l'applicazione e come distribuire l'applicazione nell'ambiente.

Per creare un nuovo ambiente e distribuire l'applicazione

1. In Visual Studio seleziona View (Visualizza), quindi Solution Explorer (Esplora soluzioni).
2. In Solution Explorer (Esplora soluzioni), apri il menu contestuale (fai clic con il pulsante destro del mouse) per l'applicazione e quindi seleziona Publish to (Pubblica su) AWS Elastic Beanstalk.
3. Nella procedura guidata Publish to (Pubblica su) AWS Elastic Beanstalk, inserisci le informazioni sul tuo account.
   a. In Account profile to use (Profilo account da utilizzare), seleziona l’account di default (predefinito) oppure seleziona l’icona Add another account (Aggiungi un altro account) per inserire le informazioni relative al nuovo account.
   b. In Region (Regione), seleziona la regione in cui vuoi distribuire l'applicazione. Per informazioni sulle AWS regioni disponibili, consulta AWS Elastic Beanstalk Endpoints and Quotas in. Riferimenti generali di AWS Se selezioni una regione non supportata da Elastic Beanstalk, l'opzione per la distribuzione in Elastic Beanstalk non è disponibile.
   c. Seleziona Create a new application environment (Crea un nuovo ambiente di applicazione), quindi scegli Next (Avanti).
4. Nella finestra di dialogo Application Environment (Ambiente applicazione) immetti i dettagli relativi al nuovo ambiente dell’applicazione.

5. Nella finestra di dialogo AWSdelle opzioni successiva, imposta EC2 le opzioni Amazon e altre opzioni AWS correlate per l'applicazione distribuita.

   a. Per il tipo di contenitore, seleziona Amazon Linux 2 v ***<n.n.n>*** a 64 bit con .NET Core.

   > **Note**
   >
   > Si consiglia di selezionare la versione della piattaforma corrente di Linux. Questa versione contiene le correzioni più recenti di sicurezza e bug incluse nella nostra ultima Amazon Machine Image (AMI).

   b. Per Instance Type (Tipo di istanza), seleziona t2.micro. La scelta di un tipo di istanza micro ridurrà al minimo il costo associato all'esecuzione dell'istanza.

   c. In Key pair (Coppia di chiavi), seleziona Create new key pair (Crea nuova coppia di chiavi). Immettere un nome per la nuova coppia di chiavi, quindi scegliere OK. In questo esempio viene utilizzato **`myuseastkeypair`**. Una key pair consente l'accesso da desktop remoto alle istanze Amazon EC2 . Per ulteriori informazioni sulle coppie di EC2 chiavi Amazon, consulta Using Credentials nella Amazon Elastic Compute Cloud User Guide.

   d. Per un'applicazione semplice con traffico ridotto, seleziona Single instance environment (Ambiente a istanza singola). Per ulteriori informazioni, consulta Tipi di ambiente

   e. Seleziona Avanti.

   Per ulteriori informazioni sulle AWS opzioni non utilizzate in questo esempio, consulta le seguenti pagine:

   - PerUtilizzo di AMI personalizzate consulta Utilizzo di un'immagine Amazon Machine Image (AMI) personalizzata nell'ambiente Elastic Beanstalk.
   - Se non selezioni Single instance environment (Ambiente a istanza singola) devi scegliere un Load balance type (Tipo di bilanciamento del carico). Per ulteriori informazioni, consultare Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk.
   - Elastic Beanstalk utilizza la configurazione predefinita di Amazon VPC (Amazon Virtual Private Cloud) se non hai scelto Use non-default VPC (Utilizza VPC non predefinito). Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon VPC.
   - Scegliendo l'opzione Abilita distribuzioni in sequenza una distribuzione viene suddivisa in batch per evitare potenziali tempi di inattività durante le distribuzioni. Per ulteriori informazioni, consultare . Distribuzione di applicazioni in ambienti Elastic Beanstalk.

- L'opzione Relational Database Access (Accesso al database relazionale) consente di connettere l'ambiente Elastic Beanstalk a un database Amazon RDS creato in precedenza con gruppi di sicurezza DB Amazon RDS. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza nella Guida per l'utente di Amazon RDS.

6. Seleziona Next (Avanti) nella finestra di dialogo Permissions (Autorizzazioni) .
7. Seleziona Next (Avanti) nella finestra di dialogo Applications Options (Opzioni applicazioni).
8. Verifica le opzioni di distribuzione. Dopo aver verificato che le impostazioni siano corrette, seleziona Distribuisci.

L'applicazione Web ASP.NET Core viene esportata come file di distribuzione Web. Questo file viene quindi caricato in Amazon S3 e registrato come nuova versione dell'applicazione con Elastic Beanstalk. La caratteristica di distribuzione di Elastic Beanstalk monitora il tuo ambiente finché non sarà disponibile con il codice appena distribuito. Visualizzerai lo Status (Stato) per il tuo ambiente nella scheda Env:<environment name>. Dopo che lo stato passa a Environment is healthy (Ambiente integro), seleziona l'indirizzo URL per avviare l'applicazione Web.

Terminazione di un ambiente

Per evitare di incorrere in addebiti per le AWS risorse non utilizzate, è possibile utilizzare il per AWS Toolkit for Visual Studio terminare un ambiente in esecuzione.

> **Note**
>
> Puoi sempre avviare un nuovo ambiente utilizzando la stessa versione in un secondo momento.

Come terminare un ambiente

1. Espandi il nodo Elastic Beanstalk e il nodo dell'applicazione. In AWS Explorer, apri il menu contestuale (fai clic con il pulsante destro del mouse) per l'ambiente dell'applicazione e seleziona Terminate Environment (Termina ambiente).
2. Quando richiesto, seleziona Yes (Sì) per confermare che intendi terminare l'ambiente. Elastic Beanstalk impiega alcuni minuti per AWS interrompere le risorse in esecuzione nell'ambiente.

Lo Status (Stato) per l'ambiente nella scheda Env:<environment name> cambierà in Terminating (Terminazione in corso) e infine in Terminated (Terminato).

> **Note**
>
> Quando termini l'ambiente, il CNAME associato all'ambiente terminato diventa disponibile e può essere utilizzato da un altro utente.

Gestione degli ambienti applicativi Elastic Beanstalk

Questa sezione illustra le impostazioni specifiche del servizio che puoi modificare in AWS Toolkit for Visual Studio come parte della configurazione dell'ambiente applicativo. Con AWS Toolkit for Visual Studio e AWS Management Console, puoi modificare il provisioning e la configurazione delle AWS risorse utilizzate dagli ambienti applicativi. Per informazioni su come gestire gli ambienti applicativi utilizzando la Console di AWS gestione, consulta. Gestione degli ambienti Elastic Beanstalk

Modifica delle impostazioni di configurazione dell'ambiente

Quando distribuisci la tua applicazione, Elastic Beanstalk configura diversi servizi di cloud computing connessi. AWS È possibile controllare la configurazione di questi singoli servizi utilizzando AWS Toolkit for Visual Studio.

Per modificare le impostazioni dell'ambiente di un'applicazione

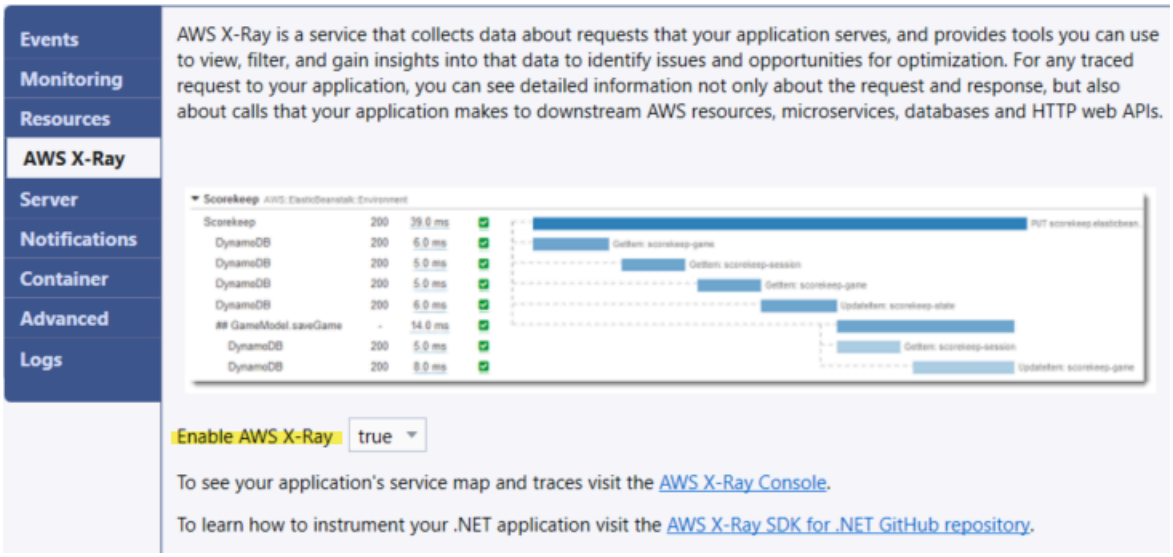
1. In Visual Studio, nel menu File, seleziona AWS Explorer.
2. Espandi il nodo Elastic Beanstalk e il nodo dell'applicazione. Apri il menu contestuale (pulsante destro del mouse) per l'ambiente dell'applicazione e seleziona View Status (Visualizza stato).

   È ora possibile configurare le impostazioni per i componenti seguenti:

   - AWS X-Ray
   - Server
   - Sistema di bilanciamento del carico (si applica solo agli ambienti con più istanze)
   - Auto Scaling (si applica solo agli ambienti con più istanze)
   - Notifications
   - Contenitore
   - Opzioni di configurazione avanzate

Configurazione di AWS X-Ray utilizzando AWS il toolkit per Visual Studio

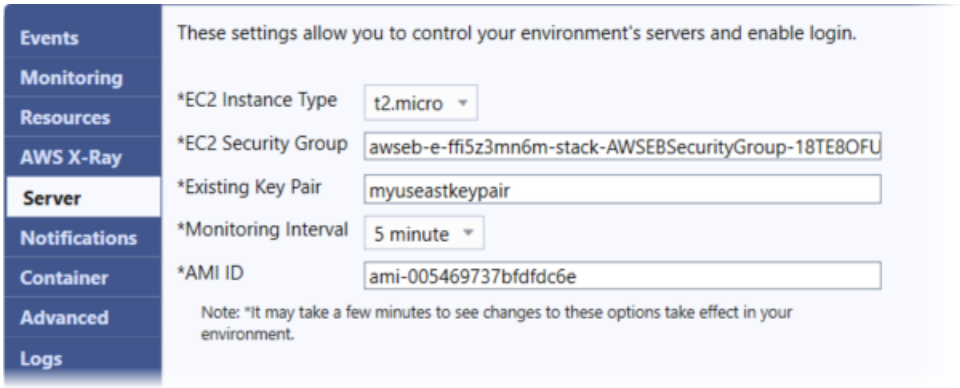
AWS X-Ray fornisce funzionalità di tracciamento delle richieste, raccolta delle eccezioni e profilazione. Con il pannello AWS X-Ray, puoi abilitare o disabilitare X-Ray per la tua applicazione. Per ulteriori informazioni su X-Ray, consulta la Guida per gli sviluppatori di AWS X-Ray.



Configurazione delle istanze utilizzando EC2 il toolkit per Visual Studio AWS

Puoi utilizzare Amazon Elastic Compute Cloud (Amazon EC2) per avviare e gestire istanze di server nei data center di Amazon. Puoi utilizzare le istanze EC2 del server Amazon in qualsiasi momento, per tutto il tempo necessario e per qualsiasi scopo legale. Le istanze sono disponibili in diverse dimensioni e configurazioni. Per ulteriori informazioni, consulta Amazon EC2.

Puoi modificare la configurazione dell' EC2 istanza Amazon con la scheda Server all'interno della scheda Ambiente applicativo in AWS Toolkit for Visual Studio.



Tipi di EC2 istanze Amazon

L'opzione Tipo di istanza visualizza i tipi di istanza disponibili per l'applicazione Elastic Beanstalk. Cambia il tipo di istanza per selezionare un server con le caratteristiche (come quantità di memoria e potenza della CPU) più adatte alla tua applicazione. Ad esempio, le applicazioni con operazioni intensive e di lunga durata possono richiedere più memoria o CPU.

Per ulteriori informazioni sui tipi di EC2 istanze Amazon disponibili per la tua applicazione Elastic Beanstalk, consulta Instance Types nella Amazon Elastic Compute Cloud User Guide.

Gruppi EC2 di sicurezza Amazon

Puoi controllare l'accesso alla tua applicazione Elastic Beanstalk utilizzando EC2 un Amazon Security Group. Un gruppo di sicurezza definisce le regole del firewall per le proprie istanze. Tali regole specificano quale traffico in entrata deve essere distribuito sulla tua istanza. Tutto l'altro traffico in entrata viene scartato. Puoi modificare le regole per un gruppo in qualsiasi momento. Le nuove regole vengono automaticamente applicate per tutte le istanze in esecuzione e le istanze avviate in futuro.

Puoi specificare quali Amazon EC2 Security Groups controllano l'accesso alla tua applicazione Elastic Beanstalk. A tale scopo, inserisci i nomi di specifici gruppi di EC2 sicurezza Amazon (separando più gruppi di sicurezza con virgole) nella casella di testo Gruppi EC2 di sicurezza. È possibile eseguire questa operazione utilizzando la Console di AWS gestione o il AWS Toolkit for Visual Studio.

Per creare un gruppo di sicurezza utilizzando il AWS toolkit per Visual Studio

1. In Visual Studio, in AWS Explorer, espandi il EC2 nodo Amazon, quindi seleziona Security Groups.
2. Seleziona Create Security Group (Crea gruppi di sicurezza), quindi immetti un nome e una descrizione per il gruppo di sicurezza.
3. Seleziona OK.

Per ulteriori informazioni su Amazon EC2 Security Groups, consulta Using Security Groups nella Amazon Elastic Compute Cloud User Guide.

Coppie di EC2 chiavi Amazon

Puoi accedere in modo sicuro alle EC2 istanze Amazon fornite per la tua applicazione Elastic Beanstalk con una coppia di chiavi Amazon. EC2

> **Important**
>
> Devi creare una coppia di EC2 chiavi Amazon e configurare le EC2 istanze Amazon fornite da Elastic Beanstalk per poter accedere a queste istanze. Puoi creare la tua coppia di chiavi utilizzando la AWS procedura guidata Publish to all'interno di AWS Toolkit for Visual Studio quando distribuisci l'applicazione su Elastic Beanstalk. Per creare ulteriori coppie di chiavi utilizzando il Toolkit, segui le fasi descritte qui di seguito. In alternativa, puoi configurare le tue coppie di EC2 chiavi Amazon utilizzando la console di AWS gestione. Per istruzioni sulla creazione di una coppia di key pair per Amazon EC2, consulta la Amazon Elastic Compute Cloud Getting Started Guide.

La casella di testo Existing Key Pair ti consente di specificare il nome di una coppia di EC2 chiavi Amazon che puoi utilizzare per accedere in modo sicuro alle EC2 istanze Amazon che eseguono la tua applicazione Elastic Beanstalk.

Per specificare il nome di una coppia di EC2 chiavi Amazon

1. Espandi il EC2 nodo Amazon e seleziona Key Pairs.
2. Seleziona Create Key Pair (Crea coppia di chiavi) e inserisci il nome della coppia di chiavi.
3. Seleziona OK.

Per ulteriori informazioni sulle coppie di EC2 chiavi Amazon, consulta Using Amazon EC2 Credentials nella Amazon Elastic Compute Cloud User Guide. Per ulteriori informazioni sulla connessione alle EC2 istanze Amazon, consulta

Intervallo di monitoraggio

Per impostazione predefinita, sono abilitate solo le CloudWatch metriche Amazon di base. che restituiscono dati in periodi di cinque minuti. Puoi abilitare CloudWatch metriche più granulari di un minuto selezionando 1 minuto per l'intervallo di monitoraggio nella sezione Server della scheda Configurazione per il tuo ambiente in. AWS Toolkit for Eclipse

> **Note**
>
> I costi CloudWatch del servizio Amazon possono essere applicati per metriche a intervalli di un minuto. CloudWatchPer ulteriori informazioni, consulta Amazon.

ID AMI personalizzata

Puoi sostituire l'AMI predefinita utilizzata per le tue EC2 istanze Amazon con la tua AMI personalizzata inserendo l'identificatore dell'AMI personalizzato nella casella ID AMI personalizzato nella sezione Server della scheda Configurazione per il tuo ambiente in. AWS Toolkit for Eclipse

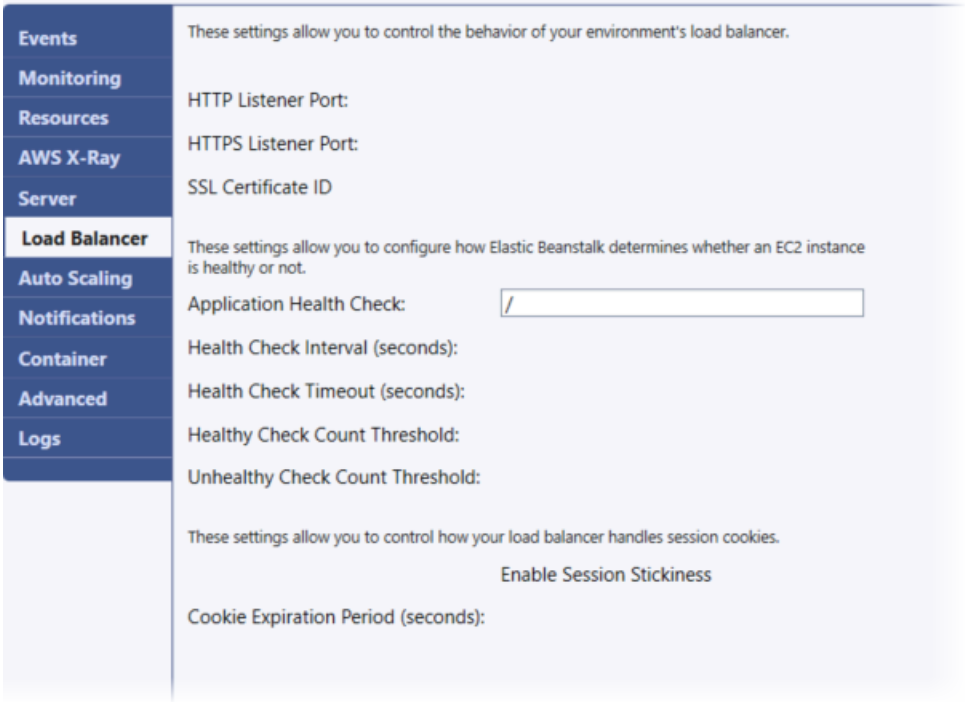
> **Important**
>
> L'utilizzo di un'AMI personalizzata è un'attività avanzata che devi completare con estrema attenzione. Se hai bisogno di un'AMI personalizzata, ti consigliamo di iniziare con l'AMI predefinita di Elastic Beanstalk e di modificarla. Per essere considerate affidabili, Elastic Beanstalk EC2 si aspetta che le istanze Amazon soddisfino una serie di requisiti, tra cui un host manager funzionante. Se questi requisiti non vengono soddisfatti, il tuo ambiente potrebbe non funzionare in modo corretto.

Configurazione di Elastic Load Balancing utilizzando AWS il toolkit per Visual Studio

Elastic Load Balancing è un servizio Web Amazon che aiuta a migliorare la disponibilità e la scalabilità della tua applicazione. Questo servizio semplifica la distribuzione dei carichi di applicazioni tra due o più EC2 istanze Amazon. Elastic Load Balancing migliora la disponibilità attraverso la ridondanza e supporta l'aumento di traffico per la tua applicazione.

Con Elastic Load Balancing, è possibile distribuire e bilanciare automaticamente il traffico delle applicazioni in entrata tra tutte le istanze in esecuzione. È inoltre possibile aggiungere facilmente nuove istanze quando è necessario aumentare la capacità dell'applicazione.

Elastic Beanstalk effettua automaticamente il provisioning di Elastic Load Balancing quando distribuisci un'applicazione. Puoi modificare la configurazione dell'istanza Amazon dell'ambiente EC2 Elastic Beanstalk con la scheda Load Balancer all'interno della scheda dell'ambiente applicativo in Toolkit for AWS Visual Studio.



Le seguenti sezioni descrivono i parametri di Elastic Load Balancing che puoi configurare per la tua applicazione.

Porte

Il sistema di bilanciamento del carico predisposto per gestire le richieste per l'applicazione Elastic Beanstalk invia le richieste alle istanze EC2 Amazon che eseguono l'applicazione. Il sistema di bilanciamento del carico fornito può ascoltare le richieste sulle porte HTTP e HTTPS e instradare le richieste verso le EC2 istanze Amazon dell'applicazione. AWS Elastic Beanstalk Per impostazione predefinita, il sistema di bilanciamento del carico gestisce le richieste sulla porta HTTP. Per fare in modo che funzioni, almeno una delle porte (HTTP o HTTPS) deve essere attiva.

> **Important**
>
> Verifica che la porta specificata non sia bloccata; in caso contrario, gli utenti non saranno in grado di connettersi alla tua applicazione Elastic Beanstalk.

Controllo della porta HTTP

Per disattivare la porta HTTP, seleziona OFF per HTTP Listener Port (Porta HTTP del listener). Per attivare la porta HTTP, seleziona una porta HTTP (ad esempio, 80) dall'elenco.

> **Note**
>
> Per accedere al tuo ambiente utilizzando una porta diversa dalla porta 80 predefinita, come la porta 8080, aggiungi un listener al sistema di bilanciamento del carico esistente e configura il nuovo listener per l'ascolto su quella porta.
> Ad esempio, utilizzando il comando AWS CLI for Classic load balancer, digita il seguente comando, sostituendolo *LOAD_BALANCER_NAME* con il nome del tuo load balancer per Elastic Beanstalk.
>
> ```
> aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
> --listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
> InstancePort=80"
> ```
>
> Ad esempio, utilizzando AWS CLI for Application Load Balancers, digita il comando seguente, sostituendolo *LOAD_BALANCER_ARN* con l'ARN del tuo load balancer per Elastic Beanstalk.
>
> ```
> aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
> --port 8080
> ```
>
> Se desideri che Elastic Beanstalk monitori il tuo ambiente, non rimuovere il listener sulla porta 80.

Controllo della porta HTTPS

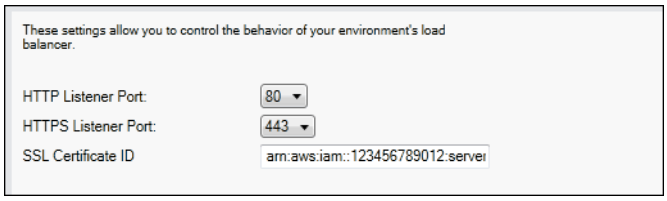
Elastic Load Balancing supporta il HTTPS/TLS protocollo per abilitare la crittografia del traffico per le connessioni dei client al sistema di bilanciamento del carico. Le connessioni dal sistema di bilanciamento del carico alle EC2 istanze utilizzano la crittografia in testo semplice. Per impostazione predefinita, la porta HTTPS non è attiva.

Per attivare la porta HTTPS

1. Crea un nuovo certificato utilizzando AWS Certificate Manager (ACM) o carica un certificato e una chiave su (IAM). AWS Identity and Access Management Per ulteriori informazioni sulla

richiesta di un certificato ACM, consulta Richiesta di un certificato nella Guida per l'utente di AWS Certificate Manager . Per ulteriori informazioni sull'importazione di certificati di terza parte in ACM, consulta Importazione di certificati nella Guida per l'utente di AWS Certificate Manager . Se ACM non è disponibile nella tua regione, utilizza AWS Identity and Access Management (IAM) per caricare un certificato di terze parti. I servizi ACM e IAM archiviano il certificato e forniscono un Amazon Resource Name (ARN) per il certificato SSL. Per ulteriori informazioni sulla creazione e sul caricamento dei certificati in IAM, consulta l'argomento relativo all'utilizzo dei certificati server nella Guida per l'utente di IAM.

2. Specifica la porta HTTPS selezionando una porta per HTTPS Listener Port (Porta HTTPS del listener).



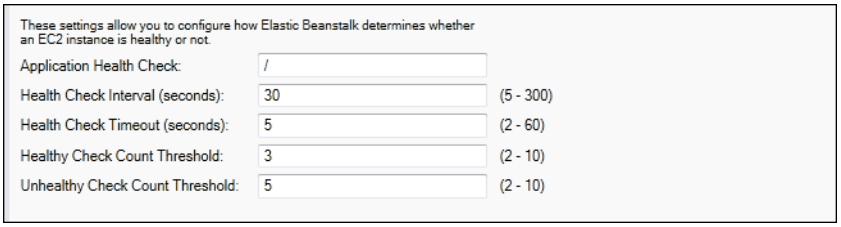
3. In SSL Certificate ID (ID certificato SSL) immetti l'Amazon Resources Name (ARN) del certificato SSL. Ad esempio, **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** o **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**. Usa il certificato SSL creato o caricato nella fase 1.

Per disattivare la porta HTTPS, seleziona OFF per HTTPS Listener Port (Porta HTTPS del listener).

Controlli dello stato

La definizione di controllo dello stato include un URL a cui inviare una query per lo stato dell'istanza. Per impostazione predefinita, Elastic Beanstalk usa TCP: 80 per container non legacy e HTTP: 80 per container legacy. Puoi sostituire l'URL di default con un URL di una risorsa esistente nella tua applicazione (ad esempio, `/myapp/default.aspx`) immettendolo nella casella Application health check URL (URL controllo dello stato applicazione). Se ignori l'URL predefinito, Elastic Beanstalk utilizza HTTP per eseguire le query sulla risorsa. Per verificare se stai utilizzando un tipo di container legacy, consulta the section called “Perché alcune versioni di piattaforma sono contrassegnate come legacy?”

Puoi controllare le impostazioni per il controllo dello stato utilizzando la sezione EC2 Instance Health Check del pannello Load Balancing.



La definizione di controllo dello stato include un URL a cui inviare una query per lo stato dell'istanza. Sostituisci l'URL di default con un URL di una risorsa esistente nella tua applicazione (ad esempio, `/myapp/index.jsp`) immettendolo nella casella Application Health Check URL (URL controllo dello stato applicazione).

Di seguito sono descritti i parametri di controllo dello stato che puoi impostare per l'applicazione.

- Per Health Check Interval (secondi), inserisci il numero di secondi di attesa di Elastic Load Balancing tra i controlli di integrità per le istanze Amazon dell'applicazione. EC2
- In Health Check Timeout (Timeout del controllo dello stato (secondi)), specifica il numero di secondi che Elastic Load Balancing deve aspettare per avere una risposta prima di ritenere che l'istanza non risponda.
- In Healthy Check Count Threshold (Soglia di conteggio dei controlli integri) e Unhealthy Check Count Threshold (Soglia di conteggio dei controlli non integri), specifica il numero di sondaggi URL consecutivi riusciti o non riusciti prima che Elastic Load Balancing modifichi lo stato di integrità dell'istanza. Ad esempio, specificando **5** nella casella Unhealthy Check Count Threshold (Soglia di conteggio dei controlli non integri), l'URL restituisce un messaggio di errore o di timeout per cinque volte consecutive prima che Elastic Load Balancing consideri il controllo dello stato come non riuscito.

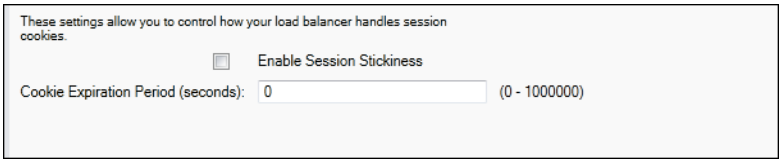
Sessioni

Per impostazione predefinita, un sistema di bilanciamento del carico prevede l'instradamento di ogni richiesta in modo indipendente all'istanza del server con il carico minore. In confronto, una sticky session associa una sessione utente a una determinata istanza del server, in modo che tutte le richieste provenienti dall'utente durante la sessione siano inviate alla stessa istanza del server.

Elastic Beanstalk utilizza i cookie HTTP generati dal sistema di bilanciamento del carico quando le sticky session sono attivate per un'applicazione. Il sistema di bilanciamento del carico utilizza un cookie speciale generato dal sistema di bilanciamento del carico per monitorare l'istanza dell'applicazione per ogni richiesta. Quando il sistema di bilanciamento del carico riceve una richiesta, verifica innanzitutto se questo cookie è presente nella richiesta. Se è presente, la richiesta viene

inviata all'istanza dell'applicazione specificata nel cookie. Se non vengono trovati cookie, il sistema di bilanciamento del carico sceglie un'istanza delle applicazioni basata su algoritmo di bilanciamento del carico esistente. Un cookie è inserito nella risposta per le richieste successive vincolanti dallo stesso utente per quell'istanza dell'applicazione. La configurazione della policy definisce la scadenza di un cookie che stabilisce la durata di validità per ogni cookie.

Puoi utilizzare la sezione Sessions (Sessioni) nella scheda Load Balancer (Sistema di bilanciamento del carico) per specificare se il sistema di bilanciamento del carico della tua applicazione consente l'adesione della sessione.



Per ulteriori informazioni su Elastic Load Balancing, consulta la Guida per sviluppatori di Elastic Load Balancing.

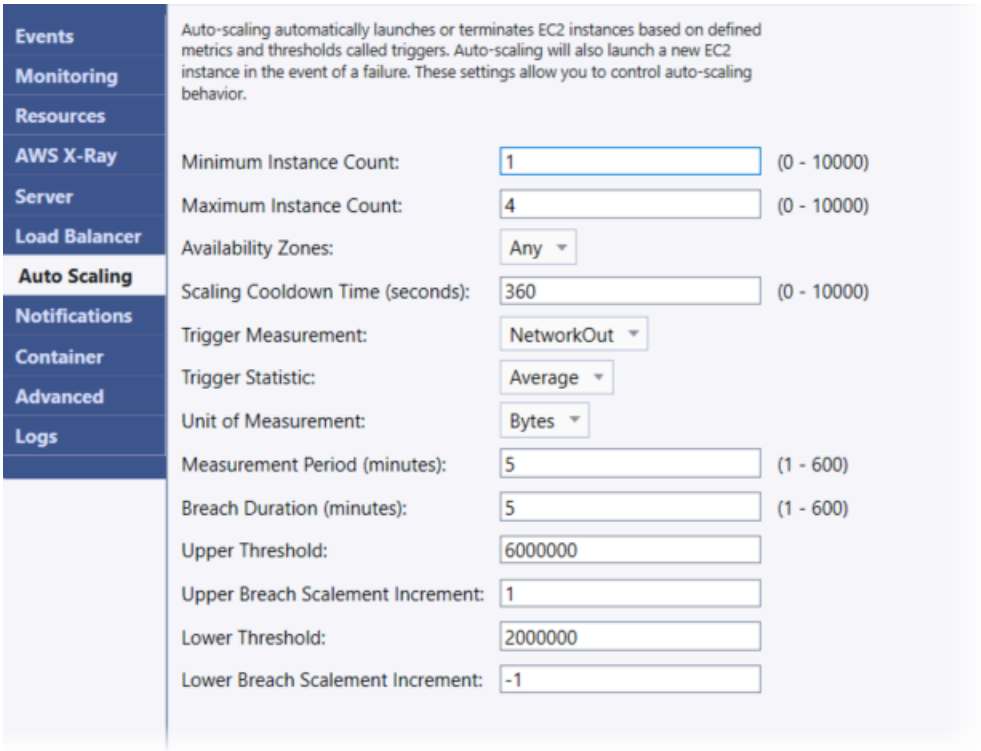
Configurazione dell'Auto Scaling utilizzando AWS il toolkit per Visual Studio

Amazon EC2 Auto Scaling è un servizio web Amazon progettato per avviare o terminare automaticamente le istanze EC2 Amazon in base a trigger definiti dall'utente. Puoi configurare gruppi Auto Scaling e associare trigger a questi gruppi per dimensionare automaticamente le risorse di calcolo basate su parametri come l'utilizzo della banda o della CPU. Amazon EC2 Auto Scaling collabora con Amazon CloudWatch per recuperare i parametri per le istanze del server che eseguono la tua applicazione.

Amazon EC2 Auto Scaling ti consente di prendere un gruppo di EC2 istanze Amazon e impostare vari parametri per aumentare o diminuire automaticamente il numero di questo gruppo. Amazon EC2 Auto Scaling può aggiungere o rimuovere EC2 istanze Amazon da quel gruppo per aiutarti a gestire senza problemi le variazioni del traffico della tua applicazione.

Amazon EC2 Auto Scaling monitora anche lo stato di ogni EC2 istanza Amazon che avvia. Se un'istanza si interrompe in modo imprevisto, Amazon EC2 Auto Scaling rileva la terminazione e avvia un'istanza sostitutiva. Questa funzionalità consente di mantenere automaticamente un numero fisso e desiderato di EC2 istanze Amazon.

Elastic Beanstalk fornisce EC2 Amazon Auto Scaling per la tua applicazione. Puoi modificare la configurazione dell'istanza Amazon dell'ambiente EC2 Elastic Beanstalk con la scheda Auto Scaling all'interno della scheda Ambiente applicativo in Toolkit for AWS Visual Studio.

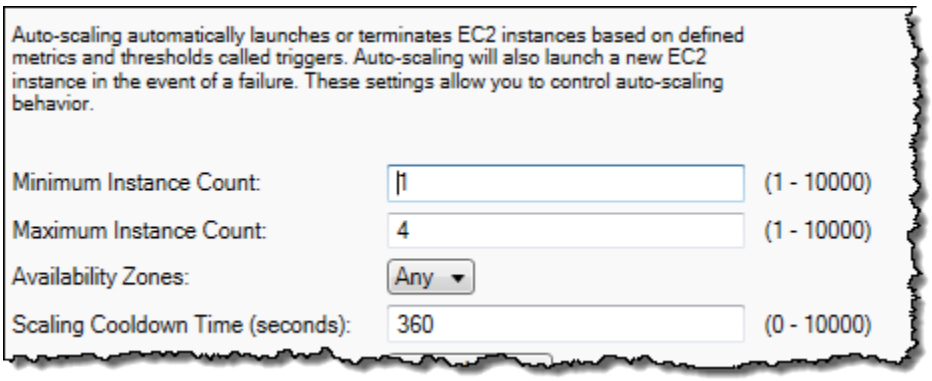


La seziona seguente illustra come configurare i parametri di Auto Scaling per la tua applicazione.

Avvio della configurazione

Puoi modificare la configurazione di avvio per controllare il modo in cui l'applicazione Elastic Beanstalk fornisce le risorse di EC2 Amazon Auto Scaling.

Le caselle Minimum Instance Count (Conteggio istanze minimo) e Maximum Instance Count (Conteggio istanze massimo) consentono di specificare la dimensione minima e massima del gruppo Auto Scaling usato dall'applicazione Elastic Beanstalk.



> **Note**
>
> Per mantenere un numero fisso di EC2 istanze Amazon, imposta Numero minimo di istanze e Numero massimo di istanze sullo stesso valore.

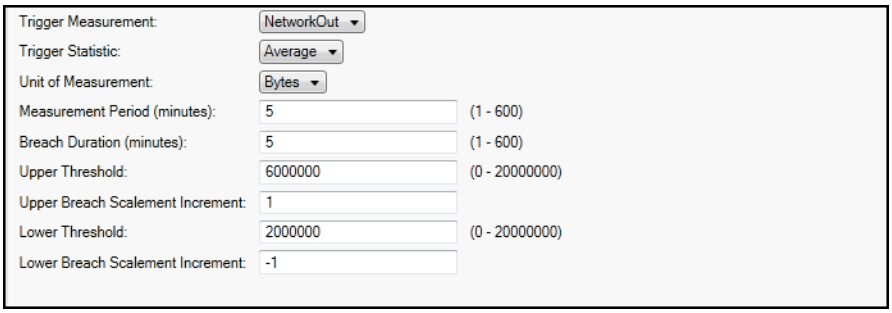
La casella Zone di disponibilità ti consente di specificare il numero di zone di disponibilità in cui desideri che si trovino le tue EC2 istanze Amazon. È importante impostare questo numero se desideri creare applicazioni con tolleranza ai guasti. Se una zona di disponibilità non è raggiungibile, le istanze saranno ancora in esecuzione in altre zone di disponibilità.

> **Note**
>
> Al momento, non è possibile specificare in quale zona di disponibilità si trova l'istanza.

Triggers

Un trigger è un meccanismo di Amazon EC2 Auto Scaling impostato per indicare al sistema quando si desidera aumentare (scalare in orizzontale) o diminuire (scalabilità interna) il numero di istanze. Puoi configurare i trigger in modo che si attivino su qualsiasi metrica pubblicata su Amazon CloudWatch (ad esempio, l'utilizzo della CPU) e determinare se le condizioni specificate sono state soddisfatte. Quando le soglie superiori o inferiori delle condizioni specificate per il parametro sono state violate per il periodo di tempo specificato, il trigger avvia un processo di lunga durata denominato attività di dimensionamento.

Puoi definire un trigger di scalabilità per la tua applicazione Elastic Beanstalk utilizzando Toolkit AWS for Visual Studio.



I trigger di Amazon EC2 Auto Scaling funzionano monitorando una CloudWatch metrica Amazon specifica di una particolare istanza. I trigger includono l'utilizzo della CPU, il traffico di rete e l'attività

del disco. Utilizza l'impostazione Trigger Measurement (Misurazione del trigger) per selezionare un parametro per il trigger.

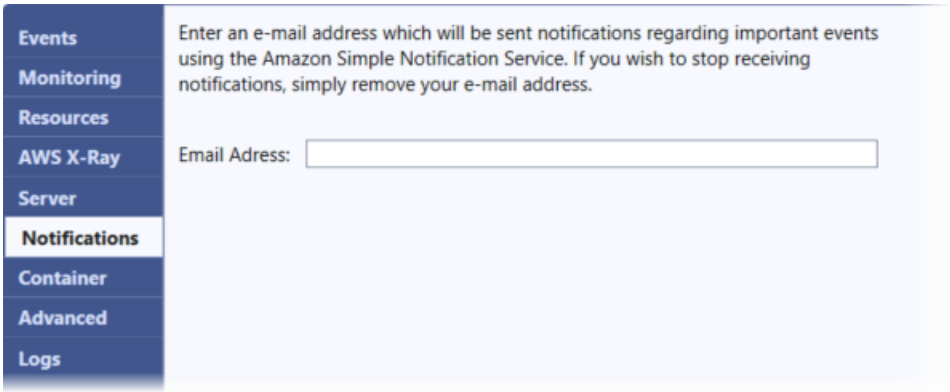
L'elenco seguente descrive i parametri di attivazione che puoi configurare utilizzando la AWS console di gestione.

- Puoi specificare le statistiche che il trigger deve utilizzare. Puoi selezionare Minimum (Minimo), Maximum (Massimo), Sum (Somma) o Average (Media) per Trigger Statistic (Statistica trigger).
- In Unit of Measurement (Unità di misura), specifica l'unità per la misurazione del trigger.
- Il valore nella casella Periodo di misurazione specifica la frequenza con cui Amazon CloudWatch misura le metriche relative al trigger. La Breach Duration (Durata della violazione) specifica la quantità di tempo in cui un parametro può essere oltre il limite definito, specificato in Upper Treshold (Soglia superiore) e in Lower Threshold (Soglia inferiore) prima che il trigger si attivi.
- Per Upper Breach Scale Increment e Lower Breach Scale Increment, specifica quante EC2 istanze Amazon aggiungere o rimuovere durante l'esecuzione di un'attività di scalabilità.

Per ulteriori informazioni su Amazon EC2 Auto Scaling, consulta la sezione Amazon Auto EC2 Scaling nella documentazione di Amazon Elastic Compute Cloud.

Configurazione delle notifiche utilizzando AWS il toolkit per Visual Studio

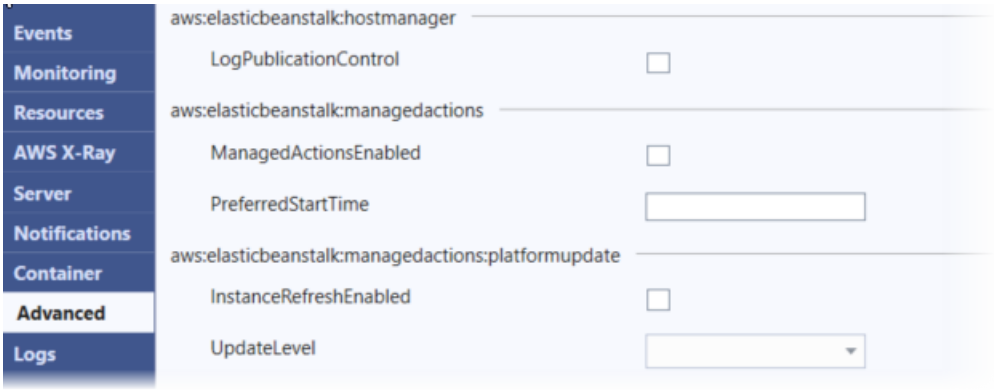
Elastic Beanstalk utilizza Amazon Simple Notification Service (Amazon SNS) per la notifica di eventi importanti che interessano l'applicazione. Per abilitare le notifiche di Amazon SNS, è sufficiente inserire il tuo indirizzo e-mail nella casella Email Address (Indirizzo e-mail). Per disabilitare queste notifiche, elimina il tuo indirizzo e-mail dalla casella di testo.



Configurazione di opzioni di ambiente aggiuntive utilizzando il AWS toolkit per Visual Studio

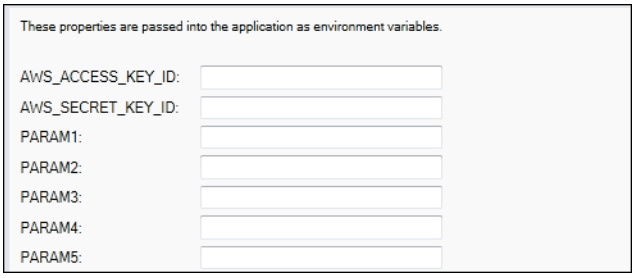
Elastic Beanstalk definisce un numero elevato di opzioni di configurazione che puoi utilizzare per configurare il comportamento dell'ambiente e le risorse che contiene. Le opzioni di configurazione sono organizzate in spazi dei nomi come `aws:autoscaling:asg`. Ogni spazio dei nomi definisce le opzioni per il gruppo Auto Scaling di un ambiente. Il pannello Advanced (Opzioni avanzate) elenca gli spazi dei nomi delle opzioni di configurazione in ordine alfabetico che è possibile aggiornare dopo la creazione dell'ambiente.

Per un elenco completo delle opzioni e dei namespace, inclusi i valori supportati e quelli predefiniti, vedi Opzioni generali per tutti gli ambienti e Opzioni della piattaforma .NET Core su Linux.



Configurazione dei contenitori.NET Core utilizzando il AWS toolkit per Visual Studio

Il pannello Container consente di specificare variabili di ambiente che puoi leggere dal codice dell'applicazione.



Monitoraggio dello stato dell'applicazione

Questo argomento spiega come monitorare lo stato del sito Web dell'applicazione. È importante sapere che il sito Web di produzione è disponibile e risponde alle richieste. Elastic Beanstalk fornisce caratteristiche che consentono di monitorare la reattività dell'applicazione. Monitora le statistiche sull'applicazione e ti avvisa quando le soglie vengono superate.

Per informazioni sul monitoraggio dello stato fornito da Elastic Beanstalk, consulta Report dello stato di base.

È possibile accedere alle informazioni operative sull'applicazione utilizzando il AWS Toolkit for Visual Studio o AWS la Console di gestione.

Il kit di strumenti mostra lo stato dell'ambiente e delle applicazioni nel campo Status (Stato).

Per monitorare lo stato dell'applicazione

1. In AWS Toolkit for Visual Studio, AWS in Explorer, espandi il nodo Elastic Beanstalk, quindi espandi il nodo dell'applicazione.
2. Apri il menu contestuale (pulsante destro del mouse) per l'ambiente dell'applicazione e seleziona View Status (Visualizza stato).
3. Nella scheda dell'ambiente dell'applicazione, seleziona Monitoring (Monitoraggio).

   Il pannello Monitoring (Monitoraggio) include un set di grafici che mostrano l'utilizzo delle risorse per un ambiente applicativo specifico.

   > **Note**
   >
   > Per impostazione predefinita, l'intervallo di tempo è impostato sull'ultima ora. Per modificare questa impostazione, nell'elenco Time Range (Intervallo tempo) seleziona un altro intervallo temporale.

È possibile utilizzare AWS Toolkit for Visual Studio o AWS la Console di gestione per visualizzare gli eventi associati all'applicazione.

Per visualizzare gli eventi dell'applicazione

1. Nel AWS Toolkit for Visual Studio, AWS in Explorer, espandi il nodo Elastic Beanstalk e il nodo dell'applicazione.
2. Apri il menu contestuale (pulsante destro del mouse) per l'ambiente dell'applicazione e seleziona View Status (Visualizza stato).
3. Nella scheda dell'ambiente dell'applicazione, seleziona Events (Eventi).

Migrazione da.NET su piattaforma Windows Server a .NET Core su piattaforma Linux su Elastic Beanstalk

Puoi migrare le applicazioni eseguite su .NET su piattaforme Windows Server a .NET Core su piattaforma Linux. Di seguito sono riportate alcune considerazioni sulla migrazione da piattaforme Windows a Linux.

Considerazioni sulla migrazione a .NET Core su piattaforma Linux

Area	Modifiche e informazioni
Configurazione dell'applicazione	Sulle piattaforme Windows, puoi utilizzare un manifesto di distribuzione per specificare le applicazioni eseguite nell'ambiente. .NET Core su piattaforma Linux utilizza un Procfile per specificare le applicazioni che vengono eseguite nelle istanze dell'ambiente. Per informazioni dettagliate sulle applicazioni di raggruppamento, consulta the section called "Raggruppamento delle applicazioni".
Server proxy	Sulle piattaforme Windows, è possibile utilizzare IIS come server proxy dell'applicazione. .NET Core su piattaforma Linux include nginx come proxy inverso per impostazione predefinita. Puoi scegliere di non usare alcun server proxy e usare Kestrel come server web della tua applicazione. Per ulteriori informazioni, consulta the section called "Server proxy".
Routing	Sulle piattaforme Windows, è possibile utilizzare IIS nel codice dell'applicazione e includere un manifesto di distribuzione per configurare il percorso IIS. Per .NET Core su piattaforma Linux usi il routing ASP.NET Core nel codice dell'applicazione e aggiorni la configurazione nginx dell'ambiente. Per ulteriori informazioni, consulta the section called "Server proxy".
Log	Le piattaforme Linux e Windows trasmettono log diversi. Per informazioni dettagliate, vedi the section called "Come Elastic Beanstalk configura i log CloudWatch".

Distribuzione di applicazioni Go con Elastic Beanstalk

Questo capitolo fornisce istruzioni per configurare e distribuire l'applicazione web Go su. AWS Elastic Beanstalk Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni web Go utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, puoi continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure puoi utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Segui le step-by-step istruzioni QuickStart per Go per creare e distribuire un'applicazione web Hello World Go con l'EB CLI.

Argomenti

- QuickStart: distribuisci un'applicazione Go su Elastic Beanstalk
- Configurazione dell'ambiente di sviluppo Go per Elastic Beanstalk
- Utilizzo della piattaforma Go di Elastic Beanstalk

QuickStart: distribuisci un'applicazione Go su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione Go e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Passaggio 1: creare un'applicazione Go
- Fase 2: Implementa l'applicazione Go con l'EB CLI

- Fase 3: Esegui l'applicazione su Elastic Beanstalk
- Fase 4: pulizia
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Passaggio 1: creare un'applicazione Go

Crea una directory di progetto.

```
~$ mkdir eb-go
~$ cd eb-go
```

Successivamente, crea un'applicazione che distribuirai utilizzando Elastic Beanstalk. A questo proposito, creeremo un servizio Web RESTful "Hello World".

Questo esempio stampa un saluto personalizzato che varia in base al percorso utilizzato per accedere al servizio.

In questa directory, crea un file di testo denominato `application.go` con il seguente contenuto:

Example~/eb- .go go/application

```
package main

import (
 "fmt"
 "net/http"
)

func handler(w http.ResponseWriter, r *http.Request) {
 if r.URL.Path == "/" {
  fmt.Fprintf(w, "Hello World! Append a name to the URL to say hello. For example, use
 %s/Mary to say hello to Mary.", r.Host)
 } else {
  fmt.Fprintf(w, "Hello, %s!", r.URL.Path[1:])
 }
}

func main() {
 http.HandleFunc("/", handler)
 http.ListenAndServe(":5000", nil)
}
```

Fase 2: Implementa l'applicazione Go con l'EB CLI

Quindi, potrai creare l'ambiente applicativo e distribuire l'applicazione configurata con Elastic Beanstalk.

Creazione di un ambiente e distribuzione dell'applicazione Go

1. Inizializza il repository della CLI EB con il comando eb init.

   ```
   ~/eb-go$ eb init -p go go-tutorial --region us-east-2
   Application go-tutorial has been created.
   ```

   Questo comando crea un'applicazione denominata `go-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma Go.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

   ```
   ~/eb-go$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   2) [ Create new KeyPair ]
   ```

   Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

   ```
   ~/eb-go$ eb create go-env
   ```

   Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 3: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-go$ eb open
```

Complimenti! Hai distribuito un'applicazione Go con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 4: pulizia

Puoi chiudere il tuo ambiente quando hai finito di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-go$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.

- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

Dopo aver distribuito una o due applicazioni di esempio, è possibile iniziare lo sviluppo e l'esecuzione delle applicazioni Go in locale. Per supporto, consulta Configurazione dell'ambiente di sviluppo Go per Elastic Beanstalk.

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Configurazione dell'ambiente di sviluppo Go per Elastic Beanstalk

Questo argomento fornisce istruzioni per configurare un ambiente di sviluppo Go in cui testare l'applicazione localmente prima di distribuirla. AWS Elastic Beanstalk Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Installazione di Go

Per eseguire le applicazioni Go in locale, installare Go. Se non è necessaria una versione specifica, ottenere la versione più recente supportata da Elastic Beanstalk. Per un elenco di versioni supportate, consulta Gonel documento Platforms (Piattaforme)AWS Elastic Beanstalk .

Scarica Go all'indirizzo https://golang. org/doc/install.

Installazione dell' AWS SDK for Go

Se devi gestire AWS le risorse dall'interno dell'applicazione, installa l' AWS SDK for Go utilizzando il seguente comando.

```
$ go get github.com/aws/aws-sdk-go
```

Per ulteriori informazioni, consulta SDK AWS per Go.

Utilizzo della piattaforma Go di Elastic Beanstalk

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Go su Elastic Beanstalk.

AWS Elastic Beanstalk supporta una serie di rami della piattaforma per diverse versioni del linguaggio di programmazione Go. Vedi Go nel documento AWS Elastic Beanstalk Platforms per un elenco completo.

Per applicazioni Go semplici, sono disponibili due modi per distribuire l'applicazione:

- Fornisci un bundle di origine con un file sorgente nella radice chiamata `application.go` che contiene il pacchetto principale per la tua applicazione. Elastic Beanstalk crea il binario utilizzando il comando seguente:

  ```
  go build -o bin/application application.go
  ```

  Dopo che l'applicazione è stata creata, Elastic Beanstalk la avvia sulla porta 5000.
- Fornire un bundle di origine con un file binario chiamato `application`. Il file binario può essere situato nella radice del bundle di origine o nella directory `bin/` del bundle di origine. Se posizioni il file binario `application` in entrambi i percorsi, Elastic Beanstalk usa il file nella directory `bin/`.

  Elastic Beanstalk lancia questa applicazione sulla porta 5000.

In entrambi i casi, con le nostre filiali supportate della piattaforma Go, puoi anche fornire i requisiti del modulo in un file chiamato`go.mod`. Per ulteriori informazioni, consulta Migrazione a moduli Go nel blog Go.

Per applicazioni Go più complesse, sono disponibili due modi per distribuire l'applicazione:

- Fornisci un bundle di origine che includa i file di origine della tua applicazione, insieme a un Buildfile e a un Procfile. Il Buildfile include un comando per creare l'applicazione, mentre il Procfile include istruzioni per l'esecuzione dell'applicazione.
- Fornisci un bundle di origine che includa i file binari della tua applicazione, insieme a un Procfile. Il Procfile include istruzioni per l'esecuzione dell'applicazione.

La piattaforma Go include un server proxy che serve risorse statiche e inoltra il traffico all'applicazione. Puoi estendere o sostituire la configurazione proxy predefinita per scenari avanzati.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Configurazione dell'ambiente Go

Le impostazioni della piattaforma Go ti consentono di ottimizzare il comportamento delle tue istanze Amazon EC2 . Puoi modificare la configurazione dell'istanza Amazon dell'ambiente Elastic Beanstalk utilizzando EC2 la console Elastic Beanstalk.

Utilizza la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3 e configurare variabili che l'applicazione è in grado di leggere dall'ambiente.

Per configurare l'ambiente Go nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.

- Abilita la rotazione dei file di registro su Amazon S3: specifica se i file di log per le EC2 istanze Amazon dell'applicazione vengono copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Proprietà dell'ambiente

La sezione Environment Properties consente di specificare le impostazioni di configurazione dell'ambiente sulle EC2 istanze Amazon su cui è in esecuzione l'applicazione. Le proprietà di ambiente sono passate come coppie chiave-valore all'applicazione.

All'interno dell'ambiente Go in esecuzione in Elastic Beanstalk, le variabili di ambiente sono accessibili tramite la funzione `os.Getenv`. Ad esempio, è possibile leggere una proprietà denominata `API_ENDPOINT` su una variabile con il codice seguente:

```
endpoint := os.Getenv("API_ENDPOINT")
```

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazi dei nomi per la configurazione di Go

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

La piattaforma Go non definisce spazi dei nomi specifici della piattaforma. È possibile configurare il proxy per servire file statici utilizzando lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles`. Per informazioni dettagliate e un esempio, consulta the section called “File statici”.

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consultare Opzioni di configurazione.

La piattaforma Go di AMI Amazon Linux (precedente ad Amazon Linux 2)

Se l'ambiente Go di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Namespace di configurazione Go — Amazon Linux AMI () AL1

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

> **Note**
>
> Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.

Oltre agli spazi dei nomi supportati da tutte le piattaforme, la piattaforma Go di AMI Amazon Linux supporta uno spazio dei nomi di configurazione specifico della piattaforma. Lo spazio dei nomi `aws:elasticbeanstalk:container:golang:staticfiles` consente di definire le opzioni che mappano i percorsi nell'applicazione Web alle cartelle del bundle di origine dell'applicazione con contenuti statici.

Ad esempio, questo file di configurazione indica al server proxy di servire i file nella cartella `staticimages` al percorso `/images`:

Example.ebextensions/go-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:golang:staticfiles:
    /html: statichtml
    /images: staticimages
```

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Configurazione di comandi di avvio personalizzati con un Procfile su Elastic Beanstalk

Per specificare i comandi personalizzati per iniziare un'applicazione Go, includere un file chiamato `Procfile` nella radice del bundle di origine.

Per informazioni dettagliate sulla scrittura e l'utilizzo di un `Procfile` file, consulta. Buildfile e Procfile

Example Procfile

```
web: bin/server
queue_process: bin/queue_processor
foo: bin/fooapp
```

È necessario chiamare l'applicazione `web` principale ed elencarla come primo comando nel file `Procfile`. Elastic Beanstalk espone l'applicazione `web` principale nell'URL radice dell'ambiente, ad esempio `http://my-go-env.elasticbeanstalk.com`.

Elastic Beanstalk esegue anche qualsiasi applicazione il cui nome non include il prefisso `web_`, ma queste applicazioni non sono disponibili al di fuori dell'istanza.

Elastic Beanstalk si aspetta che i processi del `Procfile` vengano eseguiti continuamente. Elastic Beanstalk monitora queste applicazioni e riavvia qualsiasi processo che viene terminato. Per i processi con esecuzione breve, utilizza un comando Buildfile.

Utilizzo di un Procfile su AMI Amazon Linux (precedente ad Amazon Linux 2)

Se l'ambiente Go di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Passaggio delle porte — Amazon Linux AMI (AL1)

> **Note**
>
> Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.

Elastic Beanstalk configura il proxy nginx per inoltrare le richieste all'applicazione sul numero di porta specificato nella proprietà di ambiente `PORT` per l'applicazione. La tua applicazione deve sempre

essere in ascolto su tale porta. È possibile accedere a questa variabile all'interno dell'applicazione chiamando il metodo `os.Getenv("PORT")`.

Elastic Beanstalk usa il numero di porta specificato nella proprietà di ambiente `PORT` per la porta per la prima applicazione nel `Procfile`, quindi incrementa di 100 il numero di porta per ogni applicazione successiva nel `Procfile`. Se la proprietà `PORT` dell'ambiente non è impostata, Elastic Beanstalk utilizza 5000 per la porta iniziale.

Nell'esempio precedente, la proprietà di ambiente `PORT` per l'applicazione `web` è 5000, l'applicazione `queue_process` è 5100 e l'applicazione `foo` è 5200.

È possibile specificare la porta iniziale impostando l'opzione `PORT` con lo spazio dei nomi aws:elasticbeanstalk:application:environment, come mostrato nel seguente esempio.

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  PORT
    value:  <first_port_number>
```

Per ulteriori informazioni sull'impostazione delle proprietà di ambiente per l'applicazione, consulta Impostazioni opzioni.

Compilazione e configurazione personalizzate con un Buildfile su Elastic Beanstalk

Per specificare il comando personalizzato di creazione e configurazione di un'applicazione Go, includi un file denominato `Buildfile` nella radice del bundle di origine. Per il nome del file è prevista la distinzione tra maiuscole e minuscole. Utilizza il seguente formato per il `Buildfile`:

```
<process_name>: <command>
```

Il comando nel `Buildfile` deve corrispondere alla seguente espressione regolare: `^[A-Za-z0-9_]+:\s*.+$`.

Elastic Beanstalk non monitora l'applicazione eseguita con un `Buildfile`. Usa un `Buildfile` per comandi che vengono eseguiti per brevi periodi e terminano dopo aver completato le proprie attività. Per i processi di applicazioni a lungo termine da non terminare, utilizza il Procfile.

Nel seguente esempio di `Buildfile`, `build.sh` è uno script di shell che si trova nella radice del bundle di origine:

```
make: ./build.sh
```

Tutti i percorsi nel `Buildfile` sono relativi alla radice del bundle di origine. Se sai in anticipo dove risiedono i file nell'istanza, puoi includere percorsi assoluti nel `Buildfile`.

Configurazione del server proxy

Elastic Beanstalk usa nginx come proxy inverso per mappare l'applicazione al load balancer Elastic Load Balancing sulla porta 80. Elastic Beanstalk fornisce una configurazione nginx predefinita che puoi estendere o ignorare completamente a favore di quella personalizzata.

Per impostazione predefinita, Elastic Beanstalk configura il proxy nginx per inoltrare le richieste alla tua applicazione sulla porta 5000. È possibile sostituire la porta predefinita impostando la proprietà dell'ambiente `PORT` sulla porta di ascolto dell'applicazione principale.

> **Note**
>
> La porta di ascolto della tua applicazione non influisce sulla porta che il server nginx ascolta per ricevere richieste dal load balancer.

Configurazione del server proxy sulla versione della piattaforma

Tutte le AL2 piattaforme AL2 023/ supportano una funzionalità di configurazione proxy uniforme. Per ulteriori informazioni sulla configurazione del server proxy sulle versioni della piattaforma che eseguono AL2 AL2 023/, consulta. Configurazione del proxy inverso

Configurazione del proxy su AMI Amazon Linux (precedente ad Amazon Linux 2)

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e

completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Se l'ambiente Go di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni presenti in questa sezione.

Estensione e sovrascrittura della configurazione proxy predefinita — Amazon Linux AMI () AL1

Elastic Beanstalk usa nginx come proxy inverso per mappare l'applicazione al load balancer sulla porta 80. Se desideri fornire una configurazione personalizzata per nginx, puoi sostituire la configurazione predefinita fornita da Elastic Beanstalk includendo il file `.ebextensions/nginx/nginx.conf` nel bundle di origine. Se il file è presente, Elastic Beanstalk lo utilizza al posto del file di configurazione nginx predefinito.

Se desideri includere direttive oltre a quelle nel blocco `nginx.conf` `http`, è anche possibile fornire ulteriori file di configurazione nella directory `.ebextensions/nginx/conf.d/` del bundle di origine. Tutti i file contenuti in questa directory devono avere l'estensione `.conf`.

Per trarre vantaggio dalle funzionalità offerte da Elastic Beanstalk, ad esempio Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk, le mappature automatiche delle applicazioni e i file statici, è necessario includere la riga seguente nel blocco `server` del file di configurazione nginx:

```
include conf.d/elasticbeanstalk/*.conf;
```

Distribuzione di applicazioni Java con Elastic Beanstalk

Questo capitolo fornisce istruzioni per configurare e distribuire le applicazioni Java su. AWS Elastic Beanstalk Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni Web Java utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, puoi continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure puoi utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Segui le step-by-step istruzioni QuickStart per Java per creare e distribuire un'applicazione web Java Hello World con l'EB CLI. Se sei interessato alle step-by-step istruzioni per creare una semplice applicazione Hello World Java JSP da distribuire con l'EB CLI sulla nostra piattaforma basata su Tomcat, prova il. QuickStart per Java su Tomcat

La piattaforma Java si ramifica

AWS Elastic Beanstalk supporta due piattaforme per applicazioni Java.

- Tomcat: una piattaforma basata su Apache Tomcat, un contenitore web open source per applicazioni che utilizzano servlet Java e JavaServer Pages (JSPs) per soddisfare le richieste HTTP. Tomcat facilita lo sviluppo di applicazioni Web offrendo multithreading, configurazione di sicurezza dichiarativa e uso esteso della personalizzazione. Elastic Beanstalk include rami di piattaforma per ciascuna delle attuali versioni principali di Tomcat. Per ulteriori informazioni, consulta Piattaforma Tomcat.
- Java SE: una piattaforma per applicazioni che non utilizzano un contenitore Web o ne utilizzano uno diverso da Tomcat, come Jetty o. GlassFish Puoi includere qualsiasi libreria Java Archives (JARs) utilizzata dall'applicazione nel pacchetto sorgente che distribuisci su Elastic Beanstalk. Per ulteriori informazioni, consulta Piattaforma Java SE.

Le filiali recenti delle piattaforme Tomcat e Java SE sono basate su Amazon Linux 2 e versioni successive e utilizzano Corretto, la distribuzione AWS Java SE. I nomi di questi rami della piattaforma includono la parola Corretto anziché Java.

Per un elenco delle versioni correnti della piattaforma, consulta Tomcat e Java SE nella guida Piattaforme AWS Elastic Beanstalk .

AWS strumenti

AWS fornisce diversi strumenti per lavorare con Java ed Elastic Beanstalk. Indipendentemente dal ramo della piattaforma scelto, puoi utilizzare l'AWS SDK for Java per utilizzare AWS altri servizi dall'interno della tua applicazione Java. L' AWS SDK for Java è un insieme di librerie che consentono di AWS APIs utilizzare il codice dell'applicazione senza scrivere da zero le chiamate HTTP non elaborate.

Se preferisci gestire le tue applicazioni dalla riga di comando, installa l'Elastic Beanstalk Command Line Interface (EB CLI) e usala per creare, monitorare e gestire gli ambienti Elastic Beanstalk. Se si eseguono più ambienti per l'applicazione, la CLI EB si integra con Git per consentire di associare ciascuno di essi a un ramo Git diverso.

Argomenti

- QuickStart: distribuzione di un'applicazione Java su Elastic Beanstalk
- QuickStart: Distribuisci un'applicazione web Java JSP per Tomcat su Elastic Beanstalk
- Configurazione dell'ambiente di sviluppo Java
- Altri esempi di applicazioni e tutorial di Elastic Beanstalk per Java
- Utilizzo della piattaforma Tomcat di Elastic Beanstalk
- Utilizzo della piattaforma Java SE di Elastic Beanstalk
- Aggiungere un'istanza Amazon RDS DB all'ambiente Java Elastic Beanstalk
- Strumenti e risorse Java

QuickStart: distribuzione di un'applicazione Java su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione Java e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Passaggio 1: creare un'applicazione Java
- Fase 2: Esegui l'applicazione localmente
- Fase 3: Distribuire l'applicazione Java con l'EB CLI
- Fase 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Java e Maven

Se Amazon Corretto non è installato sul tuo computer locale, puoi installarlo seguendo le istruzioni di installazione nella Amazon Corretto User Guide.

Verifica l'installazione di Java eseguendo il comando seguente.

```
~$ java -version
```

Questo tutorial utilizza Maven. Segui le istruzioni per il download e l'installazione sul sito Web di Apache Maven Project. Per ulteriori informazioni su Maven, consulta il Maven Users Center sul sito web del progetto Apache Maven.

Verifica l'installazione di Maven eseguendo il seguente comando.

```
~$ mvn -v
```

Passaggio 1: creare un'applicazione Java

Crea una directory di progetto.

```
~$ mkdir eb-java
~$ cd eb-java
```

Successivamente, crea un'applicazione che distribuirai utilizzando Elastic Beanstalk. A questo proposito, creeremo un servizio Web RESTful "Hello World".

Questo esempio utilizza il framework Spring Boot. Questa applicazione apre un listener sulla porta 5000. Elastic Beanstalk inoltra le richieste all'applicazione sulla porta 5000 per impostazione predefinita.

Crea i seguenti file:

Questo file crea una semplice applicazione Spring Boot.

Example~/eb-java/src///main/javacom/exampleApplication.java

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Questo file crea una mappatura che restituisce una stringa che definiamo qui.

Example~/eb-java/src///main/javacom/exampleController.java

```
package com.example;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class Controller {

    @GetMapping("/")
    public String index() {
       return "Hello Elastic Beanstalk!";
    }
}
```

Questo file definisce la configurazione del progetto Maven.

Example~/eb- .xml java/pom

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
```

```
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.3</version>
  </parent>

  <groupId>com.example</groupId>
  <artifactId>BeanstalkJavaExample</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <java.version>21</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-rest</artifactId>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>

</project>
```

Questo file di proprietà sostituisce la porta predefinita impostata su 5000. Questa è la porta predefinita a cui Elastic Beanstalk invia il traffico per le applicazioni Java.

Example~/eb- .properties java/application

```
server.port=5000
```

Fase 2: Esegui l'applicazione localmente

Package della tua applicazione con il seguente comando:

```
~/eb-java$ mvn clean package
```

Esegui l'applicazione localmente con il seguente comando:

```
~/eb-java$ java -jar target/BeanstalkJavaExample-1.0-SNAPSHOT.jar
```

Mentre l'applicazione è in esecuzione, accedi al `http://127.0.0.1:5000/` menu nel tuo browser. Dovresti vedere il testo «Hello Elastic Beanstalk!».

Fase 3: Distribuire l'applicazione Java con l'EB CLI

Prima di distribuire l'applicazione Java su Elastic Beanstalk, puliamo l'applicazione di compilazione dalla directory e creiamo un Buildfile e un Procfile per controllare come l'applicazione viene creata ed eseguita nell'ambiente Elastic Beanstalk.

Per preparare e configurare la distribuzione delle applicazioni

1. Pulisci l'applicazione integrata.

   ```
   ~/eb-java$ mvn clean
   ```

2. Crea un `Buildfile`.

   Example~/eb- java/Buildfile

   ```
   build: mvn clean package
   ```

   Questo `Buildfile` specifica il comando usato per creare l'applicazione. Se non includi un file `Buildfile` per un'applicazione Java, Elastic Beanstalk non tenterà di creare la tua applicazione.

3. Crea un `Procfile`.

   Example~/eb- java/Procfile

   ```
   web: java -jar target/BeanstalkJavaExample-1.0-SNAPSHOT.jar
   ```

Questo `Procfile` specifica il comando usato per eseguire l'applicazione. Se non includi un file `Procfile` per un'applicazione Java, Elastic Beanstalk presuppone che ci sia un file JAR nella radice del pacchetto sorgente e tenta di eseguirlo con il comando. `java -jar`

Ora che hai impostato i file di configurazione per creare e avviare l'applicazione, sei pronto per distribuirla.

Per creare un ambiente e distribuire l'applicazione Java

1. Inizializza il repository della CLI EB con il comando eb init.

```
~/eb-java eb init -p corretto java-tutorial --region us-east-2

Application java-tutorial has been created.
```

   Questo comando crea un'applicazione denominata `java-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma Java.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

```
~/eb-java$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

   Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

```
~/eb-java$ eb create java-env
```

   Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-java eb open
```

Complimenti! Hai distribuito un'applicazione Java con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 5: rimozione

Puoi chiudere il tuo ambiente quando hai finito di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-java$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.

- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. `subdomain region`.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

> **Prova altri tutorial**
>
> Se desideri provare altri tutorial con diverse applicazioni di esempio, consulta. Esempi di applicazioni e tutorial

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire applicazioni Java localmente, vedi. Configurazione dell'ambiente di sviluppo Java

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

QuickStart: Distribuisci un'applicazione web Java JSP per Tomcat su Elastic Beanstalk

Questo tutorial illustra il processo di creazione di una semplice applicazione web Java utilizzando JavaServer Pages (JSP). Se desideri raggruppare più applicazioni Web sotto forma di file WAR in un unico ambiente Elastic Beanstalk, consulta. Raggruppamento di più file WAR per ambienti Tomcat

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Fase 1: Creare un'applicazione Java JSP
- Fase 2: Distribuire l'applicazione Java JSP con l'EB CLI
- Fase 3: Esegui l'applicazione su Elastic Beanstalk
- Fase 4: pulizia
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Fase 1: Creare un'applicazione Java JSP

Crea una directory di progetto.

```
~$ mkdir eb-tomcat
~$ cd eb-tomcat
```

Successivamente, crea un'applicazione che distribuirai utilizzando Elastic Beanstalk. Creeremo un'applicazione web «Hello World».

Crea un semplice file JSP denominato`index.jsp`.

Example~/eb- .jsp tomcat/index

```
<html>
  <body>
    <%out.println("Hello Elastic Beanstalk!");%>
  </body>
</html>
```

Fase 2: Distribuire l'applicazione Java JSP con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire l'applicazione Java JSP

1. Inizializza il repository della CLI EB con il comando eb init.

   ```
   ~/eb-tomcat$ eb init -p tomcat tomcat-tutorial --region us-east-2
   ```

   Questo comando crea un'applicazione denominata `tomcat-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma Tomcat.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

   ```
   ~/eb-go$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   2) [ Create new KeyPair ]
   ```

   Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

   ```
   ~/eb-tomcat$ eb create tomcat-env
   ```

Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 3: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-tomcat$ eb open
```

Complimenti! Hai distribuito un'applicazione Java JSP con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 4: pulizia

Puoi chiudere il tuo ambiente quando finisci di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-tomcat$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.

- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. `subdomain region`.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

> **Prova altri tutorial**
>
> Se desideri provare altri tutorial con diverse applicazioni di esempio, consulta. Esempi di applicazioni e tutorial

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire applicazioni Java in un contenitore web Tomcat locale, vedi. Configurazione dell'ambiente di sviluppo Java

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Configurazione dell'ambiente di sviluppo Java

Questo argomento fornisce istruzioni per configurare un ambiente di sviluppo Java per testare l'applicazione localmente prima di distribuirla AWS Elastic Beanstalk. Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Sections

- Installazione di Java Development Kit
- Installazione di un container Web
- Download di librerie
- Installazione dell' AWS SDK for Java
- Installazione di un IDE o un editor di testo

Installazione di Java Development Kit

Installa Java Development Kit (JDK). Se non hai preferenze, scarica la versione più recente Scarica il JDK dal sito oracle.com.

JDK include il compilatore Java, con cui è possibile creare file di origine in file di classe eseguibili su un server Web Elastic Beanstalk.

Installazione di un container Web

Se non disponi già di un altro contenitore o framework Web, installa una versione di Tomcat supportata da Elastic Beanstalk per il tuo sistema operativo Amazon Linux. Per un elenco delle versioni correnti di Apache Tomcat supportate da Elastic Beanstalk, consulta Tomcat nel documento Piattaforme.AWS Elastic Beanstalk Scarica la versione di Tomcat adatta al tuo ambiente dal sito Web di Apache Tomcat.

Download di librerie

Le piattaforme Elastic Beanstalk includono alcune librerie per impostazione predefinita. Scarica le librerie che l'applicazione utilizzerà e salvale nella tua cartella di progetto per una distribuzione nel bundle di origine dell'applicazione.

Se hai installato Tomcat localmente, puoi copiare le librerie API servlet e JavaServer Pages (JSP) dalla cartella di installazione. Se esegui la distribuzione in una versione della piattaforma Tomcat,

questi file non devono essere nel bundle di origine, ma in `classpath` allo scopo di compilare qualsiasi classe che li utilizza.

JUnit, Google Guava e Apache Commons forniscono diverse librerie utili. Visita le relative home page per ulteriori informazioni:

- Scarica JUnit
- Scarica Google Guava
- Scarica Apache Commons

Installazione dell' AWS SDK for Java

Se devi gestire AWS le risorse dall'interno dell'applicazione, installa l' AWS SDK for Java. Ad esempio, con AWS SDK per Java, è possibile utilizzare Amazon DynamoDB (DynamoDB) per condividere gli stati di sessione delle applicazioni Apache Tomcat su più server Web. Per ulteriori informazioni, consulta Gestire lo stato della sessione Tomcat con Amazon DynamoDB nella documentazione AWS SDK for Java.

Visita la home page dell'SDK AWS per Java per ulteriori informazioni e istruzioni di installazione.

Installazione di un IDE o un editor di testo

Gli ambienti di sviluppo integrati (IDEs) offrono un'ampia gamma di funzionalità che facilitano lo sviluppo di applicazioni. Se non hai utilizzato un IDE per lo sviluppo Java, prova Eclipse e IntelliJ e determina qual è il più appropriato.

- Installazione dell'IDE di Eclipse per sviluppatori Java EE
- Installazione di IntelliJ

Un IDE potrebbe aggiungere file alla tua cartella di progetto di cui non intendi eseguire il commit nel controllo del codice sorgente. Per impedire il commit di questi file nel controllo del codice sorgente, utilizza `.gitignore` o uno strumento di controllo del codice sorgente equivalente.

Se vuoi solo iniziare a scrivere codice e non hai bisogno di tutte le caratteristiche di un IDE, prendi in considerazione l'installazione di Sublime Text.

> **Note**
>
> Il 31 maggio 2023, AWS Toolkit for Eclipseha raggiunto la fine del ciclo di vita e non è più supportato da AWS. Per ulteriori dettagli sulla fine del ciclo di vita di AWS Toolkit for Eclipse, consultate il file README.md nel repository. AWS Toolkit for Eclipse GitHub

Altri esempi di applicazioni e tutorial di Elastic Beanstalk per Java

Questa sezione fornisce applicazioni e tutorial aggiuntivi. QuickStart per Java su TomcatGli argomenti QuickStart per Java descritti in precedenza in questo argomento illustrano l'avvio di un'applicazione Java di esempio con l'EB CLI.

Per iniziare a utilizzare le applicazioni Java AWS Elastic Beanstalk, è sufficiente un pacchetto di sorgenti di applicazioni da caricare come prima versione dell'applicazione e da distribuire in un ambiente. Quando crei un ambiente, Elastic Beanstalk alloca tutte le risorse AWS necessarie per eseguire un'applicazione Web scalabile.

Avvio di un ambiente con un'applicazione Java di esempio

Elastic Beanstalk fornisce applicazioni di esempio a pagina singola per ogni piattaforma, nonché esempi più complessi che mostrano l'uso di risorse AWS aggiuntive come Amazon RDS e funzionalità specifiche del linguaggio o della piattaforma e. APIs

Gli esempi di singole pagine hanno lo stesso codice che ottieni al momento della creazione di un ambiente senza fornire il tuo codice sorgente. Gli esempi più complessi sono ospitati su GitHub e potrebbe essere necessario compilarli o crearli prima della distribuzione in un ambiente Elastic Beanstalk.

Esempi

Name	Versioni supportate	Tipo di ambie	Origine	Descrizione
Tomc (pagi singo	Tutti i rami della piattafor ma Tomcat con Corretto	Serve Web Worke	tomcat.zi p	Applicazione Web Tomcat con un'unica pagina (`index.jsp`) configurata per essere visualizzata nel sito Web radice.

Name	Versioni supportate	Tipo di ambie	Origine	Descrizione
				Per gli ambienti lavoratore, questo esempio include un file `cron.yaml` che configura un'attività pianificata per chiamare `scheduled.jsp` una volta al minuto. Quando viene chiamato `scheduled.jsp`, viene eseguita la scrittura in un file di log in `/tmp/sample-app.log`. Infine, un file di configurazione viene incluso in `.ebextensions` che copia i log da `/tmp/` per le posizioni lette da Elastic Beanstalk al momento della richiesta dei log di ambiente. Se abiliti l'integrazione X-Ray in un ambiente con questo esempio in esecuzione, l'applicazione mostra contenuti aggiuntivi riguardanti X-Ray e offre un'opzione per generare informazioni di debug che puoi visualizzare nella console X-Ray.
Corre (pagi singo	Corretto 11 Corretto 8	Serve Web	corretto.zip	Applicazione Corretto con i file di configurazione `Buildfile` e `Procfile`. Se abiliti l'integrazione X-Ray in un ambiente con questo esempio in esecuzione, l'applicazione mostra contenuti aggiuntivi riguardanti X-Ray e offre un'opzione per generare informazioni di debug che puoi visualizzare nella console X-Ray.

Name	Versioni supportate	Tipo di ambie	Origine	Descrizione
Score	Java 8	Serve Web	Clona il repositor y su .com GitHub	Scorekeep è un'API RESTful web che utilizza il framework Spring per fornire un'interfaccia per la creazione e la gestione di utenti, sessioni e giochi. L'API è integrata con un'app Web Angular 1.5 che consuma l'API sull'HTTP. L'applicazione utilizza caratteristiche della piattaforma Java SE per scaricare dipendenze e creare on-instance, riducendo al minimo le dimensioni del bundle di origine. L'applicazione include anche file di configurazione nginx che sovrascrivono la configurazione predefini ta per servire l'app Web di front-end in modo statico sulla porta 80 tramite il proxy, e instradare le richieste su percorsi in `/ api` all'API in esecuzione su `localhost :5000` . Scorekeep include anche un ramo `xray` che mostra come approntare un'applic azione Java per l'uso con AWS X-Ray. Mostra la strumentazione delle richieste HTTP in entrata con un filtro servlet, la strumentazione client AWS SDK automatic a e manuale, la configurazione del registrat ore e la strumentazione delle richieste HTTP in uscita e dei client SQL. Consulta il file Readme per istruzioni oppure utilizza il Tutorial sulle nozioni di

Name	Versioni supportate	Tipo di ambie	Origine	Descrizione
				base di AWS X-Ray per provare l'applica zione con X-Ray.
Does it Have Snak	Tomcat 8 con Java 8	Serve Web	Clona il repositor y su .com GitHub	Ha serpenti? è un'applicazione web Tomcat che mostra l'uso dei file di configurazione Elastic Beanstalk, Amazon RDS, JDBC, PostgreSQL, JSPs Servlets, Simple Tag Support, Tag Files, Log4J, Bootstrap e Jackson. Il codice sorgente per questo progetto include uno script di build minimo che compila i servlet e i modelli in file delle classi e crea un pacchetto dei file richiesti in un archivio Web che puoi distribuire in un ambiente Elastic Beanstalk. Vedi il file Readme nell'archivio del progetto per le istruzioni complete.
Locus Load Gene	Java 8	Serve Web	GitHubClo na il repositor y su .com	Applicazione Web che puoi utilizzare per eseguire il test di carico di un'altra applicazi one Web in esecuzione in un ambiente Elastic Beanstalk diverso. Mostra l'uso dei file `Buildfile` e `Procfile`, DynamoDB e Locust, uno strumento di test del carico open source.

Scarica una delle applicazioni di esempio e distribuiscila su Elastic Beanstalk seguendo questi passaggi:

Per avviare un ambiente con un'applicazione (console)

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Nel riquadro di navigazione, scegliere Applications (Applicazioni). Seleziona un'applicazione esistente nell'elenco. Puoi anche scegliere di crearne una seguendo le istruzioni riportate inGestione delle applicazioni .
3. Nella pagina di panoramica dell'applicazione, scegli Crea ambiente.

   Verrà avviata la procedura guidata Create environment (Creazione dell'ambiente). La procedura guidata fornisce una procedura che consente di creare un nuovo ambiente.
4. Per il livello Ambiente, scegli il livello Ambiente server Webo Ambiente di lavoro. Non è possibile modificare un piano di un ambiente dopo la creazione.

   > **Note**
   >
   > La piattaforma .NET su Windows Server non supporta il piano di ambiente operatore.

   I campi di informazioni sull'applicazione sono predefiniti, in base all'applicazione scelta in precedenza.

   Nel gruppo di informazioni sull'ambiente, il valore predefinito del nome dell'ambiente è basato sul nome dell'applicazione. Se preferisci un nome di ambiente diverso, puoi inserire un altro valore nel campo. Facoltativamente, puoi inserire un nome di dominio; in caso contrario Elastic Beanstalk genera automaticamente un valore. Facoltativamente, puoi anche inserire una descrizione dell'ambiente.
5. Per Piattaforma, selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.

   > **Note**
   >
   > Elastic Beanstalk supporta più versioni per la maggior parte delle piattaforme elencate. Per impostazione predefinita, la console seleziona la versione consigliata per il ramo della piattaforma e per la piattaforma di scelta. Se l'applicazione richiede una versione diversa, puoi selezionarla qui. Per informazioni sulle versioni della piattaforma supportate, consulta the section called “Piattaforme supportate”.
6. Per il codice dell'applicazione, sono disponibili diverse scelte per procedere.

- Per avviare l'applicazione di esempio predefinita senza fornire il codice sorgente, scegliete Applicazione di esempio. Questa azione sceglie l'applicazione a pagina singola fornita da Elastic Beanstalk per la piattaforma selezionata in precedenza.
- Se hai scaricato un'applicazione di esempio da questa guida o disponi del tuo codice sorgente per un'applicazione, procedi nel seguente modo.

  a. Seleziona Carica il tuo codice.

  b. Quindi scegli File locale, quindi in Carica applicazione, seleziona Scegli file.

  c. Il sistema operativo del computer client ti presenterà un'interfaccia per selezionare il file locale che hai scaricato. Seleziona il file del pacchetto sorgente e continua.

7. La scelta delle impostazioni predefinite dipende dallo scopo dell'ambiente.

   - Se stai creando un ambiente di esempio per conoscere Elastic Beanstalk o un ambiente di sviluppo, scegli Istanza singola (idoneo al piano gratuito).
   - Se stai creando un ambiente di produzione o un ambiente per saperne di più sul bilanciamento del carico, scegli una delle opzioni di alta disponibilità.

8. Scegli Next (Successivo).

Per configurare l'accesso al servizio

Successivamente, sono necessari due ruoli. Un ruolo di servizio consente a Elastic Beanstalk di monitorare le istanze EC2 e aggiornare la piattaforma dell'ambiente. Un ruolo di profilo di istanza EC2 consente attività come la scrittura di log e l'interazione con altri servizi.

Per creare o selezionare il ruolo Service

1. Se hai già creato un ruolo di servizio e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa Ruolo di servizio e salta il resto di questi passaggi per creare un ruolo di servizio.
2. Se non vedi alcun valore elencato per il ruolo di servizio o desideri crearne uno nuovo, continua con i passaggi successivi.
3. Per Ruolo di servizio, scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Environment.
6. Scegli Next (Successivo).

7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`
8. Scegli Crea ruolo.
9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il ruolo di servizio appena creato.

Per creare o selezionare un profilo di istanza EC2

1. Se hai già creato un profilo di istanza EC2 e desideri sceglierne uno esistente, seleziona il valore dal menu a discesa del profilo dell'istanza EC2 e salta il resto di questi passaggi per creare un profilo di istanza EC2.
2. Se non vedi alcun valore elencato per il profilo dell'istanza EC2 o desideri crearne uno nuovo, continua con i passaggi successivi.
3. Scegli Crea ruolo.
4. Per Tipo di entità attendibile, scegli Servizio AWS .
5. Per il caso d'uso, scegli Elastic Beanstalk — Compute.
6. Scegli Next (Successivo).
7. Verifica che i criteri di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
8. Scegli Crea ruolo.
9. Torna alla scheda Configura l'accesso al servizio, aggiorna l'elenco, quindi seleziona il profilo di istanza EC2 appena creato.

Per completare la configurazione e la creazione dell'applicazione

1. (Facoltativo) Se in precedenza hai creato una coppia di chiavi EC2, puoi selezionarla dal menu a discesa del campo delle coppie di chiavi EC2. Puoi utilizzarlo per accedere in modo sicuro all'istanza Amazon EC2 fornita da Elastic Beanstalk per la tua applicazione. Se salti questo

passaggio, puoi sempre creare e assegnare una coppia di key pair EC2 dopo la creazione dell'ambiente. Per ulteriori informazioni, consulta Coppia di chiavi EC2.

2. Scegli Skip to Review (Passa alla revisione) nella pagina Configure service access (Configura l'accesso al servizio).
3. La pagina Review (Revisione) mostra un riepilogo di tutte le tue scelte.

   Per personalizzare ulteriormente l'ambiente, scegli Edit (Modifica) accanto al passaggio che include tutti gli elementi che desideri configurare. Le seguenti opzioni possono essere impostate solo durante la creazione dell'ambiente:

   - Nome ambiente
   - Nome dominio
   - Versione della piattaforma
   - Processore
   - Nuovo tipo di load balancer
   - Livello

   Le seguenti impostazioni possono essere modificate dopo la creazione dell'ambiente, ma richiedono nuove istanze o altre risorse per il provisioning e i tempi di applicazione possono essere lunghi:

   - Tipo di istanza, volume root, key pair e ruolo AWS Identity and Access Management (IAM)
   - Database interno Amazon RDS
   - VPC

   Per ulteriori dettagli sulle impostazioni disponibili, consulta Procedura guidata per la creazione del nuovo ambiente.
4. Scegli Submit (Invia) nella parte inferiore della pagina per inizializzare la creazione del nuovo ambiente.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione completamente diversa in qualsiasi momento. La distribuzione di

una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2.

Se, dopo aver distribuito una o due applicazioni di esempio, ti senti pronto per iniziare a sviluppare e a eseguire applicazioni Java in locale, consulta la sezione successiva per impostare un ambiente di sviluppo Java con tutti gli strumenti e le librerie di cui hai bisogno.

Utilizzo della piattaforma Tomcat di Elastic Beanstalk

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Java eseguite sulla piattaforma Elastic Beanstalk Tomcat.

La piattaforma AWS Elastic Beanstalk Tomcat è un insieme di versioni della piattaforma per applicazioni Web Java che possono essere eseguite in un contenitore Web Tomcat. Tomcat viene eseguito dietro un server proxy nginx. Ogni ramo della piattaforma corrisponde a una versione principale di Tomcat.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

La piattaforma Tomcat di Elastic Beanstalk include un proxy inverso che inoltra le richieste alla tua applicazione. È possibile utilizzare le opzioni di configurazione per configurare il server proxy in modo che possa servire asset statici da una cartella nel codice sorgente, per ridurre il carico sull'applicazione. Per gli scenari avanzati, puoi includere file `.conf` personalizzati nel bundle di origine per estendere la configurazione proxy di Elastic Beanstalk o sovrascriverla completamente.

> **Note**
>
> Elastic Beanstalk supporta nginx (impostazione predefinita) e Apache HTTP Server come server proxy sulla piattaforma Tomcat. Se il tuo ambiente Tomcat di Elastic Beanstalk utilizza un ramo della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), hai anche la

possibilità di utilizzare Apache HTTP Server versione 2.2. Apache (versione più recente) è l'impostazione predefinita su questi rami della piattaforma più vecchi.
Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Devi creare un pacchetto di applicazioni Java in un file WAR con una struttura specifica. Per informazioni sulla struttura necessaria e sulla relazione con la struttura della tua directory di progetto, consulta Strutturazione della cartella di progetto.

Per eseguire più applicazioni sullo stesso server Web, puoi raggruppare più file WAR in un singolo bundle di origine. Ogni applicazione in un bundle di origine con più file WAR viene eseguita nel percorso root (`ROOT.war` eseguito in *`myapp`*`.elasticbeanstalk.com/`) o in un percorso immediatamente sottostante (`app2.war` eseguito in *`myapp`*`.elasticbeanstalk.com/`*`app2`*`/`), come indicato dal nome del file WAR. In un bundle di origine con un solo file WAR, l'applicazione viene eseguita sempre nel percorso root.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Argomenti

- Configurazione dell'ambiente Tomcat
- Spazio dei nomi della configurazione Tomcat
- Raggruppamento di più file WAR per ambienti Tomcat
- Strutturazione della cartella di progetto
- Configurazione del server proxy

Configurazione dell'ambiente Tomcat

La piattaforma Tomcat di Elastic Beanstalk offre alcune opzioni specifiche della piattaforma in aggiunta alle opzioni standard disponibili in tutte le piattaforme. Queste opzioni permettono di configurare Java Virtual Machine (JVM), in esecuzione nei server Web dell'ambiente, e di definire proprietà di sistema che forniscono le stringhe di configurazione delle informazioni all'applicazione.

Puoi utilizzare la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3 e configurare variabili che l'applicazione è in grado di leggere dall'ambiente.

Per configurare l'ambiente Tomcat nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni del contenitore

È possibile specificare le seguenti opzioni specifiche della piattaforma:

- Proxy server (Server proxy): il server proxy da utilizzare nelle istanze dell'ambiente. Per impostazione predefinita, viene utilizzato nginx.

Opzioni del container JVM

La dimensione heap nella Java Virtual Machine (JVM) determina il numero di oggetti che la tua applicazione può creare in memoria prima della garbage collection. Puoi modificare le opzioni Dimensione heap iniziale di JVM (`-Xms option`) e Dimensione heap JVM massima (`-Xmx`). Una dimensione heap iniziale più grande consente la creazione di un maggior numero di oggetti prima della garbage collection, ma il garbage collector impiegherà più tempo per compattare l'heap. La dimensione heap massima specifica la quantità massima di memoria che JVM può allocare durante l'espansione dell'heap nel quadro di un'attività intensa.

> **Note**
>
> La memoria disponibile dipende dal tipo di istanza Amazon EC2. Per ulteriori informazioni sui tipi di istanza EC2 disponibili per l'ambiente Elastic Beanstalk, consulta Tipi di istanza nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

La generazione permanente è una sezione dell'heap JVM che memorizza le definizioni di classi e i metadati associati. Per modificare la dimensione della generazione permanente, digita la nuova dimensione nell'opzione Maximum JVM PermGen Size (`-XX:MaxPermSize`). Questa impostazione è valida solo per Java 7 e versioni precedenti. Questa opzione era obsoleta in JDK 8 e sostituita dall'opzione Size (). MaxMetaspace `-XX:MaxMetaspaceSize`

> **Important**
>
> JDK 17 ha rimosso il supporto dell'opzione Java `-XX:MaxPermSize`. L'utilizzo di questa opzione con un ambiente in esecuzione su una ramificazione della piattaforma Elastic Beanstalk con Corretto 17 genererà un errore. Elastic Beanstalk ha rilasciato la prima ramificazione della piattaforma che esegue Tomcat con Corretto 17 il 13 luglio 2023. Per ulteriori informazioni, consulta le risorse seguenti.
>
> - Sito Web della documentazione di Oracle Java: Removed Java Options
> - Sito Web della documentazione di Oracle Java: sezione Class Metadata in Other Considerations

Per ulteriori informazioni sulle piattaforme Elastic Beanstalk e sui relativi componenti, consulta la pagina Piattaforme supportate nella Guida alle piattaforme AWS Elastic Beanstalk .

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Enable log file rotation to Amazon S3 (Abilita rotazione file di log in Amazon S3): specifica se i file di log per le istanze Amazon EC2 dell'applicazione debbano essere copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Proprietà dell'ambiente

Nella sezione Environment Properties (Proprietà ambiente) puoi specificare le impostazioni di configurazione dell'ambiente nelle istanze Amazon EC2 che eseguono l'applicazione. Le proprietà di ambiente sono passate come coppie chiave-valore all'applicazione.

La piattaforma Tomcat definisce una proprietà segnaposto denominata `JDBC_CONNECTION_STRING` per gli ambienti Tomcat, utilizzata per passare una stringa di connessione a un database esterno.

> **Note**
>
> Se colleghi un'istanza database RDS all'ambiente, crea dinamicamente la stringa di connessione JDBC dalle proprietà dell'ambiente Amazon Relational Database Service (Amazon RDS) fornite da Elastic Beanstalk. Utilizza JDBC_CONNECTION_STRING solo per le istanze database il cui provisioning non viene effettuato mediante Elastic Beanstalk.
> Per ulteriori informazioni sull'uso di Amazon RDS con l'applicazione Java, consulta Aggiungere un'istanza Amazon RDS DB all'ambiente Java Elastic Beanstalk.

Per le versioni della piattaforma Tomcat rilasciate prima del 26 marzo 2025, le variabili di ambiente sono accessibili tramite. `System.getProperty()` Ad esempio, è possibile leggere una proprietà denominata `API_ENDPOINT` da una variabile con il codice seguente.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

Le versioni della piattaforma Tomcat rilasciate a partire dal 26 marzo 2025 possono essere utilizzate anche `System.getenv` per accedere a variabili di ambiente in testo non crittografato. È possibile

continuare a utilizzare per accedere alle variabili di ambiente in testo `System.getProperty` non crittografato. Tuttavia, le variabili di ambiente memorizzate come segrete sono disponibili solo utilizzando. `System.getenv` Ad esempio, è possibile leggere una variabile di ambiente denominata `API_KEY` con il codice seguente.

```
String apiKey = System.getenv("API_KEY");
```

> **Important**
>
> L'aggiunta dell'`System.getenv()`accesso per le variabili di ambiente nelle versioni della piattaforma Tomcat rilasciate il o dopo il 26 marzo 2025 può causare comportamenti imprevisti nelle applicazioni che attribuiscono alle variabili di ambiente la precedenza sulle proprietà del sistema Java o quando si passa esplicitamente da a. `System.getProperty()` `System.getenv()`
> Poiché le proprietà di sistema (passate tramite riga di comando) richiedono l'escape della shell per i caratteri speciali mentre le variabili di ambiente no, i valori possono essere risolti in modo diverso quando si utilizzano le variabili di ambiente anziché le proprietà di sistema Java.
> Se la tua applicazione è interessata, considera:
>
> - Rimozione dei caratteri di escape dai valori delle proprietà dell'ambiente durante l'utilizzo `System.getenv()`
> - Configurazione dell'applicazione per l'uso esplicito `System.getProperty()`
> - Verifica a fondo l'applicazione durante l'aggiornamento per garantire un comportamento coerente

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazio dei nomi della configurazione Tomcat

Puoi utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

Oltre alle opzioni supportate per tutti gli ambienti Elastic Beanstalk, la piattaforma Tomcat supporta quelle incluse negli spazi dei nomi seguenti:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: modifica le impostazioni JVM. Le opzioni in questo spazio dei nomi corrispondono a quelle nella console di gestione, in questo modo:
  - `Xms` – JVM command line options (Opzioni della riga di comando JVM
  - `JVM Options` – JVM command line options (Opzioni della riga di comando JVM
- `aws:elasticbeanstalk:environment:proxy`: scegli il server proxy dell'ambiente.

L'esempio di file di configurazione seguente mostra l'utilizzo delle opzioni di configurazione specifiche di Tomcat.

Example.ebextensions/tomcat-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xms: 512m
    JVM Options: '-Xmn128m'
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: mywebapi.zkpexsjtmd.us-west-2.elasticbeanstalk.com
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

La piattaforma Tomcat di AMI Amazon Linux (precedente ad Amazon Linux 2)

Se il tuo ambiente Tomcat di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2023I rami AL2 della piattaforma/sono incompatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.

- Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Namespace di configurazione Tomcat — Amazon Linux AMI () AL1

La piattaforma Tomcat di AMI Amazon Linux supporta opzioni aggiuntive nei seguenti spazi dei nomi:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: oltre alle opzioni menzionate in precedenza in questa pagina per questo spazio dei nomi, le versioni precedenti della piattaforma AMI Amazon Linux supportano anche:
  - `XX:MaxPermSize` – Maximum JVM permanent generation size (Dimensione massima di generazione permanente JVM
- `aws:elasticbeanstalk:environment:proxy`: oltre a scegliere il server proxy, configura anche la compressione delle risposte.

L'esempio di file di configurazione seguente mostra l'utilizzo delle opzioni di configurazione dello spazio dei nomi proxy.

Example.ebextensions/tomcat-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    GzipCompression: 'true'
    ProxyServer: nginx
```

Includi i file di configurazione di Elastic Beanstalk — Amazon Linux AMI () AL1

Per distribuire i file di configurazione `.ebextensions`, occorre includerli nella sorgente dell'applicazione. Per una singola applicazione, aggiungi `.ebextensions` a un file WAR compresso eseguendo il seguente comando:

Example

```
zip -ur your_application.war .ebextensions
```

Per un'applicazione che richiede più file WAR, consulta Raggruppamento di più file WAR per ambienti Tomcat per ulteriori informazioni.

Raggruppamento di più file WAR per ambienti Tomcat

Se la tua app Web comprende più componenti dell'applicazione Web, è possibile semplificare le distribuzioni e ridurre i costi operativi tramite l'esecuzione di componenti in un singolo ambiente, invece di eseguire un ambiente separato per ogni componente. Questa strategia è efficace per applicazioni leggere che non necessitano di una notevole quantità di risorse e per ambienti di sviluppo e test.

Per distribuire più applicazioni Web nell'ambiente, combina i file WAR (Web Application Archive) di ogni componente in un singolo bundle di origine.

Per creare un bundle di origine dell'applicazione che contenga più file WAR, organizza i file WAR utilizzando la seguente struttura.

```
MyApplication.zip
### .ebextensions
### .platform
### foo.war
### bar.war
### ROOT.war
```

Quando si distribuisce un pacchetto sorgente contenente più file WAR in un AWS Elastic Beanstalk ambiente, ogni applicazione è accessibile da un percorso diverso dal nome di dominio principale. L'esempio precedente include tre applicazioni: `foo`, `bar` e `ROOT`. `ROOT.war` è uno speciale nome di file che indica a Elastic Beanstalk di eseguire l'applicazione nel dominio radice, in modo che le tre applicazioni siano disponibili in `http://MyApplication.elasticbeanstalk.com/foo`, `http://MyApplication.elasticbeanstalk.com/bar` e `http://MyApplication.elasticbeanstalk.com`.

Il bundle di origine può includere file WAR, una cartella opzionale `.ebextensions` e una cartella opzionale `.platform`. Per informazioni dettagliate su queste cartelle di configurazione facoltative, consulta the section called “Estensione delle piattaforme Linux”.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk con questo link preconfigurato: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare la piattaforma e il ramo della piattaforma corrispondenti al linguaggio utilizzato dall'applicazione o la piattaforma Docker per le applicazioni basate su container.
3. Per Application code (Codice applicazione), scegliere Carica il tuo codice.
4. Selezionare Local file (File locale), quindi Browse (Sfoglia) e aprire il bundle di origine.
5. Selezionare Review and launch (Controlla e avvia).
6. Controlla le impostazioni disponibili, quindi seleziona Create app (Crea applicazione).

Per ulteriori informazioni su come creare bundle di origine, consulta Crea un bundle di sorgenti dell'applicazione Elastic Beanstalk.

Strutturazione della cartella di progetto

Per poter funzionare quando vengono distribuiti in un server Tomcat, gli archivi dell'applicazione Web (file WAR) Java Platform Enterprise Edition (Java EE) compilati devono essere strutturati in base a determinate linee guida. La directory di progetto non deve necessariamente soddisfare gli stessi standard, ma è una buona idea strutturarla allo stesso modo per semplificare la compilazione e la creazione di pacchetti. La strutturazione della cartella di progetto come contenuto di file WAR, inoltre, aiuta a comprendere la relazione tra i file e il relativo comportamento su un server Web.

Nella seguente gerarchia consigliata il codice sorgente per l'applicazione Web viene posizionato in una directory `src`, in modo da isolarlo dallo script di creazione e dal file WAR che genera.

```
~/workspace/my-app/
|-- build.sh            - Build script that compiles classes and creates a WAR
|-- README.MD           - Readme file with information about your project, notes
|-- ROOT.war            - Source bundle artifact created by build.sh
`-- src                 - Source code folder
    |-- WEB-INF         - Folder for private supporting files
    |   |-- classes     - Compiled classes
    |   |-- lib         - JAR libraries
    |   |-- tags        - Tag files
    |   |-- tlds        - Tag Library Descriptor files
```

```
    |   `-- web.xml     - Deployment Descriptor
    |-- com             - Uncompiled classes
    |-- css             - Style sheets
    |-- images          - Image files
    |-- js              - JavaScript files
    `-- default.jsp     - JSP (JavaServer Pages) webpage
```

Il contenuto della cartella `src` corrisponde a ciò che verrà inserito in pacchetti e distribuito al server, ad eccezione della cartella `com`. La cartella `com` contiene le classi non compilate (file `.java`). Queste devono essere compilate e inserite nella directory `WEB-INF/classes` per essere accessibili dal codice dell'applicazione.

La directory `WEB-INF` contiene il codice e le configurazioni che non vengono gestiti pubblicamente nel server Web. Le altre cartelle nella root della directory di origine (`css`, `images` e `js`) sono disponibili pubblicamente nel percorso corrispondente nel server Web.

L'esempio seguente è identico alla directory di progetto precedente, ma contiene ulteriori file e sottodirectory. Questo progetto di esempio include tag semplici, classi di supporto e modello e un file Java Server Page (JSP) per una risorsa `record`. Include anche un foglio di stile e JavaScript per Bootstrap, un file JSP predefinito e una pagina di errore per gli errori 404.

`WEB-INF/lib` include un file Java Archive (JAR) che contiene il driver Java Database Connectivity (JDBC) per PostgreSQL. `WEB-INF/classes` è vuoto perché i file di classe non sono ancora state compilate.

```
~/workspace/my-app/
|-- build.sh
|-- README.MD
|-- ROOT.war
`-- src
    |-- WEB-INF
    |   |-- classes
    |   |-- lib
    |   |   `-- postgresql-9.4-1201.jdbc4.jar
    |   |-- tags
    |   |   `-- header.tag
    |   |-- tlds
    |   |   `-- records.tld
    |   `-- web.xml
    |-- com
    |   `-- myapp
```

```
    |       |-- model
    |       |   `-- Record.java
    |       `-- web
    |           `-- ListRecords.java
    |-- css
    |   |-- bootstrap.min.css
    |   `-- myapp.css
    |-- images
    |   `-- myapp.png
    |-- js
    |   `-- bootstrap.min.js
    |-- 404.jsp
    |-- default.jsp
    `-- records.jsp
```

Creazione di un file WAR con uno script di shell

`build.sh` è uno script di shell molto semplice che compila classi Java, crea un file WAR e lo copia nella directory `webapps` di Tomcat per test locali.

```
cd src
javac -d WEB-INF/classes com/myapp/model/Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/model/
Record.java
javac -classpath WEB-INF/lib/*:WEB-INF/classes -d WEB-INF/classes com/myapp/web/
ListRecords.java

jar -cvf ROOT.war *.jsp images css js WEB-INF
cp ROOT.war /Library/Tomcat/webapps
mv ROOT.war ../
```

All'interno del file WAR troverai la stessa struttura esistente nella directory `src` dell'esempio precedente, esclusa la cartella `src/com`. Il comando `jar` crea automaticamente il file `META-INF/MANIFEST.MF`.

```
~/workspace/my-app/ROOT.war
|-- META-INF
|   `-- MANIFEST.MF
|-- WEB-INF
|   |-- classes
|   |   `-- com
|   |       `-- myapp
```

```
|   |           |-- model
|   |           |   `-- Records.class
|   |           `-- web
|   |               `-- ListRecords.class
|   |-- lib
|   |   `-- postgresql-9.4-1201.jdbc4.jar
|   |-- tags
|   |   `-- header.tag
|   |-- tlds
|   |   `-- records.tld
|   `-- web.xml
|-- css
|   |-- bootstrap.min.css
|   `-- myapp.css
|-- images
|   `-- myapp.png
|-- js
|   `-- bootstrap.min.js
|-- 404.jsp
|-- default.jsp
`-- records.jsp
```

Utilizzo `.gitignore`

Per evitare il commit dei file di classe e dei file WAR compilati nel repository Git o la visualizzazione di messaggi relativi a questi file quando esegui comandi Git, aggiungi i tipi di file rilevanti a un file denominato `.gitignore` nella cartella di progetto.

~/workspace/myapp/.gitignore

```
*.zip
*.class
```

Configurazione del server proxy

La piattaforma Tomcat utilizza nginx (impostazione predefinita) o Apache HTTP Server come proxy inverso per inoltrare le richieste dalla porta 80 dell'istanza al container Web Tomcat in ascolto sulla porta 8080. Elastic Beanstalk fornisce una configurazione proxy predefinita che puoi estendere o sostituire completamente a favore di quella personalizzata.

Configurazione del server proxy sulla versione della piattaforma

Tutte le AL2 piattaforme AL2023/supportano una funzionalità di configurazione proxy uniforme. Per ulteriori informazioni sulla configurazione del server proxy sulle versioni della piattaforma che eseguono AL2023/AL2, consultaConfigurazione del proxy inverso.

Configurazione del proxy sulla piattaforma Tomcat di AMI Amazon Linux (precedente ad Amazon Linux 2)

Se il tuo ambiente Tomcat di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2023I rami AL2 della piattaforma/sono incompatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Scelta di un server proxy per il tuo ambiente Tomcat — Amazon Linux AMI () AL1

Le versioni della piattaforma Tomcat basate su AMI Amazon Linux (precedente ad Amazon Linux 2) utilizzano per impostazione predefinita Apache 2.4 per il proxy. Puoi scegliere di usare Apache 2.2 o nginx includendo un file di configurazione nel codice sorgente. L'esempio seguente configura Elastic Beanstalk per l'uso di nginx.

Example.ebextensions/nginx-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: nginx
```

Migrazione da Apache 2.2 ad Apache 2.4 — Amazon Linux AMI () AL1

Se l'applicazione è stata sviluppata per Apache 2.2, leggi questa sezione per informazioni sulla migrazione ad Apache 2.4.

A partire dalle configurazioni della piattaforma Tomcat versione 3.0.0, rese disponibili con l'aggiornamento della piattaforma Java con Tomcat del 24 maggio 2018, Apache 2.4 è il proxy predefinito della piattaforma Tomcat. I file `.conf` di Apache 2.4 sono in gran parte, ma non interamente, compatibili con quelli della versione Apache 2.2 precedente. Elastic Beanstalk include file `.conf` predefiniti che funzionano correttamente con ogni versione di Apache. Se la tua applicazione non personalizza la configurazione di Apache, come descritto in Estensione e sovrascrittura della configurazione predefinita di Apache — Amazon Linux AMI () AL1, dovrebbe supportare la migrazione ad Apache 2.4 senza problemi.

Se l'applicazione estende o sostituisce la configurazione di Apache, potrebbe essere necessario effettuare alcune modifiche per migrare ad Apache 2.4. Per ulteriori informazioni, consulta la pagina relativa all'aggiornamento alla versione 2.4 dalla 2.2 sul sito Apache Software Foundation. Finché non esegui la migrazione ad Apache 2.4, come misura temporanea puoi scegliere di usare Apache 2.2 con l'applicazione includendo il file di configurazione seguente nel codice sorgente.

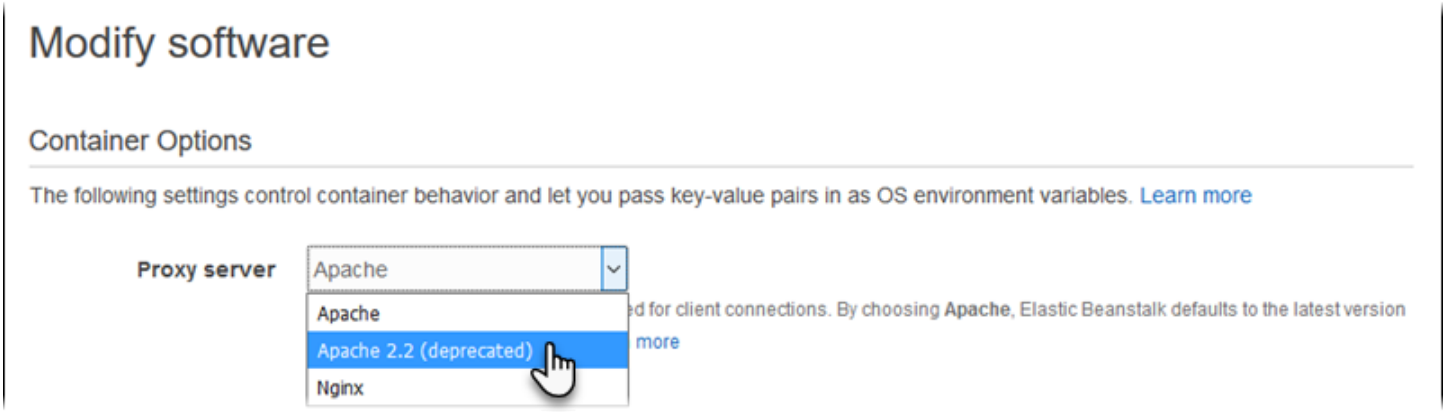
Example apache-legacy-proxy.ebextensions/ .config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache/2.2
```

Per una soluzione rapida, puoi anche selezionare il server proxy nella console Elastic Beanstalk.

Per selezionare il proxy nell'ambiente Tomcat nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. In Proxy server (Server proxy) scegli `Apache 2.2 (deprecated)`.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.



Estensione e sovrascrittura della configurazione predefinita di Apache — Amazon Linux AMI () AL1

Puoi estendere la configurazione Apache predefinita di Elastic Beanstalk con file di configurazione personalizzati aggiuntivi. In alternativa, puoi sostituire completamente la configurazione Apache predefinita di Elastic Beanstalk.

> **Note**
>
> - Tutte le piattaforme Amazon Linux 2 supportano una caratteristica di configurazione proxy uniforme. Per informazioni dettagliate sulla configurazione del server proxy sulle versioni della piattaforma Tomcat che eseguono Amazon Linux 2, consulta. Configurazione del proxy inverso
> - Se stai eseguendo la migrazione dell'applicazione Elastic Beanstalk a una piattaforma Amazon Linux 2, leggi anche le informazioni in the section called “Migrare a AL2 023/ AL2”.

Per estendere la configurazione Apache predefinita di Elastic Beanstalk, aggiungi i file di configurazione `.conf` a una cartella denominata `.ebextensions/httpd/conf.d` nel bundle di origine dell'applicazione. La configurazione Apache di Elastic Beanstalk include automaticamente i file `.conf` in questa cartella.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd
|      -- conf.d
|         -- myconf.conf
|         -- ssl.conf
-- index.jsp
```

Ad esempio, la configurazione Apache 2.4 seguente aggiunge un listener sulla porta 5000.

Example. ebextensions/httpd/conf.d/port5000. conf

```
listen 5000
<VirtualHost *:5000>
  <Proxy *>
    Require all granted
  </Proxy>
  ProxyPass / http://localhost:8080/ retry=0
  ProxyPassReverse / http://localhost:8080/
  ProxyPreserveHost on

  ErrorLog /var/log/httpd/elasticbeanstalk-error_log
</VirtualHost>
```

Per sostituire completamente la configurazione Apache predefinita di Elastic Beanstalk, includi una configurazione nel bundle di origine in `.ebextensions/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- httpd
|       `-- conf
|           `-- httpd.conf
`-- index.jsp
```

Se vuoi sostituire completamente la configurazione Apache di Elastic Beanstalk, aggiungi le righe seguenti a `httpd.conf` per ottenere le configurazioni di Elastic Beanstalk per Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk, la compressione delle risposte e i file statici.

```
IncludeOptional conf.d/*.conf
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Se l'ambiente usa Apache 2.2 come proxy, sostituisci le direttive `IncludeOptional` con `Include`. Per dettagli sul comportamento di queste due direttive nelle due versioni di Apache, consultate Include in Apache 2.4, in Apache 2.4 e Include IncludeOptional in Apache 2.2.

> **Note**
>
> Per sostituire il listener predefinito sulla porta 80, includi un file denominato `00_application.conf` in `.ebextensions/httpd/conf.d/elasticbeanstalk/` per sovrascrivere la configurazione di Elastic Beanstalk.

Per un esempio di utilizzo, osserva il file di configurazione predefinito di Elastic Beanstalk in `/etc/httpd/conf/httpd.conf` in un'istanza nell'ambiente. Tutti i file della cartella `.ebextensions/httpd` nel bundle di origine vengono copiati in `/etc/httpd` durante le distribuzioni.

Estensione della configurazione nginx predefinita — Amazon Linux AMI () AL1

Per estendere la configurazione nginx predefinita di Elastic Beanstalk, aggiungi file di configurazione `.conf` in una cartella denominata `.ebextensions/nginx/conf.d/` nel bundle di origine dell'applicazione. La configurazione nginx di Elastic Beanstalk include automaticamente i file `.conf` in questa cartella.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- conf.d
|           |-- elasticbeanstalk
|           |   `-- my-server-conf.conf
|           `-- my-http-conf.conf
`-- index.jsp
```

I file con estensione .conf nella cartella `conf.d` vengono inclusi nel blocco `http` della configurazione predefinita. I file nella cartella `conf.d/elasticbeanstalk` vengono inclusi nel blocco `server` all'interno del blocco `http`.

Per sostituire completamente la configurazione nginx predefinita di Elastic Beanstalk, includi una configurazione nel bundle di origine in `.ebextensions/nginx/nginx.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- nginx.conf
`-- index.jsp
```

Note

- Se sostituisci la configurazione nginx di Elastic Beanstalk, aggiungi la riga seguente al blocco `server` della configurazione per ottenere le configurazioni di Elastic Beanstalk per il listener della porta 80, la compressione delle risposte e i file statici.

```
include conf.d/elasticbeanstalk/*.conf;
```

- Per sostituire il listener predefinito sulla porta 80, includi un file denominato `00_application.conf` in `.ebextensions/nginx/conf.d/elasticbeanstalk/` per sovrascrivere la configurazione di Elastic Beanstalk.
- Includi anche la riga seguente nel blocco `http` della configurazione per ottenere le configurazioni di Elastic Beanstalk per Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk e la registrazione.

```
    include       conf.d/*.conf;
```

Per un esempio di utilizzo, osserva il file di configurazione predefinito di Elastic Beanstalk in `/etc/nginx/nginx.conf` in un'istanza nell'ambiente. Tutti i file della cartella `.ebextensions/nginx` nel bundle di origine vengono copiati in `/etc/nginx` durante le distribuzioni.

Utilizzo della piattaforma Java SE di Elastic Beanstalk

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Java eseguite sulla piattaforma AWS Elastic Beanstalk Java SE.

La piattaforma Elastic Beanstalk Java SE è un insieme di versioni della piattaforma per applicazioni Web Java che possono essere eseguite autonomamente da un file JAR compilato. Puoi compilare l'applicazione in locale o caricare il codice sorgente con uno script di compilazione per compilarlo nell'istanza. Le versioni della piattaforma Java SE sono raggruppate in rami della piattaforma, ognuno dei quali corrisponde a una versione principale di Java.

> **Note**
>
> Elastic Beanstalk non analizza il file JAR dell'applicazione. Mantieni i file necessari a Elastic Beanstalk all'esterno del file JAR. Ad esempio, è possibile includere il file `cron.yaml` di un ambiente lavoratore nella radice del bundle di origine dell'applicazione, accanto al file JAR.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

La piattaforma Java SE di Elastic Beanstalk include un server nginx che funge da proxy inverso che fornisce i contenuti statici memorizzati nella cache e trasmette le richieste all'applicazione. La piattaforma offre opzioni di configurazione per configurare il server proxy per fornire asset statici da una cartella nel codice sorgente, in modo da ridurre il carico sull'applicazione. Per gli scenari avanzati, puoi includere file .conf personalizzati nel bundle di origine per estendere la configurazione proxy di Elastic Beanstalk o sovrascriverla completamente.

Se fornisci solo un singolo file JAR per l'origine dell'applicazione (da solo, non all'interno di un bundle di origine), Elastic Beanstalk rinomina il tuo file JAR in `application.jar` e quindi lo esegue utilizzando `java -jar application.jar`. Per configurare i processi in esecuzione nelle istanze del server nell'ambiente, includi un oggetto Procfile opzionale nel bundle di origine. Un oggetto `Procfile` è obbligatorio se è presente più di un file JAR nella radice del bundle di origine o se desideri personalizzare il comando Java per l'impostazione delle opzioni JVM.

Si consiglia di fornire sempre un `Procfile` nel bundle di origine insieme all'applicazione. In questo modo si controlla con precisione quali processi vengono eseguiti da Elastic Beanstalk per l'applicazione e quali argomenti ricevono questi processi.

Per compilare le classi Java ed eseguire altri comandi di compilazione sulle EC2 istanze del tuo ambiente al momento della distribuzione, includi un Buildfile nel bundle dei sorgenti dell'applicazione. A `Buildfile` consente di distribuire il codice sorgente così com'è e di compilarlo sul server anziché

compilarlo localmente. JARs La piattaforma Java SE include strumenti di compilazione comuni per consentire la compilazione direttamente nel server.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Configurazione dell'ambiente Java SE

Le impostazioni della piattaforma Java SE ti consentono di ottimizzare il comportamento delle tue istanze Amazon EC2 . Puoi modificare la configurazione dell'istanza Amazon dell'ambiente Elastic Beanstalk utilizzando EC2 la console Elastic Beanstalk.

Utilizza la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3 e configurare variabili che l'applicazione è in grado di leggere dall'ambiente.

Per configurare l'ambiente Java SE nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Abilita la rotazione dei file di registro su Amazon S3: specifica se i file di log per le EC2 istanze Amazon dell'applicazione vengono copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura

delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Proprietà dell'ambiente

La sezione Environment Properties consente di specificare le impostazioni di configurazione dell'ambiente sulle EC2 istanze Amazon su cui è in esecuzione l'applicazione. Le proprietà di ambiente sono passate come coppie chiave-valore all'applicazione.

All'interno dell'ambiente Java SE in esecuzione in Elastic Beanstalk, le variabili di ambiente sono accessibili tramite `System.getenv()`. Ad esempio, è possibile leggere una proprietà denominata `API_ENDPOINT` su una variabile con il codice seguente:

```
String endpoint = System.getenv("API_ENDPOINT");
```

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazio dei nomi per la configurazione di Java SE

Puoi utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

La piattaforma Java SE non definisce spazi dei nomi specifici della piattaforma. È possibile configurare il proxy per servire file statici utilizzando lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles`. Per informazioni dettagliate e un esempio, consulta the section called “File statici”.

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

La piattaforma Java SE di AMI Amazon Linux (precedente ad Amazon Linux 2)

Se il tuo ambiente Java SE di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Namespace di configurazione Java SE — Amazon Linux AMI () AL1

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

La piattaforma Java SE supporta uno spazio dei nomi di configurazione specifico della piattaforma, in aggiunta agli spazi dei nomi supportati da tutte le piattaforme. Lo spazio dei nomi `aws:elasticbeanstalk:container:java:staticfiles` consente di definire le opzioni che mappano i percorsi nell'applicazione Web alle cartelle del bundle di origine dell'applicazione con contenuti statici.

Il frammento di codice option_settings definisce ad esempio due opzioni nello spazio dei nomi dei file statici. La prima mappa il percorso `/public` a una cartella denominata `public` e la seconda mappa il percorso `/images` a una cartella denominata `img`:

```
option_settings:
  aws:elasticbeanstalk:container:java:staticfiles:
    /html: statichtml
    /images: staticimages
```

Le cartelle mappate usando questo spazio dei nomi devono essere cartelle reali nella radice del bundle di origine. Non è possibile mappare un percorso a una cartella in un file JAR.

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Creazione JARs su server con un Buildfile

Puoi creare i file di classe e i file JAR dell'applicazione sulle EC2 istanze del tuo ambiente richiamando un comando build da un `Buildfile` file nel tuo bundle sorgente.

I comandi in un file `Buildfile` vengono eseguiti solo una volta e devono terminare al completamento, mentre i comandi in un Procfile vengono eseguiti per l'intera durata dell'applicazione e verranno riavviati se terminati. Per eseguirlo JARs nella tua applicazione, usa un. `Procfile`

Per informazioni dettagliate sul posizionamento e sulla sintassi di a`Buildfile`, vedereBuildfile e Procfile.

Nell'esempio `Buildfile` seguente viene eseguito Apache Maven per creare un'applicazione Web dal codice di origine. Consulta gli esempi dell'applicazione Web Java per un'applicazione di esempio che utilizza questa funzionalità.

Example Buildfile

```
build: mvn assembly:assembly -DdescriptorId=jar-with-dependencies
```

La piattaforma Java SE include i seguenti strumenti di build, che puoi chiamare dal tuo script di build:

- `javac` - Java compiler
- `ant` - Apache Ant
- `mvn` - Apache Maven
- `gradle` - Gradle

Configurazione del processo dell'applicazione con un Procfile

Se disponi di più file JAR nella cartella principale del bundle di origine dell'applicazione, è necessario includere un file `Procfile` che indichi a Elastic Beanstalk quale/i JAR eseguire. È anche possibile includere un file `Procfile` per una singola applicazione JAR per configurare la macchina virtuale Java (JVM) che esegue la tua applicazione.

Si consiglia di fornire sempre un `Procfile` nel bundle di origine insieme all'applicazione. In questo modo si controlla con precisione quali processi vengono eseguiti da Elastic Beanstalk per l'applicazione e quali argomenti ricevono questi processi.

Per dettagli sulla scrittura e l'uso di un file, `Procfile` vedereBuildfile e Procfile.

Example Procfile

```
web: java -Xms256m -jar server.jar
cache: java -jar mycache.jar
web_foo: java -jar other.jar
```

Il comando che esegue lo JAR principale nella tua applicazione deve essere chiamato `web` e deve essere il primo comando elencato nel `Procfile`. Il server nginx inoltra tutte le richieste HTTP che riceve dal load balancer del tuo ambiente a questa applicazione.

Elastic Beanstalk presuppone che tutte le voci in Procfile siano eseguite in qualsiasi momento e riavvia automaticamente qualsiasi applicazione definita nel Procfile che viene terminato. Per eseguire comandi che termineranno e non devono essere riavviati, utilizza un `Buildfile`.

Utilizzo di un Procfile su AMI Amazon Linux (precedente ad Amazon Linux 2)

Se il tuo ambiente Java SE di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Passaggio delle porte — Amazon Linux AMI (AL1)

Per impostazione predefinita, Elastic Beanstalk configura il proxy nginx per inoltrare le richieste alla tua applicazione sulla porta 5000. È possibile sostituire la porta predefinita impostando la proprietà dell'ambiente `PORT` sulla porta di ascolto dell'applicazione principale.

Se utilizzi un `Procfile` per eseguire più applicazioni, Elastic Beanstalk sulle versioni della piattaforma AMI Amazon Linux si aspetta che ogni applicazione aggiuntiva ascolti su una porta superiore di 100 rispetto alla precedente. Elastic Beanstalk imposta la variabile PORT accessibile all'interno di ogni applicazione per la porta su cui si aspetta venga eseguita l'applicazione. È possibile accedere a questa variabile all'interno dell'applicazione chiamando `System.getenv("PORT")`.

Nell'esempio precedente `Procfile`, l'applicazione `web` è in ascolto sulla porta 5000, `cache` è in ascolto sulla porta 5100 e `web_foo` sulla porta 5200. `web` configura la propria porta di ascolto leggendo la variabile `PORT` e aggiungendo 100 al numero di porta per determinare su quale porta è in ascolto `cache`, in modo che sia possibile inviarvi delle richieste.

Configurazione del server proxy

Elastic Beanstalk usa nginx come proxy inverso per mappare l'applicazione al load balancer Elastic Load Balancing sulla porta 80. Elastic Beanstalk fornisce una configurazione nginx predefinita che puoi estendere o ignorare completamente a favore di quella personalizzata.

Per impostazione predefinita, Elastic Beanstalk configura il proxy nginx per inoltrare le richieste alla tua applicazione sulla porta 5000. È possibile sostituire la porta predefinita impostando la proprietà dell'ambiente `PORT` sulla porta di ascolto dell'applicazione principale.

> **Note**
>
> La porta di ascolto della tua applicazione non influisce sulla porta che il server nginx ascolta per ricevere richieste dal load balancer.

Configurazione del server proxy sulla versione della piattaforma

Tutte le AL2 piattaforme AL2 023/ supportano una funzionalità di configurazione proxy uniforme. Per ulteriori informazioni sulla configurazione del server proxy sulle versioni della piattaforma che eseguono AL2 AL2 023/, consulta. Configurazione del proxy inverso

Configurazione del proxy su AMI Amazon Linux (precedente ad Amazon Linux 2)

Se il tuo ambiente Java SE di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2I rami della AL2 piattaforma 023/ non sono compatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Estensione e sovrascrittura della configurazione proxy predefinita — Amazon Linux AMI () AL1

Per estendere la configurazione nginx predefinita di Elastic Beanstalk, aggiungi file di configurazione `.conf` in una cartella denominata `.ebextensions/nginx/conf.d/` nel bundle di origine dell'applicazione. La configurazione nginx di Elastic Beanstalk include automaticamente i file `.conf` in questa cartella.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- conf.d
|           `-- myconf.conf
`-- web.jar
```

Per sostituire completamente la configurazione nginx predefinita di Elastic Beanstalk, includi una configurazione nel bundle di origine in `.ebextensions/nginx/nginx.conf`:

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- nginx.conf
```

```
`-- web.jar
```

Se ignori la configurazione nginx predefinita di Elastic Beanstalk, aggiungi al file `nginx.conf` la riga seguente per inserire le configurazioni di Elastic Beanstalk per Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk, le mappature automatiche delle applicazioni e i file statici.

```
include conf.d/elasticbeanstalk/*.conf;
```

La seguente configurazione di esempio dall'applicazione di esempio Scorekeep sostituisce la configurazione predefinita di Elastic Beanstalk per utilizzare un'applicazione Web statica dalla sottodirectory `public` di `/var/app/current`, in cui la piattaforma Java SE copia il codice sorgente dell'applicazione. La posizione `/api` inoltra il traffico verso i percorsi di `/api/` dell'applicazione Spring sulla porta 5000. Il traffico restante viene distribuito dall'applicazione Web al percorso root.

Example

```
user                    nginx;
error_log               /var/log/nginx/error.log warn;
pid                     /var/run/nginx.pid;
worker_processes        auto;
worker_rlimit_nofile    33282;

events {
    worker_connections  1024;
}

http {
  include       /etc/nginx/mime.types;
  default_type  application/octet-stream;

  log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

  include       conf.d/*.conf;

  map $http_upgrade $connection_upgrade {
      default     "upgrade";
  }

  server {
      listen        80 default_server;
```

```
      root /var/app/current/public;

      location / {
      }git pull


      location /api {
          proxy_pass          http://127.0.0.1:5000;
          proxy_http_version  1.1;

          proxy_set_header    Connection          $connection_upgrade;
          proxy_set_header    Upgrade             $http_upgrade;
          proxy_set_header    Host                $host;
          proxy_set_header    X-Real-IP           $remote_addr;
          proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_for;
      }

      access_log    /var/log/nginx/access.log main;

      client_header_timeout 60;
      client_body_timeout   60;
      keepalive_timeout     60;
      gzip                  off;
      gzip_comp_level       4;

      # Include the Elastic Beanstalk generated locations
      include conf.d/elasticbeanstalk/01_static.conf;
      include conf.d/elasticbeanstalk/healthd.conf;
  }
}
```

Aggiungere un'istanza Amazon RDS DB all'ambiente Java Elastic Beanstalk

Questo argomento fornisce istruzioni per creare un Amazon RDS utilizzando la console Elastic Beanstalk. Puoi utilizzare un'istanza database Amazon Relational Database Service (Amazon RDS) per archiviare i dati raccolti e modificati dall'applicazione. Il database può essere collegato all'ambiente e gestito da Elastic Beanstalk oppure può essere creato e gestito esternamente.

Se utilizzi Amazon RDS per la prima volta, aggiungi un'istanza database in un ambiente di test con la console Elastic Beanstalk e verifica che la tua applicazione sia in grado di connettersi a essa.

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'aggiunta di un'istanza database richiede circa 10 minuti. Quando l'aggiornamento dell'ambiente è completo, il nome host dell'istanza database e altre informazioni di connessione sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente:

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).

Nome proprietà	Descrizione	Valore proprietà
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Per ulteriori informazioni sulla configurazione di una istanza database interna, consulta Aggiunta di un database all'ambiente Elastic Beanstalk. Per istruzioni su come configurare un database esterno per l'uso con Elastic Beanstalk, consulta Utilizzo di Elastic Beanstalk con Amazon RDS.

Per connetterti al database, aggiungi il file JAR driver appropriato all'applicazione, carica la classe del driver nel codice e crea un oggetto di connessione con le proprietà dell'ambiente fornite da Elastic Beanstalk.

Sezioni

- Download del driver JDBC
- Connessione a un database (piattaforme Java SE)
- Connessione a un database (piattaforme Tomcat)
- Risoluzione dei problemi di connessione del database

Download del driver JDBC

Avrai bisogno del file JAR del driver JDBC per il motore di database scelto. Salva il file JAR nel tuo codice sorgente e includilo nel tuo classpath quando compili la classe che crea le connessioni al database.

Puoi trovare il driver più recente per il motore di database nelle seguenti posizioni:

- MySQL - MySQL Connector/J
- Oracle SE-1 - Oracle JDBC Driver
- Postgres - PostgreSQL JDBC Driver
- SQL Server - Microsoft JDBC Driver

Per usare il driver JDBC, effettuare la chiamata a `Class.forName()` per caricarlo prima di creare la connessione con `DriverManager.getConnection()` nel codice.

JDBC utilizza una stringa di connessione nel formato seguente:

```
jdbc:driver://hostname:port/dbName?user=userName&password=password
```

Puoi recuperare il nome host, la porta, il nome del database, il nome utente e la password dalle variabili di ambiente che Elastic Beanstalk fornisce alla tua applicazione. Il nome del driver è specifico per il tuo tipo di database e per la versione del driver. Di seguito vengono riportati esempi di nomi di driver:

- `mysql` per MySQL
- `postgresql` per PostgreSQL
- `oracle:thin` per Oracle Thin
- `oracle:oci` per Oracle OCI
- `oracle:oci8` per Oracle OCI 8
- `oracle:kprb` per Oracle KPRB
- `sqlserver` per SQL Server

Connessione a un database (piattaforme Java SE)

In un ambiente Java SE, utilizza `System.getenv()` per leggere le variabili di connessione dall'ambiente. Il seguente codice di esempio mostra una classe che crea una connessione a un database PostgreSQL.

```
private static Connection getRemoteConnection() {
    if (System.getenv("RDS_HOSTNAME") != null) {
      try {
      Class.forName("org.postgresql.Driver");
      String dbName = System.getenv("RDS_DB_NAME");
      String userName = System.getenv("RDS_USERNAME");
      String password = System.getenv("RDS_PASSWORD");
      String hostname = System.getenv("RDS_HOSTNAME");
      String port = System.getenv("RDS_PORT");
      String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
      logger.trace("Getting remote connection with connection string from environment
 variables.");
      Connection con = DriverManager.getConnection(jdbcUrl);
      logger.info("Remote connection successful.");
      return con;
    }
```

```
    catch (ClassNotFoundException e) { logger.warn(e.toString());}
    catch (SQLException e) { logger.warn(e.toString());}
    }
    return null;
  }
```

Connessione a un database (piattaforme Tomcat)

In un ambiente Tomcat, le proprietà dell'ambiente vengono fornite come proprietà di sistema accessibili con `System.getProperty()`.

Il seguente codice di esempio mostra una classe che crea una connessione a un database PostgreSQL.

```
private static Connection getRemoteConnection() {
    if (System.getProperty("RDS_HOSTNAME") != null) {
      try {
      Class.forName("org.postgresql.Driver");
      String dbName = System.getProperty("RDS_DB_NAME");
      String userName = System.getProperty("RDS_USERNAME");
      String password = System.getProperty("RDS_PASSWORD");
      String hostname = System.getProperty("RDS_HOSTNAME");
      String port = System.getProperty("RDS_PORT");
      String jdbcUrl = "jdbc:postgresql://" + hostname + ":" + port + "/" + dbName + "?
user=" + userName + "&password=" + password;
      logger.trace("Getting remote connection with connection string from environment
 variables.");
      Connection con = DriverManager.getConnection(jdbcUrl);
      logger.info("Remote connection successful.");
      return con;
    }
    catch (ClassNotFoundException e) { logger.warn(e.toString());}
    catch (SQLException e) { logger.warn(e.toString());}
    }
    return null;
  }
```

In caso di problemi di connessione o di esecuzione di istruzioni SQL, prova a inserire il codice seguente in un file JSP. Questo codice si connette a un'istanza database, crea una tabella e scrive su di essa.

```
<%@ page import="java.sql.*" %>
```

```
<%
  // Read RDS connection information from the environment
  String dbName = System.getProperty("RDS_DB_NAME");
  String userName = System.getProperty("RDS_USERNAME");
  String password = System.getProperty("RDS_PASSWORD");
  String hostname = System.getProperty("RDS_HOSTNAME");
  String port = System.getProperty("RDS_PORT");
  String jdbcUrl = "jdbc:mysql://" + hostname + ":" +
    port + "/" + dbName + "?user=" + userName + "&password=" + password;

  // Load the JDBC driver
  try {
    System.out.println("Loading driver...");
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("Driver loaded!");
  } catch (ClassNotFoundException e) {
    throw new RuntimeException("Cannot find the driver in the classpath!", e);
  }

  Connection conn = null;
  Statement setupStatement = null;
  Statement readStatement = null;
  ResultSet resultSet = null;
  String results = "";
  int numresults = 0;
  String statement = null;

  try {
    // Create connection to RDS DB instance
    conn = DriverManager.getConnection(jdbcUrl);

    // Create a table and write two rows
    setupStatement = conn.createStatement();
    String createTable = "CREATE TABLE Beanstalk (Resource char(50));";
    String insertRow1 = "INSERT INTO Beanstalk (Resource) VALUES ('EC2 Instance');";
    String insertRow2 = "INSERT INTO Beanstalk (Resource) VALUES ('RDS Instance');";

    setupStatement.addBatch(createTable);
    setupStatement.addBatch(insertRow1);
    setupStatement.addBatch(insertRow2);
    setupStatement.executeBatch();
    setupStatement.close();

  } catch (SQLException ex) {
```

```
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
  } finally {
    System.out.println("Closing the connection.");
    if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
  }

  try {
    conn = DriverManager.getConnection(jdbcUrl);

    readStatement = conn.createStatement();
    resultSet = readStatement.executeQuery("SELECT Resource FROM Beanstalk;");

    resultSet.first();
    results = resultSet.getString("Resource");
    resultSet.next();
    results += ", " + resultSet.getString("Resource");

    resultSet.close();
    readStatement.close();
    conn.close();

  } catch (SQLException ex) {
    // Handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
  } finally {
       System.out.println("Closing the connection.");
      if (conn != null) try { conn.close(); } catch (SQLException ignore) {}
  }
%>
```

Per visualizzare i risultati, inserisci il codice seguente nel corpo della porzione HTML del file JSP.

```
<p>Established connection to RDS. Read first two rows: <%= results %></p>
```

Risoluzione dei problemi di connessione del database

> **Prova Amazon Q Developer CLI per la risoluzione dei problemi assistita dall'intelligenza artificiale**
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Risoluzione dei problemi degli ambienti Elastic Beanstalk con Amazon Q Developer CLI nei blog. AWS

In caso di problemi di connessione a un database all'interno dell'applicazione, rivedi il log del container Web e il database.

Revisione dei log

Puoi visualizzare tutti i log dall'ambiente Elastic Beanstalk all'interno di Eclipse. Se non hai la vista AWS Explorer aperta, scegli la freccia accanto all' AWS icona arancione nella barra degli strumenti, quindi scegli Mostra visualizzazione Explorer. AWS Espandi AWS Elastic Beanstalk e il nome dell'ambiente, quindi apri il menu contestuale (pulsante destro del mouse) per il server. Seleziona Open in WTP Server Editor (Apri in WTP Server Editor).

Scegli la scheda Log della vista del Server per visualizzare i log di aggregazione dell'ambiente. Per aprire gli ultimi log, scegli il pulsante Refresh (Aggiorna) nell'angolo in alto a destra della pagina.

Scendi verso il basso per individuare i log Tomcat in `/var/log/tomcat7/catalina.out`. Se hai caricato la pagina Web dal nostro esempio precedente più volte, potresti visualizzare quanto segue.

```
-------------------------------------
/var/log/tomcat7/catalina.out
-------------------------------------
INFO: Server startup in 9285 ms
Loading driver...
Driver loaded!
SQLException: Table 'Beanstalk' already exists
SQLState: 42S01
VendorError: 1050
Closing the connection.
```

```
Closing the connection.
```

Tutte le informazioni che l'applicazione Web invia all'output standard appaiono nel log del container Web. Nell'esempio precedente, l'applicazione cerca di creare la tabella ogni volta che la pagina viene caricata. Questo determina un'eccezione SQL su tutte le pagine caricate dopo la prima.

Ad esempio, l'operazione precedente è accettabile. Tuttavia, nelle applicazioni concrete, mantieni le definizioni del tuo database in oggetti dello schema, esegui le transazioni all'interno di classi di modelli e coordina le richieste con servlet di controller.

Connessione all'istanza database RDS

Puoi connetterti direttamente all'istanza database RDS all'interno dell'ambiente Elastic Beanstalk utilizzando l'applicazione client MySQL.

In primo luogo, apri il gruppo di sicurezza all'istanza database di RDS per consentire il traffico dal tuo computer.

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Accanto a Endpoint, scegli il collegamento della console Amazon RDS.
6. Nella pagina dei dettagli dell'istanza RDS Dashboard (Pannello di controllo RDS) in Security and Network (Sicurezza e rete), seleziona il gruppo di sicurezza che inizia con rds- accanto a Security Groups (Gruppi di sicurezza).

   > **Note**
   >
   > Il database potrebbe avere più voci etichettate con Security Groups(Gruppi di sicurezza). Utilizza il primo che inizia con awseb, solo se hai un account precedente che non dispone di un Amazon Virtual Private Cloud (Amazon VPC) predefinito.

7. In Security group details (Dettagli gruppi di sicurezza), seleziona la scheda Inbound (In ingresso), quindi Edit (Modifica).
8. Aggiungi una regola per MySQL (porta 3306) che consente il traffico dal tuo indirizzo IP, specificato nel formato CIDR.

9. Scegliere Save (Salva). Le modifiche diventano effettive immediatamente.

Torna ai dettagli di configurazione di Elastic Beanstalk per l'ambiente e prendi nota dell'endpoint. Puoi utilizzare il nome di dominio per la connessione all'istanza database di RDS.

Installa il client MySQL e avvia una connessione al database sulla porta 3306. In Windows, installa MySQL Workbench dalla home page di MySQL e segui le istruzioni.

Su Linux, installa il client MySQL utilizzando il programma di gestione dei pacchetti per la tua distribuzione. L'esempio seguente funziona con Ubuntu e altri derivati Debian.

```
// Install MySQL client
$ sudo apt-get install mysql-client-5.5
...
// Connect to database
$ mysql -h aas839jo2vwhwb.cnubrrfwfka8.us-west-2.rds.amazonaws.com -u username -
ppassword ebdb
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 117
Server version: 5.5.40-log Source distribution
...
```

Dopo aver effettuato la connessione, è possibile eseguire comandi SQL per visualizzare lo stato del database, se le tabelle e le righe sono state create e altre informazioni.

```
mysql> SELECT Resource from Beanstalk;
+--------------+
| Resource     |
+--------------+
| EC2 Instance |
| RDS Instance |
+--------------+
2 rows in set (0.01 sec)
```

Strumenti e risorse Java

Puoi trovare informazioni utili per lo sviluppo delle applicazioni Java in diverse posizioni:

Risorsa	Description
Il forum di sviluppo AWS Java	Pubblica le tue domande e ricevi un feedback.
Centro sviluppatori Java	Punto di riferimento per il codice di esempio, la documentazione, gli strumenti e risorse aggiuntive.

Distribuzione di applicazioni Node.js con Elastic Beanstalk

Questo capitolo fornisce istruzioni per configurare e distribuire l'applicazione Web Node.js su. AWS Elastic Beanstalk Fornisce inoltre procedure dettagliate per attività comuni come l'integrazione del database e l'utilizzo del framework Express. Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni Web Node.js utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, puoi continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure puoi utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Segui le step-by-step istruzioni QuickStart per Node.js per creare e distribuire un'applicazione Web Hello World Node.js con l'EB CLI.

Argomenti

- QuickStart: distribuzione di un' Node.js applicazione su Elastic Beanstalk
- Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk
- Utilizzo della piattaforma Node.js di Elastic Beanstalk
- Altri esempi di applicazioni e tutorial di Elastic Beanstalk per Node.js
- Distribuzione di un'applicazione Node.js Express su Elastic Beanstalk
- Distribuzione di un'applicazione Node.js Express con clustering su Elastic Beanstalk
- Distribuzione di un'applicazione Node.js con DynamoDB in Elastic Beanstalk
- Aggiungere un'istanza Amazon RDS DB all'ambiente Node.js Elastic Beanstalk
- Strumenti e risorse Node.js

QuickStart: distribuzione di un' Node.js applicazione su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un' Node.js applicazione e di distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Fase 1: Creare un' Node.js applicazione
- Passaggio 2: Esegui l'applicazione localmente
- Fase 3: Implementazione Node.js dell'applicazione con l'EB CLI
- Fase 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Node.js

Installa Node.js sul tuo computer locale seguendo la procedura di installazione Node.js sul Node.js sito Web.

Verifica l' Node.js installazione eseguendo il comando seguente.

```
~$ node -v
```

Fase 1: Creare un' Node.js applicazione

Crea una directory di progetto.

```
~$ mkdir eb-nodejs
~$ cd eb-nodejs
```

Successivamente, crea un'applicazione che distribuirai utilizzando Elastic Beanstalk. A questo proposito, creeremo un servizio Web RESTful "Hello World".

Example~/eb- .js nodejs/server

```
const http = require('node:http');
```

```
const hostname = '127.0.0.1';
const port = 8080;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello Elastic Beanstalk!\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Questa applicazione apre un listener sulla porta 8080. Elastic Beanstalk inoltra le richieste all'applicazione sulla porta 8080 per impostazione predefinita per. Node.js

Passaggio 2: Esegui l'applicazione localmente

Esegui il comando seguente per eseguire l'applicazione localmente.

```
~/eb-nodejs$ node server.js
```

Dovresti vedere il testo seguente.

```
Server running at http://127.0.0.1:8080/
```

Inserisci l'indirizzo URL `http://127.0.0.1:8080/` nel tuo browser web. Il browser dovrebbe mostrare «Hello Elastic Beanstalk!».

Fase 3: Implementazione Node.js dell'applicazione con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire l'applicazione Node.js

1. Inizializza il repository della CLI EB con il comando eb init.

   ```
   ~/eb-nodejs$ eb init -p node.js nodejs-tutorial --region us-east-2
   ```

Questo comando crea un'applicazione denominata `nodejs-tutorial` e configura il repository locale per creare ambienti con la versione più recente Node.js della piattaforma.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

```
~/eb-nodejs$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo distribuisce su un'istanza EC2 nell'ambiente. Dopo aver distribuito l'applicazione, Elastic Beanstalk la avvia sulla porta 8080.

```
~/eb-nodejs$ eb create nodejs-env
```

Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-nodejs$ eb open
```

Complimenti! Hai distribuito un' Node.js applicazione con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 5: rimozione

Puoi chiudere l'ambiente quando hai finito di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-nodejs$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che viene indirizzato alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

> **Prova altri tutorial**
>
> Se desideri provare altri tutorial con diverse applicazioni di esempio, consulta. Altri esempi di applicazioni e tutorial di Elastic Beanstalk per Node.js

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire Node.js applicazioni localmente, consulta. Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk

Questo argomento fornisce istruzioni per configurare un ambiente di sviluppo Node.js in cui testare l'applicazione localmente prima di distribuirla. AWS Elastic Beanstalk Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Argomenti

- Installa Node.js
- Conferma l'installazione di npm
- Installa l' AWS SDK per Node.js
- Installazione del generatore Express

- Configurazione di un framework e un server Express

Installa Node.js

Installa Node.js per eseguire applicazioni Node.js in locale. Se non hai preferenze, scarica la versione più recente supportata da Elastic Beanstalk. consulta Node.js nel documento relativo alle Piattaforme AWS Elastic Beanstalk per visionare un elenco delle versioni supportate.

Scarica Node.js all'indirizzo nodejs.org.

Conferma l'installazione di npm

Node.js utilizza un programma di gestione dei pacchetti npm che consente di installare strumenti e framework da utilizzare nell'applicazione. Poiché npm è distribuito con Node.js, verrà installato automaticamente quando si scarica e si installa Node.js. Per confermare l'installazione di npm è possibile eseguire il seguente comando:

```
$ npm -v
```

Per ulteriori informazioni su npm, visita il sito web npmjs.

Installa l' AWS SDK per Node.js

Se devi gestire AWS le risorse dall'interno dell'applicazione, installa l' AWS SDK for JavaScript in Node.js. Installazione dell'SDK con npm:

```
$ npm install aws-sdk
```

Per ulteriori informazioni, visita la home page di AWS SDK for JavaScript in Node.js.

Installazione del generatore Express

Express è un framework di applicazione Web in esecuzione su Node.js. Per usarlo, installa prima l'applicazione della riga di comando del generatore Express. Una volta installato il generatore Express, potrai eseguire il comando express per generare una struttura del progetto di base per la tua applicazione Web. Una volta installati il progetto di base, i file e le dipendenze, potrai avviare un server Express locale sul tuo computer di sviluppo.

> **Note**
>
> - I seguenti passaggi ti guideranno attraverso l'installazione del generatore Express su un sistema operativo Linux.
> - Per Linux, a seconda del livello di autorizzazione per le directory di sistema, potrebbe essere necessario aggiungere il prefisso `sudo` ad alcuni di questi comandi.

Installazione del generatore Express nel tuo ambiente di sviluppo

1. Crea una directory di lavoro per il tuo framework e server Express.

```
~$ mkdir node-express
~$ cd node-express
```

2. Installa Express globalmente per avere accesso al comando `express`.

```
~/node-express$ npm install -g express-generator
```

3. A seconda del sistema operativo, potrebbe essere necessario impostare il percorso per eseguire il comando `express`. Le informazioni sull'impostazione della variabile del percorso sono riportate nell'output del passo precedente. Di seguito è riportato un esempio per Linux.

```
~/node-express$ export PATH=$PATH:/usr/local/share/npm/bin/express
```

   Seguendo i tutorial in questo capitolo, dovrai eseguire il comando express da diverse directory. Ogni tutorial imposta una struttura di progetto Express di base nella propria directory.

A questo punto hai installato il generatore della riga di comando Express da usare per creare una directory di framework per la tua applicazione Web, configurare le dipendenze e avviare il server delle app Web. Successivamente, completeremo i passaggi per eseguire questa operazione nella directory `node-express` che abbiamo creato.

Configurazione di un framework e un server Express

Completa la seguente procedura per creare le directory e i contenuti del framework Express di base. I tutorial di questo capitolo includono anche questi passaggi per configurare il framework Express di base in ciascuna delle directory delle applicazioni del tutorial.

Configurazione di un framework e un server Express

1. Esegui il comando `express`. Questo genera `package.json`, `app.js` e poche altre directory.

   ```
   ~/node-express$ express
   ```

   Quando ti viene chiesto se desideri proseguire, digita **y**.

2. Configura le dipendenze locali.

   ```
   ~/node-express$ npm install
   ```

3. Verifica dell'avvio del server dell'applicazione Web.

   ```
   ~/node-express$ npm start
   ```

   Verrà visualizzato un output simile al seguente:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   > node ./bin/www
   ```

   Il server viene eseguito sulla porta 3000 per impostazione predefinita. Per testarlo, esegui `curl http://localhost:3000` su un altro terminale oppure apri un browser sul computer locale e immetti l'indirizzo URL `http://localhost:3000`.

   Premi Ctrl+C per arrestare il server.

Utilizzo della piattaforma Node.js di Elastic Beanstalk

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Node.js su Elastic Beanstalk.

AWS Elastic Beanstalk supporta una serie di rami di piattaforma per diverse versioni del linguaggio di programmazione Node.js. Per un elenco completo, consulta Node.js nel documento AWS Elastic Beanstalk Platforms.

Elastic Beanstalk fornisce le opzioni di configurazione che puoi utilizzare per personalizzare il software in esecuzione sulle istanze EC2 nell'ambiente Elastic Beanstalk. È possibile configurare le variabili di ambiente necessarie per l'applicazione, abilitare la rotazione dei log su Amazon S3

e mappare le cartelle nel codice sorgente dell'applicazione che contengono i file statici ai percorsi serviti dal server proxy.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

Puoi includere un file `Package.json` nel bundle di origine per installare i pacchetti durante l'implementazione, per fornire un comando start e per specificare la versione di Node.js che desideri sia utilizzata dall'applicazione. Puoi includere un file `npm-shrinkwrap.json` per bloccare le versioni delle dipendenze.

La piattaforma Node.js include un server proxy che serve asset statici, inoltra il traffico verso la tua applicazione e comprime le risposte. Puoi estendere o sostituire la configurazione proxy predefinita per scenari avanzati.

Ci sono diverse opzioni per avviare l'applicazione. Puoi aggiungere un Procfile al tuo bundle di origine per specificare il comando che avvia l'applicazione. Quando non fornisci un `Procfile` ma specifichi un file `package.json`, Elastic Beanstalk esegue `npm start`. Se non viene fornito, Elastic Beanstalk cerca il file `app.js` o `server.js`, in questo ordine, e lo esegue.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Configurazione dell'ambiente Node.js

Le impostazioni della piattaforma Node.js consentono di ottimizzare il comportamento delle istanze Amazon EC2. Puoi modificare la configurazione dell'istanza Amazon EC2 dell'ambiente Elastic Beanstalk utilizzando la console Elastic Beanstalk.

Utilizza la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3 e configurare variabili che l'applicazione è in grado di leggere dall'ambiente.

Configurazione dell'ambiente Node.js nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni del contenitore

È possibile specificare le seguenti opzioni specifiche della piattaforma:

- Proxy server (Server proxy): il server proxy da utilizzare nelle istanze dell'ambiente. Per impostazione predefinita, viene utilizzato NGINX.

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Enable log file rotation to Amazon S3 (Abilita rotazione file di log in Amazon S3): specifica se i file di log per le istanze Amazon EC2 dell'applicazione debbano essere copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Proprietà dell'ambiente

La sezione Environment Properties (Proprietà ambiente) consente di specificare le impostazioni di configurazione dell'ambiente nelle istanze Amazon EC2 che eseguono l'applicazione. Queste impostazioni sono passate come coppie chiave-valore all'applicazione.

All'interno dell'Node.jsambiente in cui viene eseguito AWS Elastic Beanstalk, è possibile accedere alle variabili di ambiente eseguendo`process.env.ENV_VARIABLE`.

```
var endpoint = process.env.API_ENDPOINT
```

La piattaforma Node.js imposta la variabile di ambiente PORT sulla porta a cui il server proxy trasferisce il traffico. Per ulteriori informazioni, consulta Configurazione del server proxy.

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Configurazione di un ambiente Node.js per l'AMI Amazon Linux (precedente ad Amazon Linux 2)

Le seguenti categorie di configurazione del software della console sono supportate solo in un ambiente Node.js di Elastic Beanstalk che utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2).

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2023I rami AL2 della piattaforma/sono incompatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.

- Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Opzioni del contenitore: Amazon Linux AMI (AL1)

Nella pagina di configurazione specifica le informazioni seguenti:

- Proxy server (Server proxy): specifica il server Web da utilizzare per le connessioni proxy a Node.js. Per impostazione predefinita, viene utilizzato NGINX. Se selezioni none, (nessuno), le mappature dei file statici non vengono applicate e la compressione GZIP è disabilitata.
- Versione di Node.js: specifica la versione di Node.js. Per un elenco delle versioni di Node.js supportate, consulta Node.js nella guida Piattaforme AWS Elastic Beanstalk .
- Compressione GZIP: specifica se la compressione GZIP è abilitata. Per impostazione predefinita, la compressione GZIP è abilitata.
- Node command (Comando nodo): consente di immettere il comando utilizzato per avviare l'applicazione Node.js. Una stringa vuota (impostazione predefinita) indica che Elastic Beanstalk usa`app.js`, quindi `server.js` e poi `npm start`.

Spazi dei nomi per la configurazione di Node.js

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

È possibile scegliere il proxy da utilizzare nelle istanze dell'ambiente utilizzando lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy`. L'esempio seguente configura l'ambiente in modo da utilizzare il server proxy HTTPD di Apache.

Example.ebextensions/nodejs-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
```

```
    ProxyServer: apache
```

È possibile configurare il proxy per servire file statici utilizzando lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles`. Per ulteriori informazioni e un esempio, consulta the section called “File statici”.

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

La piattaforma Node.js dell'AMI Amazon Linux (precedente ad Amazon Linux 2)

Se il tuo ambiente Node.js di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), considera le configurazioni e i suggerimenti specifici in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2023I rami AL2 della piattaforma/sono incompatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Node.jsopzioni di configurazione specifiche della piattaforma — Amazon Linux AMI () AL1

Elastic Beanstalk supporta alcune opzioni di configurazione specifiche della piattaforma per le versioni della piattaforma Node.js dell'AMI Amazon Linux. Puoi scegliere su quale server proxy eseguire l'applicazione, una versione specifica di Node.js da eseguire e il comando utilizzato per eseguire l'applicazione.

Per il server proxy, puoi utilizzare un NGINX o Apache. È possibile impostare il valore none sull'opzione `ProxyServer`. In questo caso, Elastic Beanstalk esegue l'applicazione come

standalone, ossia non dietro un server proxy. Se l'ambiente esegue un'applicazione autonoma, aggiorna il codice per ascoltare sulla porta su cui NGINX inoltra il traffico.

```
var port = process.env.PORT || 8080;

app.listen(port, function() {
  console.log('Server running at http://127.0.0.1:%s', port);
});
```

Node.jsversioni linguistiche — Amazon Linux AMI (AL1)

In termini di versione del linguaggio supportata, la piattaforma Node.js dell'AMI Amazon Linux è leggermente diversa rispetto ad altre piattaforme gestite da Elastic Beanstalk. Questo perché ogni versione della piattaforma Node.js supporta solo alcune versioni del linguaggio Node.js. Per un elenco delle versioni di Node.js supportate, consulta Node.js nella guida Piattaforme AWS Elastic Beanstalk .

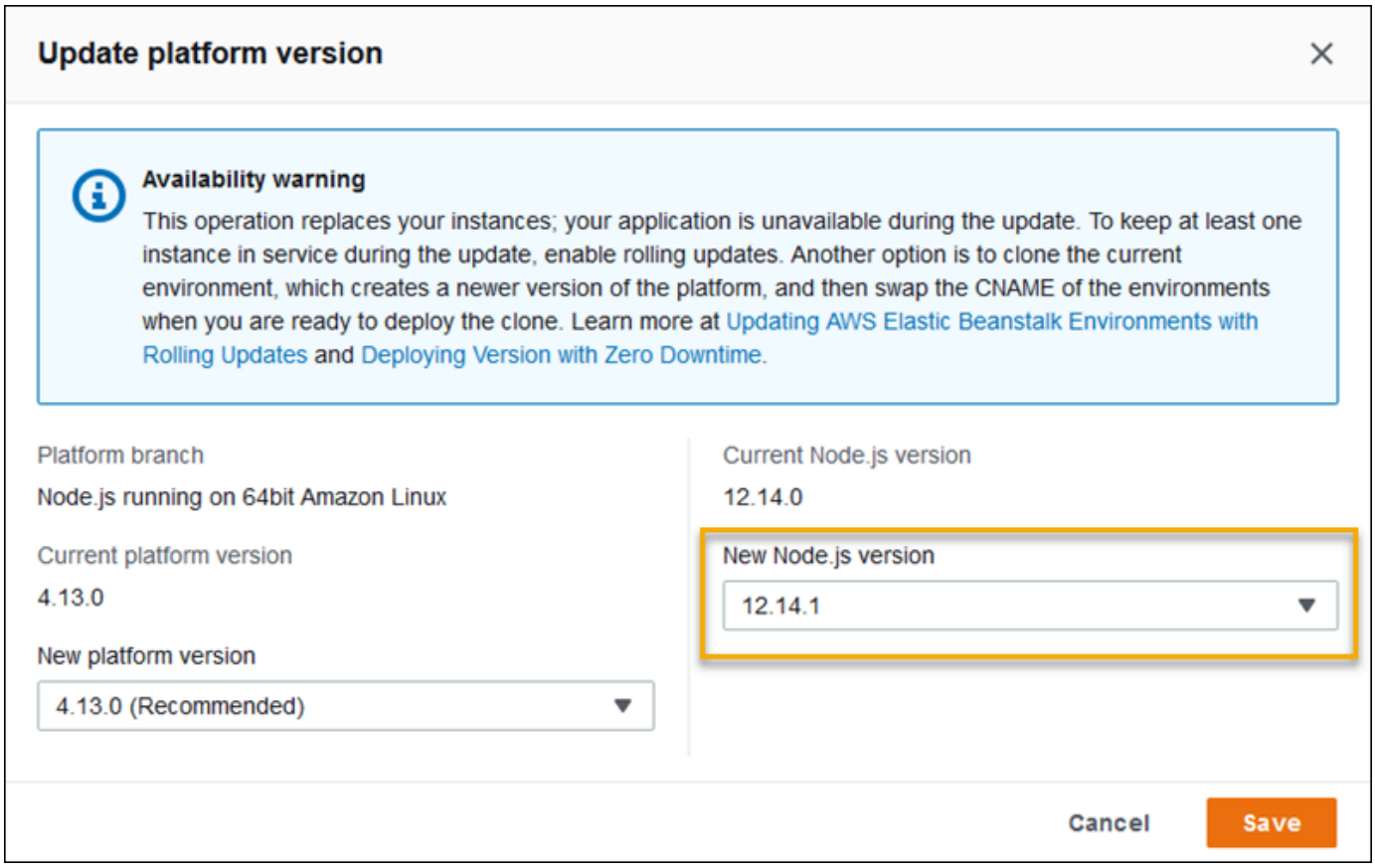
Puoi utilizzare un'opzione di configurazione specifica della piattaforma per impostare la versione del linguaggio. Per istruzioni, consulta the section called “Configurazione dell'ambiente Node.js”. In alternativa, utilizza la console Elastic Beanstalk per aggiornare la versione di Node.js usata dall'ambiente come parte dell'aggiornamento della versione della piattaforma.

> **Note**
>
> Quando il supporto della versione di Node.js in uso viene rimosso dalla piattaforma, è necessario modificare o rimuovere l'impostazione della versione prima di eseguire un aggiornamento della piattaforma. Questo può verificarsi quando viene rilevata una vulnerabilità di sicurezza per una o più versioni di Node.js
> Quando ciò accade, il tentativo di aggiornamento a una nuova versione della piattaforma che non supporta quella configurata NodeVersionfallisce. Per evitare di dover creare un nuovo ambiente, modificate l'opzione di NodeVersionconfigurazione impostando una versione di Node.js supportata sia dalla vecchia versione della piattaforma che da quella nuova, oppure rimuovete l'impostazione dell'opzione e quindi eseguite l'aggiornamento della piattaforma.

Configurazione della versione di Node.jsdell'ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina panoramica dell'ambiente, in Platform (Piattaforma), scegliere Change (Cambia).
4. Nella finestra di dialogo Update platform version (Aggiorna versione piattaforma), seleziona una versione di Node.js.



5. Scegli Save (Salva).

Node.jsnamespace di configurazione — Amazon Linux AMI () AL1

La piattaforma Node.js dell'AMI Amazon Linux definisce le opzioni aggiuntive negli spazi dei nomi `aws:elasticbeanstalk:container:nodejs:staticfiles` e `aws:elasticbeanstalk:container:nodejs`.

Il seguente file di configurazione indica a Elastic Beanstalk di utilizzare `npm start` per eseguire l'applicazione. Inoltre, imposta il tipo di proxy su Apache e abilita la compressione. Infine, configura il proxy per servire file statici da due directory fonte. Una origine è costituita da file HTML nel percorso `html` nella root del sito Web dalla directory di origine `statichtml`. L'altra fonte è costituita da file

di immagine sul percorso `images` sotto la directory principale del sito Web dalla directory fonte `staticimages`.

Example.ebextensions/node-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:nodejs:
    NodeCommand: "npm start"
    ProxyServer: apache
    GzipCompression: true
  aws:elasticbeanstalk:container:nodejs:staticfiles:
    /html: statichtml
    /images: staticimages
```

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Configurazione di comandi di avvio personalizzati con un Elastic Procfile Beanstalk

Puoi includere un file chiamato `Procfile` nella root del pacchetto di origine per specificare il comando che avvia l'applicazione.

Example Procfile

```
web: node index.js
```

Per informazioni sull'`Procfile`utilizzo, consulta. Buildfile e Procfile

> **Note**
>
> Questa funzionalità sostituisce l'opzione legacy `NodeCommand` nello spazio dei nomi `aws:elasticbeanstalk:container:nodejs`.

Configurazione delle dipendenze dell'applicazione su Elastic Beanstalk

L'applicazione potrebbe avere dipendenze su alcuni moduli Node.js, come quelli specificati nelle istruzioni `require()`. Questi moduli sono archiviati in una directory `node_modules`. Quando l'applicazione viene eseguita, Node.js carica i moduli da questa directory. Per ulteriori informazioni, consulta Caricamento dalle cartelle node_modules nella documentazione di Node.js.

È possibile specificare queste dipendenze del modulo utilizzando un file `package.json`. Se Elastic Beanstalk rileva questo file e non è presente una directory `node_modules`, Elastic Beanstalk esegue `npm install` come utente webapp. Il comando `npm install` installa le dipendenze nella directory `node_modules`, che Elastic Beanstalk crea in precedenza. Il comando `npm install` accede ai pacchetti elencati nel file `package.json` dal registro pubblico di npm o da altre posizioni. Per ulteriori informazioni, consulta il sito Web della documentazione di npm.

Se Elastic Beanstalk rileva la directory `node_modules`, allora non esegue `npm install` anche se esiste un file `package.json`. Elastic Beanstalk presuppone che i pacchetti di dipendenza siano disponibili nella directory `node_modules` perché Node.js possa accedere ed eseguire il caricamento.

Nelle sezioni seguenti vengono fornite ulteriori informazioni sulla definizione delle dipendenze tra moduli Node.js per l'applicazione.

> **Note**
>
> Se quando Elastic Beanstalk esegue `npm install` si verificano dei problemi di implementazione, prendi in considerazione un approccio alternativo. Includi la directory `node_modules` con i moduli di dipendenza nel bundle di origine dell'applicazione. In questo modo è possibile aggirare i problemi relativi all'installazione delle dipendenze dal registro pubblico di npm mentre si esamina il problema. Poiché i moduli di dipendenza provengono da una directory locale, ciò potrebbe anche aiutare a ridurre i tempi di implementazione. Per ulteriori informazioni, consulta Inclusione delle dipendenze Node.js in una directory node_modules

Specifica delle dipendenze Node.js con un file package.json

Utilizza un file `package.json` nella root fonte del progetto per specificare i pacchetti di dipendenza e per fornire un comando start. Quando è presente un file `package.json` e una directory `node_modules` non è presente nella root dell'origine del progetto, Elastic Beanstalk esegue `npm`

`install` come utente webapp per installare le dipendenze dal registro pubblico di npm. Elastic Beanstalk utilizza anche il comando `start` per avviare l'applicazione. Per ulteriori informazioni sul file `package.json`, consulta Specifica delle dipendenze in un file `package.json` sul sito Web della documentazione di npm.

Usa la parola chiave `scripts` per dare un comando di avvio. La parola chiave `scripts` viene ora utilizzata al posto dell'opzione `NodeCommand` legacy nello spazio dei nomi `aws:elasticbeanstalk:container:nodejs`.

Example package.json: Express

```
{
    "name": "my-app",
    "version": "0.0.1",
    "private": true,
    "dependencies": {
      "ejs": "latest",
      "aws-sdk": "latest",
      "express": "latest",
      "body-parser": "latest"
    },
    "scripts": {
      "start": "node app.js"
    }
  }
```

Dipendenze tra modalità di produzione e sviluppo

Per specificare le dipendenze nel file `package.json`, usa gli attributi dependencies e devDependencies. L'attributo dependencies indica i pacchetti richiesti dall'applicazione in produzione. L'attributo devDependencies indica i pacchetti necessari solo per lo sviluppo e il test locali.

Se è necessario installare i pacchetti devDependencies, imposta la proprietà di ambiente NPM_USE_PRODUCTION su `false`. Con questa impostazione, durante l'esecuzione di npm install non verranno utilizzate le opzioni precedenti. Ciò comporterà l'installazione dei pacchetti devDependencies.

Inclusione delle dipendenze Node.js in una directory node_modules

Per distribuire pacchetti di dipendenza nelle istanze di ambiente insieme al codice dell'applicazione, includili in una directory denominata `node_modules` nella directory root fonte del progetto. Per

ulteriori informazioni, consulta Download e installazione dei pacchetti in locale sul sito Web della documentazione di npm.

Quando distribuisci una `node_modules` directory in una versione della AL2 Node.js piattaforma AL2023/, Elastic Beanstalk presume che tu stia fornendo i tuoi pacchetti di dipendenze ed evita di installare le dipendenze specificate in un file package.json. Node.jscerca le dipendenze nella directory`node_modules`. Per ulteriori informazioni, consulta Caricamento dalle cartelle node_modules nella documentazione di Node.js.

> **Note**
>
> Se quando Elastic Beanstalk esegue `npm install` si verificano problemi di implementazione, considera l'utilizzo dell'approccio descritto in questo argomento come soluzione alternativa mentre analizzi il problema.

Considerazioni per Node.js su Amazon Linux 2

Leggi questa sezione se utilizzi un ramo della piattaforma Node.js in esecuzione su Amazon Linux 2.

Considerazioni per Node.js su Amazon Linux 2

> **Note**
>
> Le informazioni contenute in questo argomento si applicano ai rami Node.js della piattaforma in esecuzione su Amazon Linux 2. Il contenuto qui descrive caratteristiche e comportamenti AL2 specifici che differiscono da. AL2023

Varianti dei comandi

Le opzioni di comando variano a seconda della versione di npm inclusa nel ramo della piattaforma Amazon Linux 2 su cui viene eseguita l'applicazione.

- npm v6: per impostazione predefinita, Elastic Beanstalk installa le dipendenze in modalità di produzione. Usa il comando `npm install --production`.
- npm v7 o superiore: Elastic Beanstalk omette devDependencies. Usa il comando `npm install --omit=dev`.

Entrambi i comandi sopra elencati non installano i pacchetti che sono devDependencies.

Protocolli SSH e HTTPS per le dipendenze Git

A partire dalla versione della piattaforma Amazon Linux 2 del 7 marzo 2023, puoi utilizzare i protocolli SSH e HTTPS per recuperare i pacchetti da un repository Git. La ramificazione della piattaforma Node.js 16 supporta entrambi i protocolli SSH e HTTPS. Node.js 14 supporta solo il protocollo HTTPS.

Example package.json: Node.js 16 supporta sia HTTPS che SSH

```
    ...
    "dependencies": {
      "aws-sdk": "https://github.com/aws/aws-sdk-js.git",
      "aws-chime": "git+ssh://git@github.com:aws/amazon-chime-sdk-js.git"
    }
```

Versioni e gamme di versioni

Utilizza la parola chiave `engines` nel file `package.json` per specificare la versione di Node.js che desideri venga usata dalla tua applicazione. È inoltre possibile specificare un intervallo di versioni utilizzando la notazione npm. Per ulteriori informazioni sulla sintassi per gli intervalli di versioni, consulta Controllo delle versioni della semantica tramite npm sul sito Web di Node.js. La parola chiave `engines` nel file `package.json` di Node.js sostituisce l'opzione `NodeVersion` legacy nello spazio dei nomi `aws:elasticbeanstalk:container:nodejs`.

> **Important**
>
> La funzionalità per specificare gli intervalli di versioni non è disponibile per i rami della piattaforma Node.js in esecuzione. AL2023 Supportiamo solo una versione di Node.js all'interno di uno specifico ramo Node.js su AL2023. Se il file `package.json` specifica un intervallo di versioni, lo ignoreremo e per impostazione predefinita verrà impostata la ramificazione della piattaforma di Node.js.

Example **`package.json`**: la singola versione di Node.js

```
{
    ...
```

```
    "engines": { "node" : "14.16.0" }
  }
```

Example **`package.json`**: la gamma di versioni di Node.js

```
{
    ...
    "engines": { "node" : ">=10 <11" }
  }
```

Quando viene indicato un intervallo di versioni, Elastic Beanstalk installa nell'intervallo l'ultima versione Node.js disponibile sulla piattaforma. In questo esempio, l'intervallo indica che la versione deve essere maggiore o uguale alla versione 10, ma inferiore alla versione 11. Di conseguenza, Elastic Beanstalk installa l'ultima versione 10.x.y di Node.js, disponibile sulla piattaforma supportata.

Tieni presente che è possibile specificare solo una versione di Node.js corrispondente al ramo della piattaforma. Ad esempio, se utilizzi la ramificazione della piattaforma Node.js 16, sarà possibile specificare solo una versione 16.x.y di Node.js. È possibile utilizzare le opzioni dell'intervallo di versioni supportate da npm, per consentire una maggiore flessibilità. Per le versioni valide di Node.js per ogni ramo della piattaforma, consulta Node.js nella guida Piattaforme AWS Elastic Beanstalk .

> **Note**
>
> Quando il supporto della versione di Node.js in uso viene rimosso dalla piattaforma, è necessario modificare o rimuovere l'impostazione della versione di Node.js prima di eseguire un aggiornamento della piattaforma. Questo può verificarsi quando viene rilevata una vulnerabilità di sicurezza per una o più versioni di Node.js
> Quando ciò si verifica, i tentativi di aggiornamento della piattaforma a una nuova versione che non supporta la versione di Node.js configurata avranno esito negativo. Per evitare la necessità di creare un nuovo ambiente, modificare l'impostazione della versione Node.js in `package.json` a una versione Node.js supportata sia dalla vecchia versione della piattaforma che da quella nuova. È possibile specificare un intervallo di versioni Node.js che include una versione supportata, come descritto in precedenza in questo argomento. Hai anche la possibilità di rimuovere l'impostazione e quindi distribuire il nuovo bundle fonte.

Blocco delle dipendenze con Elastic npm shrinkwrap Beanstalk

La piattaforma Node.js esegue `npm install` come utente webapp a ogni implementazione. Eventuali nuove versioni delle dipendenze disponibili vengono installate durante l'implementazione, che pertanto potrebbe richiedere molto tempo.

È possibile evitare l'upgrade delle dipendenze creando un file `npm-shrinkwrap.json` che blocca le dipendenze dell'applicazione alla versione corrente.

```
$ npm install
$ npm shrinkwrap
wrote npm-shrinkwrap.json
```

Includi questo file nel bundle di origine per essere sicuro che le dipendenze vengano installate una sola volta.

Configurazione del server proxy

Elastic Beanstalk può utilizzare NGINX o HTTPD di Apache come proxy inverso per mappare l'applicazione al sistema di bilanciamento del carico di Elastic Load Balancing sulla porta 80. Il valore predefinito è NGINX. Elastic Beanstalk fornisce una configurazione proxy predefinita che puoi estendere o ignorare completamente in favore di quella personalizzata.

Per impostazione predefinita, Elastic Beanstalk configura il proxy per inoltrare le richieste all'applicazione sulla porta 5000. È possibile sostituire la porta predefinita impostando la `PORT`proprietà dell'ambiente sulla porta di ascolto dell'applicazione principale.

> **Note**
>
> La porta di ascolto della tua applicazione non influisce sulla porta che il server NGINX ascolta per ricevere le richieste dal sistema di bilanciamento del carico.

Configurazione del server proxy sulla versione della piattaforma

Tutte le AL2 piattaforme AL2023/supportano una funzionalità di configurazione proxy uniforme. Per ulteriori informazioni sulla configurazione del server proxy sulle versioni della piattaforma che eseguono AL2023/AL2, consultaConfigurazione del proxy inverso.

Configurazione del proxy su AMI Amazon Linux (precedente ad Amazon Linux 2)

Se l'ambiente Node.js di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni presenti in questa sezione.

> **Note**
>
> - Le informazioni contenute in questo argomento si applicano solo ai rami della piattaforma basati su Amazon Linux AMI (AL1). AL2023I rami AL2 della piattaforma/sono incompatibili con le versioni precedenti della piattaforma Amazon Linux AMI (AL1) e richiedono impostazioni di configurazione diverse.
> - Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Estensione e sovrascrittura della configurazione proxy predefinita — Amazon Linux AMI () AL1

La piattaforma Node.js utilizza un proxy inverso per l'inoltro delle richieste provenienti dalla porta 80 dell'istanza all'applicazione in ascolto sulla porta 8081. Elastic Beanstalk fornisce una configurazione proxy predefinita che puoi estendere o ignorare completamente in favore di quella personalizzata.

Per estendere la configurazione predefinita, aggiungi file `.conf` a `/etc/nginx/conf.d` con un file di configurazione. Per un esempio specifico, consulta Terminazione di HTTPS nelle istanze EC2 che eseguono Node.js.

La piattaforma Node.js imposta la variabile di ambiente PORT sulla porta a cui il server proxy trasferisce il traffico. Leggi questa variabile nel codice per configurare la porta della tua applicazione.

```
var port = process.env.PORT || 3000;

var server = app.listen(port, function () {
    console.log('Server running at http://127.0.0.1:' + port + '/');
});
```

La configurazione NGINX predefinita inoltra il traffico a un server upstream denominato `nodejs` su `127.0.0.1:8081`. È possibile rimuovere la configurazione predefinita e fornire quella personalizzata in un file di configurazione.

Example.ebextensions/proxy.config

L'esempio seguente rimuove la configurazione predefinita e aggiunge una configurazione personalizzata che inoltra il traffico alla porta 5000 anziché alla porta 8081.

```
files:
  /etc/nginx/conf.d/proxy.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      upstream nodejs {
        server 127.0.0.1:5000;
        keepalive 256;
      }

      server {
        listen 8080;

        if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
            set $year $1;
            set $month $2;
            set $day $3;
            set $hour $4;
        }
        access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
 healthd;
        access_log  /var/log/nginx/access.log  main;

        location / {
            proxy_pass  http://nodejs;
            proxy_set_header   Connection "";
            proxy_http_version 1.1;
            proxy_set_header        Host            $host;
            proxy_set_header        X-Real-IP       $remote_addr;
            proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        gzip on;
```

```
        gzip_comp_level 4;
        gzip_types text/html text/plain text/css application/json application/x-
javascript text/xml application/xml application/xml+rss text/javascript;

        location /static {
            alias /var/app/current/static;
        }

      }

  /opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh:
    mode: "000755"
    owner: root
    group: root
    content: |
      #!/bin/bash -xe
      rm -f /etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf
      service nginx stop
      service nginx start

container_commands:
  removeconfig:
    command: "rm -f /tmp/deployment/config/
#etc#nginx#conf.d#00_elastic_beanstalk_proxy.conf /etc/nginx/
conf.d/00_elastic_beanstalk_proxy.conf"
```

La configurazione di esempio, `/etc/nginx/conf.d/proxy.conf`, utilizza la configurazione predefinita specificata in `/etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf` come base per includere il blocco di server predefinito con la compressione e le impostazioni dei log, nonché una mappatura di file statici.

Il comando `removeconfig` rimuove la configurazione predefinita per il container, in modo che il server proxy utilizzi la configurazione personalizzata. Elastic Beanstalk ricrea la configurazione predefinita una volta implementate tutte le configurazioni. Per tenere conto di ciò, nell'esempio seguente viene aggiunto un post-configuration-deployment hook (`/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh`). In questo modo viene rimossa la configurazione predefinita e riavvia il server proxy.

> **Note**
>
> La configurazione predefinita potrebbe cambiare nelle versioni future della piattaforma Node.js. Per garantire la compatibilità, utilizza la versione più recente della configurazione come base per le personalizzazioni.

Se sovrascrivi la configurazione predefinita, devi definire eventuali mappature di file statici e compressione GZIP. Questo perché la piattaforma non può applicare le impostazioni standard.

Altri esempi di applicazioni e tutorial di Elastic Beanstalk per Node.js

Questa sezione fornisce applicazioni e tutorial aggiuntivi. L'QuickStart per Node.jsargomento riportato in precedenza in questo argomento illustra come avviare l'applicazione di esempio Node.js con l'EB CLI.

Per iniziare a utilizzare le applicazioni Node.js AWS Elastic Beanstalk, è sufficiente un pacchetto di sorgenti dell'applicazione da caricare come prima versione dell'applicazione e da distribuire in un ambiente.

Avvio di un ambiente con un'applicazione Node.js di esempio

Elastic Beanstalk fornisce applicazioni di esempio a pagina singola per ogni piattaforma, nonché esempi più complessi che mostrano l'uso di risorse AWS aggiuntive come Amazon RDS e funzionalità specifiche del linguaggio o della piattaforma e. APIs

> **Note**
>
> Segui i passaggi nel file del pacchetto di origine `README.md` per implementarlo.

Esempi

Tipo di ambiente	Bundle di origine	Description
Server Web	nodejs.zip	Applicazione a pagina singola. Per avviare l'applicazione di esempio con l'EB CLI, vedere. QuickStart per Node.js Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.
Server Web con Amazon RDS	nodejs-example-express-rds.zip	Applicazione Hiking Log che usa il framework Express e un Amazon Relational Database Service (RDS). Tutorial
Server Web con Amazon ElastiCache	nodejs-example-express-elasticache.zip	Applicazione web Express che utilizza Amazon ElastiCache per il clustering. Il clustering migliora l'alta disponibilità, le prestazioni e la sicurezza della tua applicazione Web. Tutorial
Server Web con DynamoDB, Amazon SNS e Amazon SQS	nodejs-example-dynamo.zip	Sito Web Express che consente di raccogliere informazioni di contatto degli utenti per una campagna di marketing di una nuova azienda. Utilizza l' AWS SDK for JavaScript in Node.js per scrivere voci in una tabella DynamoDB e i file di configurazione Elastic Beanstalk per creare risorse in DynamoDB, Amazon SNS e Amazon SQS. Tutorial

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione completamente diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Per informazioni dettagliate sulla distribuzione dell'applicazione, consulta Distribuzione di una nuova versione dell'applicazione.

Dopo aver distribuito una o due applicazioni di esempio e aver iniziato a sviluppare ed eseguire applicazioni Node.js localmente, consulta la procedura Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk per configurare un ambiente di sviluppo Node.js con tutti gli strumenti necessari.

Distribuzione di un'applicazione Node.js Express su Elastic Beanstalk

Questa sezione illustra come distribuire un'applicazione di esempio su Elastic Beanstalk utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) e quindi aggiornare l'applicazione per utilizzare il framework Express.

Prerequisiti

Di seguito sono elencati i requisiti per questo tutorial:

- I runtime Node.js
- Il software predefinito per la gestione dei pacchetti Node.js, npm
- Il generatore di righe di comando Express
- interfaccia a riga di comando Elastic Beanstalk (CLI EB)

Per ulteriori informazioni sull'installazione dei primi tre componenti riportati e la configurazione dell'ambiente di sviluppo locale, consulta Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk. Per questo tutorial, non è necessario installare l' AWS SDK per Node.js, anch'esso menzionato nell'argomento a cui si fa riferimento.

Per maggiori dettagli sull'installazione e la configurazione della CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Creazione di un ambiente Elastic Beanstalk

La tua directory delle applicazioni

Questo tutorial utilizza una directory chiamata `nodejs-example-express-rds` per il pacchetto sorgente dell'applicazione. Crea la directory `nodejs-example-express-rds` per questo tutorial.

```
~$ mkdir nodejs-example-express-rds
```

> **Note**
>
> Ogni tutorial in questo capitolo utilizza la propria directory per il pacchetto sorgente dell'applicazione. Il nome della directory corrisponde al nome dell'applicazione di esempio utilizzata dal tutorial.

Cambia la tua directory di lavoro corrente in `nodejs-example-express-rds`.

```
~$ cd nodejs-example-express-rds
```

Configura ora un ambiente Elastic Beanstalk in cui viene eseguita la piattaforma Node.js e l'applicazione di esempio. Utilizzeremo l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB).

Configurazione di un repository della CLI EB per l'applicazione e creaione di un ambiente Elastic Beanstalk in cui viene eseguita la piattaforma Node.js

1. Creazione di un repository con il comando eb init.

   ```
   ~/nodejs-example-express-rds$ eb init --platform node.js --region <region>
   ```

   Questo comando crea un file di configurazione in una cartella denominata `.elasticbeanstalk`, che specifica le impostazioni per la creazione di ambienti per l'applicazione e crea un'applicazione Elastic Beanstalk che prende nome dalla cartella corrente.

2. Crea un ambiente che esegue un'applicazione di esempio con il comando eb create.

   ```
   ~/nodejs-example-express-rds$ eb create --sample nodejs-example-express-rds
   ```

Questo comando crea un ambiente con bilanciamento del carico con le impostazioni predefinite per la piattaforma Node.js e le risorse seguenti:

- EC2 istanza: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per eseguire app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza dell'istanza: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa consente al traffico HTTP proveniente dal sistema di bilanciamento del carico di raggiungere l' EC2 istanza su cui è in esecuzione la tua app web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza Load Balancer: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.

- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

3. Quando la creazione dell'ambiente è completa, utilizza il comando eb open per aprire l'URL dell'ambiente nel browser predefinito.

```
~/nodejs-example-express-rds$ eb open
```

È stato ora creato un ambiente Elastic Beanstalk Node.js con un'applicazione di esempio. A questo punto puoi aggiornarlo con la tua applicazione. Successivamente, aggiorniamo l'applicazione di esempio per utilizzare il framework Express.

Aggiornamento dell'applicazione per usare Express

Una volta creato un ambiente con un'applicazione di esempio, potrai aggiornarlo con la tua applicazione. In questa procedura, eseguiamo innanzitutto i comandi express e npm install per configurare il framework Express nella directory dell'applicazione. Quindi usa la CLI EB per aggiornare l'ambiente Elastic Beanstalk con l'applicazione aggiornata.

Per aggiornare l'applicazione per usare Express

1. Esegui il comando `express`. Questo genera `package.json`, `app.js` e poche altre directory.

```
~/nodejs-example-express-rds$ express
```

   Quando ti viene chiesto se desideri proseguire, digita **y**.

> **Note**
>
> Se il comando express non funziona, è possibile che non sia stato installato il generatore di riga di comando Express come descritto nella precedente sezione Prerequisiti. Oppure potrebbe essere necessario configurare l'impostazione del percorso della directory per il computer locale in modo da eseguire il comando express. Consulta la sezione Prerequisiti per i passaggi dettagliati sulla configurazione dell'ambiente di sviluppo, in modo da poter procedere con questo tutorial.

2. Configura le dipendenze locali.

```
~/nodejs-example-express-rds$ npm install
```

3. (Facoltativo) Verifica dell'avvio del server dell'applicazione Web.

```
~/nodejs-example-express-rds$ npm start
```

Verrà visualizzato un output simile al seguente:

```
> nodejs@0.0.0 start /home/local/user/node-express
> node ./bin/www
```

Il server viene eseguito sulla porta 3000 per impostazione predefinita. Per testarlo, esegui `curl http://localhost:3000` su un altro terminale oppure apri un browser sul computer locale e immetti l'indirizzo URL `http://localhost:3000`.

Premi Ctrl+C per arrestare il server.

4. Implementa le modifiche apportate all'ambiente Elastic Beanstalk con il comando eb deploy.

```
~/nodejs-example-express-rds$ eb deploy
```

5. Quando l'ambiente è verde e pronto, aggiorna l'URL per verificare che funziona. Visualizzerai una pagina Web con la dicitura Welcome to Express (Benvenuto in Express).

Successivamente, aggiorniamo l'applicazione Express per distribuire i file statici e aggiungere una nuova pagina.

Per configurare i file statici e aggiungere una nuova pagina all'applicazione Express

1. Aggiungi un secondo file di configurazione nella cartella `.ebextensions` con il seguente contenuto:

   `nodejs-example-express-rds/.ebextensions/staticfiles.config`

   ```
   option_settings:
       aws:elasticbeanstalk:environment:proxy:staticfiles:
           /stylesheets: public/stylesheets
   ```

   In base a questa configurazione il server proxy serve i file nella cartella `public` al percorso `/public` dell'applicazione. Servire i file staticamente dal proxy riduce il carico di lavoro per l'applicazione. Per ulteriori informazioni, consulta la sezione File statici descritta precedentemente in questo capitolo.

2. (Facoltativo) Per essere certi che le mappature statiche siano configurate correttamente, commenta la configurazione della mappatura statica in `nodejs-example-express-rds/app.js`. Ciò rimuoverà la mappatura dall'applicazione del nodo.

   ```
   //  app.use(express.static(path.join(__dirname, 'public')));
   ```

   Le mappature statiche del file `staticfiles.config` del passaggio precedente dovrebbero comunque caricare correttamente il foglio di stile anche dopo aver commentato questa riga. Per verificare che le mappature dei file statici vengano caricate tramite la configurazione del file statico del proxy anziché tramite l'applicazione Express, rimuovi i valori che seguono `option_settings:`. Una volta rimosso sia dalla configurazione del file statico che dall'applicazione del nodo, il foglio di stile non verrà caricato.

   Una volta terminato il test, ricordati di reimpostare il contenuto di `nodejs-example-express-rds/app.js` e `staticfiles.config`.

3. Add `nodejs-example-express-rds/routes/hike.js`. Digita quanto segue:

   ```
   exports.index = function(req, res) {
    res.render('hike', {title: 'My Hiking Log'});
   };

   exports.add_hike = function(req, res) {
   };
   ```

4. Aggiorna `nodejs-example-express-rds/app.js` per includere tre nuove righe.

   In primo luogo, aggiungi la riga seguente per aggiungere un `require` a questo percorso:

   ```
   var hike = require('./routes/hike');
   ```

   Il file si presenta in maniera simile al seguente frammento:

   ```
   var express = require('express');
   var path = require('path');
   var hike = require('./routes/hike');
   ```

   Quindi, aggiungi le seguenti due righe a `nodejs-example-express-rds/app.js` dopo `var app = express();`

   ```
   app.get('/hikes', hike.index);
   app.post('/add_hike', hike.add_hike);
   ```

   Il file si presenta in maniera simile al seguente frammento:

   ```
   var app = express();
   app.get('/hikes', hike.index);
   app.post('/add_hike', hike.add_hike);
   ```

5. Copia `nodejs-example-express-rds/views/index.jade` su `nodejs-example-express-rds/views/hike.jade`.

   ```
   ~/nodejs-example-express-rds$ cp views/index.jade views/hike.jade
   ```

6. Implementa le modifiche con il comando eb deploy.

   ```
   ~/nodejs-example-express-rds$ eb deploy
   ```

7. Il tuo ambiente sarà aggiornato dopo qualche minuto. Quando l'ambiente è verde e pronto, verifica che funzioni aggiornando il browser e aggiungendo **`hikes`** al termine dell'URL (ad esempio `http://node-express-env-syypntcz2q.elasticbeanstalk.com/hikes`).

   Devi visualizzare una pagina Web intitolata My Hiking Log (Il mio log di Hiking).

A questo punto, hai creato un'applicazione Web che utilizza il framework Express. Nella prossima sezione, modificheremo l'applicazione per utilizzare un Amazon Relational Database Service (RDS) per archiviare un log hiking.

Aggiornamento dell'applicazione per usare Amazon RDS.

In questo passaggio successivo aggiorniamo l'applicazione in modo da utilizzare Amazon RDS per MySQL.

Aggiornamento dell'applicazione per usare RDS per MySQL

1. Per creare un database RDS per MySQL accoppiato all'ambiente Elastic Beanstalk, segui le istruzioni riportate nell'argomento Aggiunta di un database più avanti in questo capitolo. L'aggiunta di un'istanza del database richiede circa 10 minuti.
2. Aggiorna la sezione delle dipendenze nel file `package.json` con i seguenti contenuti:

```
"dependencies": {
    "async": "^3.2.4",
    "express": "4.18.2",
    "jade": "1.11.0",
    "mysql": "2.18.1",
    "node-uuid": "^1.4.8",
    "body-parser": "^1.20.1",
    "method-override": "^3.0.0",
    "morgan": "^1.10.0",
    "errorhandler": "^1.5.1"
  }
```

3. Esegui npm install.

```
~/nodejs-example-express-rds$ npm install
```

4. Aggiorna `app.js` per connetterti al database, creare una tabella e inserire un singolo log hiking predefinito. Ogni volta che questa app viene implementata, eliminerà la tabella di hiking precedente e la ricreerà.

```
/**
 * Module dependencies.
 */

 const express = require('express')
```

```
 , routes = require('./routes')
 , hike = require('./routes/hike')
 , http = require('http')
 , path = require('path')
 , mysql = require('mysql')
 , async = require('async')
 , bodyParser = require('body-parser')
 , methodOverride = require('method-override')
 , morgan = require('morgan')
 , errorhandler = require('errorhandler');

const { connect } = require('http2');

const app = express()

app.set('views', __dirname + '/views')
app.set('view engine', 'jade')
app.use(methodOverride())
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({ extended: true }))
app.use(express.static(path.join(__dirname, 'public')))


app.set('connection', mysql.createConnection({
host: process.env.RDS_HOSTNAME,
user: process.env.RDS_USERNAME,
password: process.env.RDS_PASSWORD,
port: process.env.RDS_PORT}));

function init() {
 app.get('/', routes.index);
 app.get('/hikes', hike.index);
 app.post('/add_hike', hike.add_hike);
}

const client = app.get('connection');
async.series([
 function connect(callback) {
   client.connect(callback);
   console.log('Connected!');
 },
 function clear(callback) {
   client.query('DROP DATABASE IF EXISTS mynode_db', callback);
 },
```

```
 function create_db(callback) {
   client.query('CREATE DATABASE mynode_db', callback);
 },
 function use_db(callback) {
   client.query('USE mynode_db', callback);
 },
 function create_table(callback) {
    client.query('CREATE TABLE HIKES (' +
                        'ID VARCHAR(40), ' +
                        'HIKE_DATE DATE, ' +
                        'NAME VARCHAR(40), ' +
                        'DISTANCE VARCHAR(40), ' +
                        'LOCATION VARCHAR(40), ' +
                        'WEATHER VARCHAR(40), ' +
                        'PRIMARY KEY(ID))', callback);
 },
 function insert_default(callback) {
   const hike = {HIKE_DATE: new Date(), NAME: 'Hazard Stevens',
         LOCATION: 'Mt Rainier', DISTANCE: '4,027m vertical', WEATHER:'Bad', ID:
 '12345'};
   client.query('INSERT INTO HIKES set ?', hike, callback);
 }
], function (err, results) {
 if (err) {
   console.log('Exception initializing database.');
   throw err;
 } else {
   console.log('Database initialization complete.');
   init();
 }
});

module.exports = app
```

5. Aggiungi il seguente contenuto a `routes/hike.js`. Ciò consentirà ai percorsi di inserire nuovi log hiking nel database HIKES.

```
const uuid = require('node-uuid');
exports.index = function(req, res) {
  res.app.get('connection').query( 'SELECT * FROM HIKES', function(err,
rows) {
    if (err) {
      res.send(err);
```

```
    } else {
      console.log(JSON.stringify(rows));
      res.render('hike', {title: 'My Hiking Log', hikes: rows});
  }});
};
exports.add_hike = function(req, res){
  const input = req.body.hike;
  const hike = { HIKE_DATE: new Date(), ID: uuid.v4(), NAME: input.NAME,
  LOCATION: input.LOCATION, DISTANCE: input.DISTANCE, WEATHER: input.WEATHER};
  console.log('Request to log hike:' + JSON.stringify(hike));
  req.app.get('connection').query('INSERT INTO HIKES set ?', hike, function(err) {
      if (err) {
        res.send(err);
      } else {
        res.redirect('/hikes');
      }
   });
};
```

6. Sostituisci il contenuto di `routes/index.js` con quanto riportato di seguito:

```
/*
 * GET home page.
 */

exports.index = function(req, res){
  res.render('index', { title: 'Express' });
};
```

7. Aggiungi il seguente modello jade a `views/hike.jade` per fornire l'interfaccia utente per l'aggiunta di log hiking.

```
extends layout

block content
  h1= title
  p Welcome to #{title}

  form(action="/add_hike", method="post")
    table(border="1")
      tr
        td Your Name
        td
```

```
          input(name="hike[NAME]", type="textbox")
      tr
        td Location
        td
          input(name="hike[LOCATION]", type="textbox")
      tr
        td Distance
        td
          input(name="hike[DISTANCE]", type="textbox")
      tr
        td Weather
        td
          input(name="hike[WEATHER]", type="radio", value="Good")
          | Good
          input(name="hike[WEATHER]", type="radio", value="Bad")
          | Bad
          input(name="hike[WEATHER]", type="radio", value="Seattle", checked)
          | Seattle
      tr
        td(colspan="2")
          input(type="submit", value="Record Hike")

  div
    h3 Hikes
    table(border="1")
      tr
        td Date
        td Name
        td Location
        td Distance
        td Weather
      each hike in hikes
        tr
          td #{hike.HIKE_DATE.toDateString()}
          td #{hike.NAME}
          td #{hike.LOCATION}
          td #{hike.DISTANCE}
          td #{hike.WEATHER}
```

8. Implementa le modifiche con il comando eb deploy.

```
~/nodejs-example-express-rds$ eb deploy
```

Eliminazione

Se hai finito di lavorare con Elastic Beanstalk, puoi terminare l'ambiente.

Utilizza il comando eb terminate per terminare l'ambiente e tutte le risorse che contiene.

```
~/nodejs-example-express-rds$ eb terminate
The environment "nodejs-example-express-rds-env" and all associated instances will be
 terminated.
To confirm, type the environment name: nodejs-example-express-rds-env
INFO: terminateEnvironment is starting.
...
```

Distribuzione di un'applicazione Node.js Express con clustering su Elastic Beanstalk

Questo tutorial illustra la distribuzione di un'applicazione di esempio su Elastic Beanstalk utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) e quindi l'aggiornamento dell'applicazione per utilizzare il framework Express, Amazon e il clustering. ElastiCache Il clustering migliora l'alta disponibilità, le prestazioni e la sicurezza della tua applicazione Web. Per ulteriori informazioni su Amazon ElastiCache, consulta What Is Amazon ElastiCache (Memcached)? nella Guida per l'utente di Amazon ElastiCache (Memcached).

> **Note**
>
> Questo esempio crea AWS risorse che potrebbero esserti addebitate. Per ulteriori informazioni sui AWS prezzi, consultahttps://aws.amazon.com/pricing/. Alcuni servizi fanno parte del piano di utilizzo AWS gratuito. Se sei un nuovo cliente, puoi provare questi servizi gratuitamente. Per ulteriori informazioni, consultare https://aws.amazon.com/free/.

Prerequisiti

Di seguito sono elencati i requisiti per questo tutorial:

- I runtime Node.js
- Il software predefinito per la gestione dei pacchetti Node.js, npm
- Il generatore di righe di comando Express

- interfaccia a riga di comando Elastic Beanstalk (CLI EB)

Per ulteriori informazioni sull'installazione dei primi tre componenti riportati e la configurazione dell'ambiente di sviluppo locale, consulta Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk. Per questo tutorial, non è necessario installare l' AWS SDK per Node.js, anch'esso menzionato nell'argomento di riferimento.

Per maggiori dettagli sull'installazione e la configurazione della CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Creazione di un ambiente Elastic Beanstalk

La tua directory delle applicazioni

Questo tutorial utilizza una directory chiamata `nodejs-example-express-elasticache` per il pacchetto sorgente dell'applicazione. Crea la directory `nodejs-example-express-elasticache` per questo tutorial.

```
~$ mkdir nodejs-example-express-elasticache
```

> **Note**
>
> Ogni tutorial in questo capitolo utilizza la propria directory per il pacchetto sorgente dell'applicazione. Il nome della directory corrisponde al nome dell'applicazione di esempio utilizzata dal tutorial.

Cambia la tua directory di lavoro corrente in `nodejs-example-express-elasticache`.

```
~$ cd nodejs-example-express-elasticache
```

Configura ora un ambiente Elastic Beanstalk in cui viene eseguita la piattaforma Node.js e l'applicazione di esempio. Utilizzeremo l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB).

Configurazione di un repository della CLI EB per l'applicazione e creaione di un ambiente Elastic Beanstalk in cui viene eseguita la piattaforma Node.js

1. Creazione di un repository con il comando eb init.

```
~/nodejs-example-express-elasticache$ eb init --platform node.js --region <region>
```

Questo comando crea un file di configurazione in una cartella denominata `.elasticbeanstalk`, che specifica le impostazioni per la creazione di ambienti per l'applicazione e crea un'applicazione Elastic Beanstalk che prende nome dalla cartella corrente.

2. Crea un ambiente che esegue un'applicazione di esempio con il comando eb create.

```
~/nodejs-example-express-elasticache$ eb create --sample nodejs-example-express-
elasticache
```

Questo comando crea un ambiente con bilanciamento del carico con le impostazioni predefinite per la piattaforma Node.js e le risorse seguenti:

- Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.

- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

3. Quando la creazione dell'ambiente è completa, utilizza il comando eb open per aprire l'URL dell'ambiente nel browser predefinito.

```
~/nodejs-example-express-elasticache$ eb open
```

È stato ora creato un ambiente Elastic Beanstalk Node.js con un'applicazione di esempio. A questo punto puoi aggiornarlo con la tua applicazione. Successivamente, aggiorniamo l'applicazione di esempio per utilizzare il framework Express.

Aggiornamento dell'applicazione per usare Express

Aggiornare l'applicazione di esempio nell'ambiente Elastic Beanstalk per utilizzare il framework Express.

Puoi scaricare il codice sorgente finale da .zip. nodejs-example-express-elasticache

Per aggiornare l'applicazione per usare Express

Una volta creato un ambiente con un'applicazione di esempio, potrai aggiornarlo con la tua applicazione. In questa procedura, eseguiamo innanzitutto i comandi express e npm install per configurare il framework Express nella directory dell'applicazione.

1. Esegui il comando `express`. Questo genera `package.json`, `app.js` e poche altre directory.

   ```
   ~/nodejs-example-express-elasticache$ express
   ```

   Quando ti viene chiesto se desideri proseguire, digita **y**.

   > **Note**
   >
   > Se il comando express non funziona, è possibile che non sia stato installato il generatore di riga di comando Express come descritto nella precedente sezione Prerequisiti. Oppure potrebbe essere necessario configurare l'impostazione del percorso della directory per il computer locale in modo da eseguire il comando express. Consulta la sezione Prerequisiti per i passaggi dettagliati sulla configurazione dell'ambiente di sviluppo, in modo da poter procedere con questo tutorial.

2. Configura le dipendenze locali.

   ```
   ~/nodejs-example-express-elasticache$ npm install
   ```

3. (Facoltativo) Verifica dell'avvio del server dell'applicazione Web.

   ```
   ~/nodejs-example-express-elasticache$ npm start
   ```

   Verrà visualizzato un output simile al seguente:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   > node ./bin/www
   ```

   Il server viene eseguito sulla porta 3000 per impostazione predefinita. Per testarlo, esegui `curl http://localhost:3000` su un altro terminale oppure apri un browser sul computer locale e immetti l'indirizzo URL `http://localhost:3000`.

   Premi Ctrl+C per arrestare il server.

4. Rinomina `nodejs-example-express-elasticache/app.js` come `nodejs-example-express-elasticache/express-app.js`.

```
~/nodejs-example-express-elasticache$ mv app.js express-app.js
```

5. Aggiornare la riga `var app = express();` in `nodejs-example-express-elasticache/express-app.js` sui seguenti elementi.

```
var app = module.exports = express();
```

6. Sul computer locale, creare un file denominato `nodejs-example-express-elasticache/app.js` con il seguente codice.

```
/**
 * Module dependencies.
 */

 const express = require('express'),
 session = require('express-session'),
 bodyParser = require('body-parser'),
 methodOverride = require('method-override'),
 cookieParser = require('cookie-parser'),
 fs = require('fs'),
 filename = '/var/nodelist',
 app = express();

let MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
 app.use(bodyParser.raw());
 app.use(methodOverride());
 if (cacheNodes.length > 0) {
   app.use(cookieParser());

   console.log('Using memcached store nodes:');
   console.log(cacheNodes);

   app.use(session({
     secret: 'your secret here',
     resave: false,
     saveUninitialized: false,
     store: new MemcachedStore({ 'hosts': cacheNodes })
```

```
    }));
  } else {
    console.log('Not using memcached store.');
    app.use(session({
      resave: false,
      saveUninitialized: false, secret: 'your secret here'
    }));
  }

  app.get('/', function (req, resp) {
    if (req.session.views) {
      req.session.views++
      resp.setHeader('Content-Type', 'text/html')
      resp.send(`You are session: ${req.session.id}. Views: ${req.session.views}`)
    } else {
      req.session.views = 1
      resp.send(`You are session: ${req.session.id}. No views yet, refresh the page!
`)
    }
  });

  if (!module.parent) {
    console.log('Running express without cluster. Listening on port %d',
  process.env.PORT || 5000)
    app.listen(process.env.PORT || 5000)
  }
}

console.log("Reading elastic cache configuration")
// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function (err, data) {
  if (err) throw err;

  let cacheNodes = []
  if (data) {
    let lines = data.split('\n');
    for (let i = 0; i < lines.length; i++) {
      if (lines[i].length > 0) {
        cacheNodes.push(lines[i])
      }
    }
  }

  setup(cacheNodes)
```

```
});

module.exports = app;
```

7. Sostituisci il contenuto del file `nodejs-example-express-elasticache/bin/www` con quanto riportato di seguito:

```
#!/usr/bin/env node

/**
 * Module dependencies.
 */

const app = require('../app');
const cluster = require('cluster');
const debug = require('debug')('nodejs-example-express-elasticache:server');
const http = require('http');
const workers = {},
  count = require('os').cpus().length;

function spawn() {
  const worker = cluster.fork();
  workers[worker.pid] = worker;
  return worker;
}


/**
 * Get port from environment and store in Express.
 */

const port = normalizePort(process.env.PORT || '3000');
app.set('port', port);

if (cluster.isMaster) {
  for (let i = 0; i < count; i++) {
    spawn();
  }

  // If a worker dies, log it to the console and start another worker.
  cluster.on('exit', function (worker, code, signal) {
    console.log('Worker ' + worker.process.pid + ' died.');
    cluster.fork();
```

```
  });

  // Log when a worker starts listening
  cluster.on('listening', function (worker, address) {
    console.log('Worker started with PID ' + worker.process.pid + '.');
  });

} else {
  /**
   * Create HTTP server.
   */

  let server = http.createServer(app);

  /**
   * Event listener for HTTP server "error" event.
   */

  function onError(error) {
    if (error.syscall !== 'listen') {
      throw error;
    }

    const bind = typeof port === 'string'
      ? 'Pipe ' + port
      : 'Port ' + port;

    // handle specific listen errors with friendly messages
    switch (error.code) {
      case 'EACCES':
        console.error(bind + ' requires elevated privileges');
        process.exit(1);
        break;
      case 'EADDRINUSE':
        console.error(bind + ' is already in use');
        process.exit(1);
        break;
      default:
        throw error;
    }
  }

  /**
   * Event listener for HTTP server "listening" event.
```

```
   */

  function onListening() {
    const addr = server.address();
    const bind = typeof addr === 'string'
      ? 'pipe ' + addr
      : 'port ' + addr.port;
    debug('Listening on ' + bind);
  }

  /**
   * Listen on provided port, on all network interfaces.
   */

  server.listen(port);
  server.on('error', onError);
  server.on('listening', onListening);
}

/**
 * Normalize a port into a number, string, or false.
 */

function normalizePort(val) {
  const port = parseInt(val, 10);

  if (isNaN(port)) {
    // named pipe
    return val;
  }

  if (port >= 0) {
    // port number
    return port;
  }

  return false;
}
```

8. Implementa le modifiche apportate all'ambiente Elastic Beanstalk con il comando eb deploy.

```
~/nodejs-example-express-elasticache$ eb deploy
```

9. Il tuo ambiente sarà aggiornato dopo qualche minuto. Quando l'ambiente è verde e pronto, aggiorna l'URL per verificare che funziona. Visualizzerai una pagina Web con la dicitura Welcome to Express (Benvenuto in Express).

Puoi accedere ai log per le Istanza EC2 che eseguono l'applicazione. Per istruzioni su come accedere ai log, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Successivamente, aggiorniamo l'applicazione Express per utilizzare Amazon ElastiCache.

Per aggiornare la tua applicazione Express per utilizzare Amazon ElastiCache

1. Sul computer locale, creare una directory `.ebextensions` nella directory di primo livello del bundle di origine. In questo esempio viene utilizzato `nodejs-example-express-elasticache/.ebextensions`.
2. Creare un file di configurazione `nodejs-example-express-elasticache/.ebextensions/elasticache-iam-with-script.config` con il seguente snippet. Per ulteriori informazioni sul file di configurazione, consulta Spazi dei nomi per la configurazione di Node.js. In questo modo viene creato un utente IAM con le autorizzazioni necessarie per scoprire i nodi elasticache e le operazioni di scrittura su un file a ogni modifica della cache. Puoi anche copiare il file nodejs-example-express-elasticacheda.zip. Per ulteriori informazioni sulle ElastiCache proprietà, vedereEsempio: ElastiCache.

> **Note**
>
> YAML si basa sulla coerenza del rientro. Utilizza lo stesso livello di rientro quando sostituisci del contenuto in un file di configurazione di esempio e assicurati che il tuo editor di testo utilizzi gli spazi e non le tabulazioni per il rientro.

```
Resources:
  MyCacheSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: "Lock cache down to webserver access only"
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort:
```

```
            Fn::GetOptionSetting:
              OptionName: CachePort
              DefaultValue: 11211
          ToPort:
            Fn::GetOptionSetting:
              OptionName: CachePort
              DefaultValue: 11211
          SourceSecurityGroupName:
            Ref: AWSEBSecurityGroup
  MyElastiCache:
    Type: 'AWS::ElastiCache::CacheCluster'
    Properties:
      CacheNodeType:
        Fn::GetOptionSetting:
          OptionName: CacheNodeType
          DefaultValue: cache.t2.micro
      NumCacheNodes:
        Fn::GetOptionSetting:
          OptionName: NumCacheNodes
          DefaultValue: 1
      Engine:
        Fn::GetOptionSetting:
          OptionName: Engine
          DefaultValue: redis
      VpcSecurityGroupIds:
        -
          Fn::GetAtt:
            - MyCacheSecurityGroup
            - GroupId
  AWSEBAutoScalingGroup :
    Metadata :
      ElastiCacheConfig :
        CacheName :
          Ref : MyElastiCache
        CacheSize :
           Fn::GetOptionSetting:
             OptionName : NumCacheNodes
             DefaultValue: 1
  WebServerUser :
    Type : AWS::IAM::User
    Properties :
      Path : "/"
      Policies:
        -
```

```
          PolicyName: root
          PolicyDocument :
            Statement :
              -
                Effect : Allow
                Action :
                  - cloudformation:DescribeStackResource
                  - cloudformation:ListStackResources
                  - elasticache:DescribeCacheClusters
                Resource : "*"
  WebServerKeys :
    Type : AWS::IAM::AccessKey
    Properties :
      UserName :
        Ref: WebServerUser

Outputs:
  WebsiteURL:
    Description: sample output only here to show inline string function parsing
    Value: |
      http://`{ "Fn::GetAtt" : [ "AWSEBLoadBalancer", "DNSName" ] }`
  MyElastiCacheName:
    Description: Name of the elasticache
    Value:
      Ref : MyElastiCache
  NumCacheNodes:
    Description: Number of cache nodes in MyElastiCache
    Value:
      Fn::GetOptionSetting:
        OptionName : NumCacheNodes
        DefaultValue: 1

files:
  "/etc/cfn/cfn-credentials" :
    content : |
      AWSAccessKeyId=`{ "Ref" : "WebServerKeys" }`
      AWSSecretKey=`{ "Fn::GetAtt" : ["WebServerKeys", "SecretAccessKey"] }`
    mode : "000400"
    owner : root
    group : root

  "/etc/cfn/get-cache-nodes" :
    content : |
      # Define environment variables for command line tools
```

```
      export AWS_ELASTICACHE_HOME="/home/ec2-user/elasticache/$(ls /home/ec2-user/
elasticache/)"
      export AWS_CLOUDFORMATION_HOME=/opt/aws/apitools/cfn
      export PATH=$AWS_CLOUDFORMATION_HOME/bin:$AWS_ELASTICACHE_HOME/bin:$PATH
      export AWS_CREDENTIAL_FILE=/etc/cfn/cfn-credentials
      export JAVA_HOME=/usr/lib/jvm/jre

      # Grab the Cache node names and configure the PHP page
      aws cloudformation list-stack-resources --stack `{ "Ref" :
 "AWS::StackName" }` --region `{ "Ref" : "AWS::Region" }` --output text | grep
 MyElastiCache | awk '{print $4}' | xargs -I {} aws elasticache describe-cache-
clusters --cache-cluster-id {} --region `{ "Ref" : "AWS::Region" }` --show-
cache-node-info --output text | grep '^ENDPOINT' | awk '{print $2 ":" $3}' >
 `{ "Fn::GetOptionSetting" : { "OptionName" : "NodeListPath", "DefaultValue" : "/
var/www/html/nodelist" } }`
    mode : "000500"
    owner : root
    group : root

  "/etc/cfn/hooks.d/cfn-cache-change.conf" :
    "content": |
      [cfn-cache-size-change]
      triggers=post.update
      path=Resources.AWSEBAutoScalingGroup.Metadata.ElastiCacheConfig
      action=/etc/cfn/get-cache-nodes
      runas=root

sources :
  "/home/ec2-user/elasticache" : "https://elasticache-downloads.s3.amazonaws.com/
AmazonElastiCacheCli-latest.zip"

commands:
  make-elasticache-executable:
    command: chmod -R ugo+x /home/ec2-user/elasticache/*/bin/*

packages :
  "yum" :
    "aws-apitools-cfn"  : []

container_commands:
  initial_cache_nodes:
    command: /etc/cfn/get-cache-nodes
```

3. Sul computer locale, create un file di configurazione `nodejs-example-express-elasticache/.ebextensions/elasticache_settings.config` con il seguente frammento di codice da configurare. ElastiCache

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     CacheNodeType: cache.t2.micro
     NumCacheNodes: 1
     Engine: memcached
     NodeListPath: /var/nodelist
```

4. Sul computer locale, sostituire `nodejs-example-express-elasticache/express-app.js` con lo snippet seguente. Questo file legge l'elenco dei nodi dal disco (`/var/nodelist`) e configura express per utilizzare `memcached` come uno store di sessione se sono presenti nodi. Il file avrà un aspetto simile al seguente.

```
/**
 * Module dependencies.
 */

var express = require('express'),
    session = require('express-session'),
    bodyParser = require('body-parser'),
    methodOverride = require('method-override'),
    cookieParser = require('cookie-parser'),
    fs = require('fs'),
    filename = '/var/nodelist',
    app = module.exports = express();

var MemcachedStore = require('connect-memcached')(session);

function setup(cacheNodes) {
  app.use(bodyParser.raw());
  app.use(methodOverride());
  if (cacheNodes) {
      app.use(cookieParser());

      console.log('Using memcached store nodes:');
      console.log(cacheNodes);

      app.use(session({
          secret: 'your secret here',
```

```
          resave: false,
          saveUninitialized: false,
          store: new MemcachedStore({'hosts': cacheNodes})
      }));
  } else {
    console.log('Not using memcached store.');
    app.use(cookieParser('your secret here'));
    app.use(session());
  }

  app.get('/', function(req, resp){
  if (req.session.views) {
      req.session.views++
      resp.setHeader('Content-Type', 'text/html')
      resp.write('Views: ' + req.session.views)
      resp.end()
   } else {
      req.session.views = 1
      resp.end('Refresh the page!')
    }
  });

  if (!module.parent) {
      console.log('Running express without cluster.');
      app.listen(process.env.PORT || 5000);
  }
}

// Load elasticache configuration.
fs.readFile(filename, 'UTF8', function(err, data) {
    if (err) throw err;

    var cacheNodes = [];
    if (data) {
        var lines = data.split('\n');
        for (var i = 0 ; i < lines.length ; i++) {
            if (lines[i].length > 0) {
                cacheNodes.push(lines[i]);
            }
        }
    }
    setup(cacheNodes);
});
```

5. Sul computer locale, aggiorna `package.json` con i seguenti contenuti:

```
"dependencies": {
  "cookie-parser": "~1.4.4",
  "debug": "~2.6.9",
  "express": "~4.16.1",
  "http-errors": "~1.6.3",
  "jade": "~1.11.0",
  "morgan": "~1.9.1",
  "connect-memcached": "*",
  "express-session": "*",
  "body-parser": "*",
  "method-override": "*"
}
```

6. Esegui npm install.

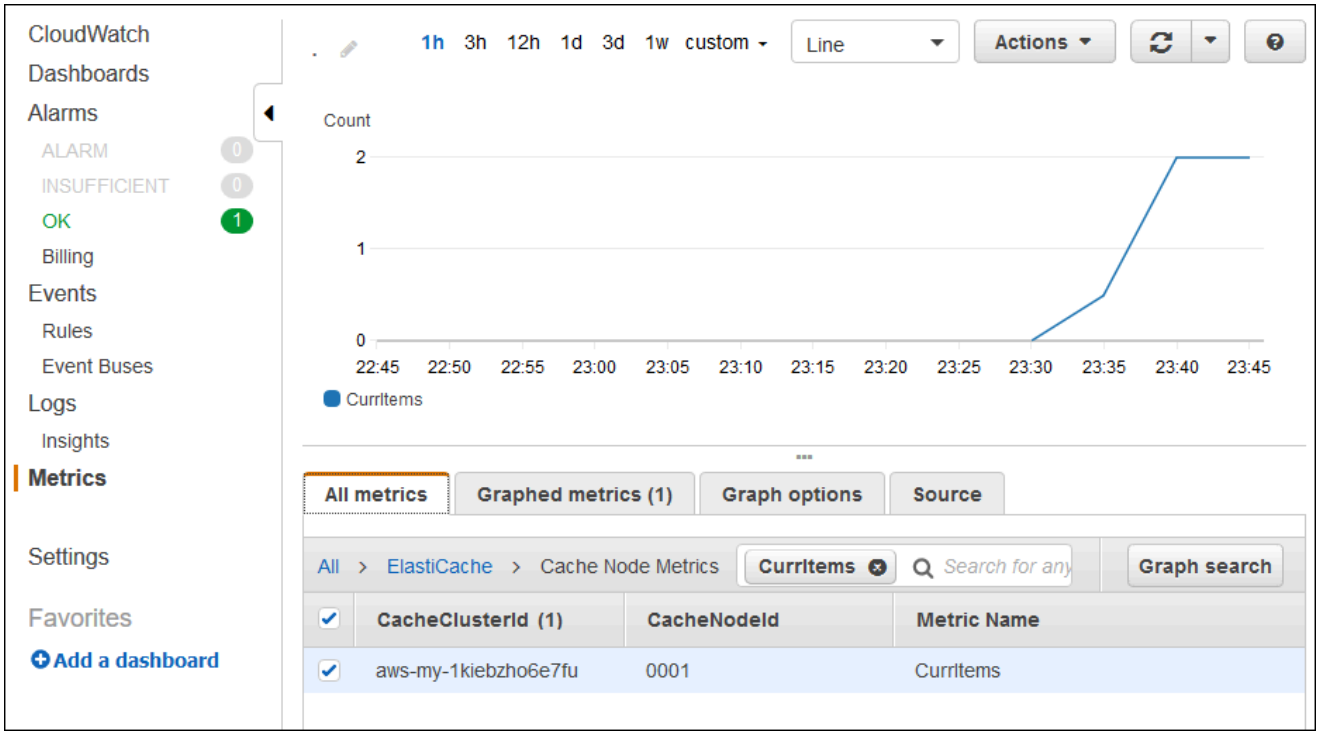
```
~/nodejs-example-express-elasticache$ npm install
```

7. Distribuisci l'applicazione aggiornata.

```
~/nodejs-example-express-elasticache$ eb deploy
```

8. Il tuo ambiente sarà aggiornato dopo qualche minuto. Quando l'ambiente è verde e pronto, verificare che il codice abbia funzionato.

   a. Controlla la CloudWatch console Amazon per visualizzare le tue ElastiCache metriche. Per visualizzare le tue ElastiCache metriche, seleziona Metriche nel riquadro a sinistra, quindi cerca. CurrItems Seleziona ElastiCache > Metriche del nodo di cache, quindi seleziona il nodo di cache per visualizzare il numero di elementi nella cache.



> **Note**
>
> Verifica di fare riferimento alla stessa regione in cui hai distribuito la tua applicazione.

Se copi e incolli l'URL dell'applicazione in un altro browser Web e aggiorni la pagina, dovresti vedere il CurrItem conteggio aumentare dopo 5 minuti.

b. Acquisire uno snapshot dei log. Per ulteriori informazioni sul recupero di oggetti, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

c. Selezionare il file `/var/log/nodejs/nodejs.log` nel bundle dei log. Dovrebbe essere visualizzato un output simile al seguente:

```
Using memcached store nodes:
[ 'aws-my-1oys9co8zt1uo.1iwtrn.0001.use1.cache.amazonaws.com:11211' ]
```

Elimina

Se non desideri più eseguire l'applicazione, puoi chiudere il tuo ambiente ed eliminarla.

Utilizza il comando `eb terminate` per chiudere il tuo ambiente e il comando `eb delete` per eliminare l'applicazione.

Per terminare l'ambiente

Nella directory in cui hai creato il tuo repository locale, esegui `eb terminate`.

```
$ eb terminate
```

Questo processo può richiedere alcuni minuti. Elastic Beanstalk visualizza un messaggio non appena l'ambiente viene chiuso correttamente.

Distribuzione di un'applicazione Node.js con DynamoDB in Elastic Beanstalk

Questo tutorial e la relativa applicazione di esempio nodejs-example-dynamo.zip illustrano il processo di distribuzione di un'applicazione Node.js che utilizza l' AWS SDK for JavaScript in Node.js per interagire con il servizio Amazon DynamoDB. Creerai una tabella DynamoDB che si trova in un database disaccoppiato, o esterno, dall'ambiente. AWS Elastic Beanstalk Configurerai anche l'applicazione per utilizzare un database disaccoppiato. In un ambiente di produzione, è consigliabile utilizzare un database disaccoppiato dall'ambiente Elastic Beanstalk in modo che sia indipendente dal ciclo di vita dell'ambiente. Questa procedura consente anche di eseguire implementazioni blu/verde.

L'applicazione di esempio illustra quanto riportato di seguito:

- Una tabella DynamoDB che archivia i dati di testo forniti dall'utente.
- I file di configurazione per creare la tabella.
- Un argomento su Amazon Simple Notification Service.
- Utilizzo di un file package.json per installare i pacchetti durante l'implementazione.

Sections

- Prerequisiti
- Creazione di un ambiente Elastic Beanstalk
- Aggiungi le autorizzazioni alle istanze dell'ambiente

- Implementazione dell'applicazione di esempio
- Creazione di una tabella DynamoDB
- Aggiornamento dei file di configurazione dell'applicazione
- Configurazione dell'ambiente per l'elevata disponibilità
- Pulizia
- Fasi successive

Prerequisiti

Di seguito sono elencati i requisiti per questo tutorial:

- I runtime Node.js
- Il software predefinito per la gestione dei pacchetti Node.js, npm
- Il generatore di righe di comando Express
- interfaccia a riga di comando Elastic Beanstalk (CLI EB)

Per ulteriori informazioni sull'installazione dei primi tre componenti riportati e la configurazione dell'ambiente di sviluppo locale, consulta Configurazione dell'ambiente di sviluppo Node.js per Elastic Beanstalk. Per questo tutorial, non è necessario installare l' AWS SDK per Node.js, anch'esso menzionato nell'argomento a cui si fa riferimento.

Per maggiori dettagli sull'installazione e la configurazione della CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Creazione di un ambiente Elastic Beanstalk

La tua directory delle applicazioni

Questo tutorial utilizza una directory chiamata `nodejs-example-dynamo` per il pacchetto sorgente dell'applicazione. Crea la directory `nodejs-example-dynamo` per questo tutorial.

```
~$ mkdir nodejs-example-dynamo
```

> **Note**
>
> Ogni tutorial in questo capitolo utilizza la propria directory per il pacchetto sorgente dell'applicazione. Il nome della directory corrisponde al nome dell'applicazione di esempio utilizzata dal tutorial.

Cambia la tua directory di lavoro corrente in `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

Configura ora un ambiente Elastic Beanstalk in cui viene eseguita la piattaforma Node.js e l'applicazione di esempio. Utilizzeremo l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB).

Configurazione di un repository della CLI EB per l'applicazione e creaione di un ambiente Elastic Beanstalk in cui viene eseguita la piattaforma Node.js

1. Creazione di un repository con il comando eb init.

   ```
   ~/nodejs-example-dynamo$ eb init --platform node.js --region <region>
   ```

   Questo comando crea un file di configurazione in una cartella denominata `.elasticbeanstalk`, che specifica le impostazioni per la creazione di ambienti per l'applicazione e crea un'applicazione Elastic Beanstalk che prende nome dalla cartella corrente.

2. Crea un ambiente che esegue un'applicazione di esempio con il comando eb create.

   ```
   ~/nodejs-example-dynamo$ eb create --sample nodejs-example-dynamo
   ```

   Questo comando crea un ambiente con bilanciamento del carico con le impostazioni predefinite per la piattaforma Node.js e le risorse seguenti:

   - Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

     Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX

come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.

- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *`subdomain region`*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi

di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

3. Quando la creazione dell'ambiente è completa, utilizza il comando eb open per aprire l'URL dell'ambiente nel browser predefinito.

```
~/nodejs-example-dynamo$ eb open
```

È stato ora creato un ambiente Elastic Beanstalk Node.js con un'applicazione di esempio. A questo punto puoi aggiornarlo con la tua applicazione. Successivamente, aggiorniamo l'applicazione di esempio per utilizzare il framework Express.

Aggiungi le autorizzazioni alle istanze dell'ambiente

L'applicazione viene eseguita su una o più istanze EC2 con un sistema di bilanciamento del carico e serve le richieste HTTP da Internet. Quando riceve una richiesta che richiede l'utilizzo di AWS servizi, l'applicazione utilizza le autorizzazioni dell'istanza su cui viene eseguita per accedere a tali servizi.

L'applicazione di esempio utilizza le autorizzazioni dell'istanza per scrivere dati su una tabella DynamoDB e per inviare notifiche a un argomento di Amazon SNS con l'SDK for in Node.js. JavaScript Aggiungi le seguenti policy gestite al profilo dell'istanza predefinito per concedere alle istanze EC2 presenti nell'ambiente l'autorizzazione di accesso a DynamoDB e Amazon SNS:

- AmazonDynamoDBFullAccesso
- SNSFullAccesso Amazon

Per aggiungere policy al profilo dell'istanza di default

1. Aprire la pagina Roles (Ruoli) nella console IAM.
2. Scegli aws-elasticbeanstalk-ec2 ruoli.
3. Nella scheda Autorizzazioni, scegli Collega policy.
4. Seleziona la policy gestita per i servizi aggiuntivi utilizzati dall'applicazione. Per questo tutorial, seleziona `AmazonSNSFullAccess` e `AmazonDynamoDBFullAccess`.
5. Scegli Collega policy.

Per ulteriori informazioni sulla gestione dei profili delle istanze, consulta Gestione dei profili dell'istanza Elastic Beanstalk.

Implementazione dell'applicazione di esempio

Ora il tuo ambiente è pronto per la distribuzione e l'esecuzione dell'applicazione di esempio per questo tutorial: nodejs-example-dynamo .zip.

Implementazione ed esecuzione dell'applicazione di esempio del tutorial

1. Modifica la directory di lavoro corrente nella directory dell'applicazione `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

2. Scaricate ed estraete il contenuto del pacchetto sorgente dell'applicazione di esempio nodejs-example-dynamo.zip nella directory dell'applicazione. `nodejs-example-dynamo`
3. Implementa l'applicazione di esempio nell'ambiente Elastic Beanstalk con il comando eb deploy.

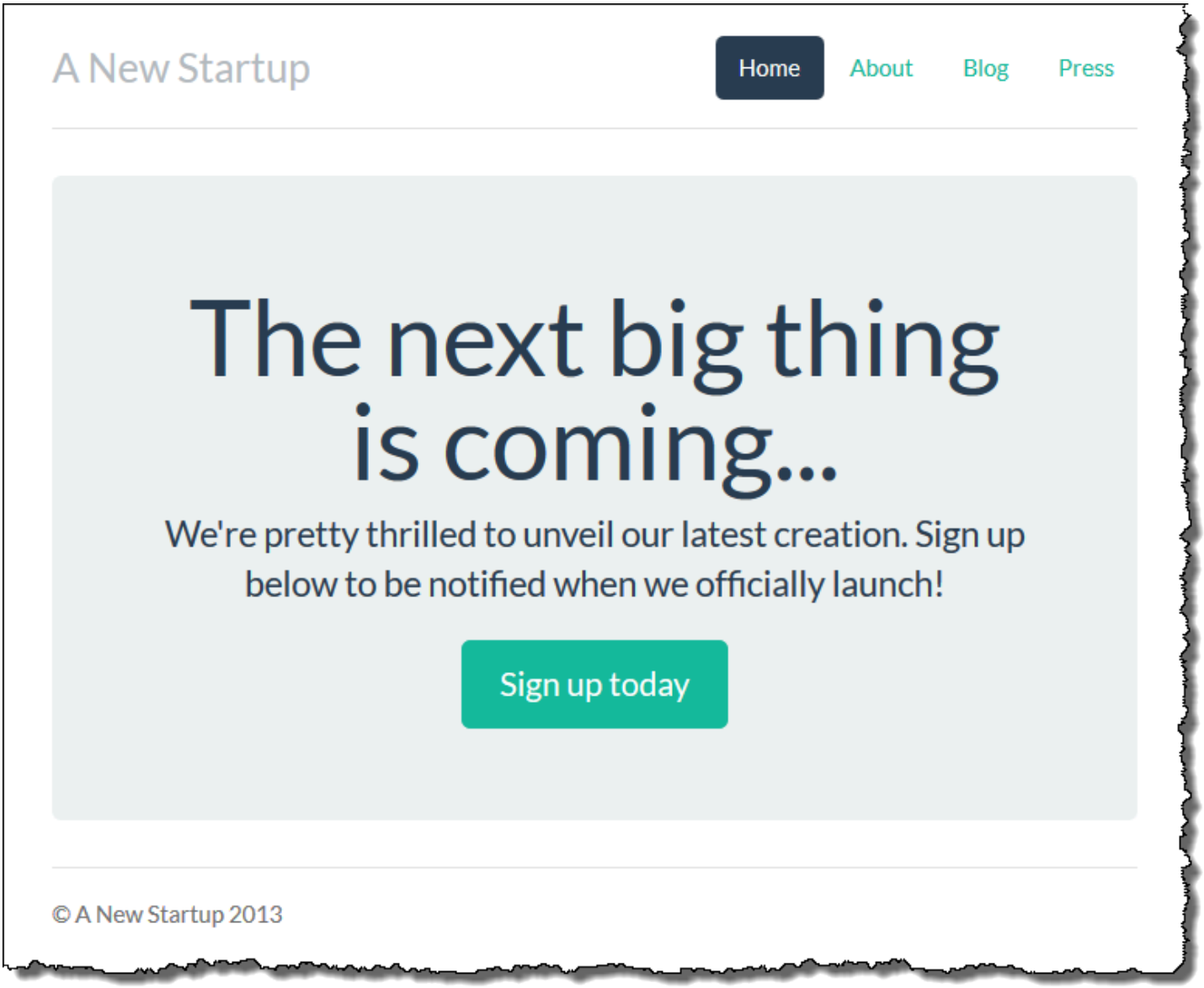
```
~/nodejs-example-dynamo$ eb deploy
```

> **Note**
>
> Per impostazione predefinita, il `eb deploy` comando crea un file ZIP della cartella del progetto. È possibile anche configurare la CLI EB per distribuire un artefatto del processo di build anziché creare un file ZIP della cartella di progetto. Per ulteriori informazioni, consulta Distribuzione di artifact invece della cartella del progetto.

4. Quando la creazione dell'ambiente è completa, utilizza il comando eb open per aprire l'URL dell'ambiente nel browser predefinito.

```
~/nodejs-example-dynamo$ eb open
```

Il sito raccoglie le informazioni di contatto degli utenti e utilizza una tabella DynamoDB per archiviare i dati. Per aggiungere una voce, seleziona Sign up today (Registrati oggi stesso), inserisci un nome e un indirizzo di posta elettronica, quindi scegli Sign up! (Registrati) L'applicazione Web scrive i contenuti del modulo sulla tabella e attiva una notifica Amazon SNS via e-mail.



In questo momento, l'argomento Amazon SNS è configurato con un'e-mail segnaposto per le notifiche. Aggiornerai la configurazione a breve, ma nel frattempo è possibile verificare la tabella DynamoDB e l'argomento Amazon SNS nella console di gestione Console di gestione AWS.

Per visualizzare la tabella

1. Apri la pagina delle tabelle nella console DynamoDB.
2. Trova la tabella creata dall'applicazione. Il nome inizia con awseb e contiene. StartupSignupsTable
3. Seleziona la tabella, quindi Items (Voci) e Start search (Avvia ricerca) per visualizzare tutti gli elementi della tabella.

La tabella contiene una voce per ogni indirizzo e-mail inviato sul sito dell'iscrizione. Oltre a scrivere sulla tabella, l'applicazione invia un messaggio a un argomento Amazon SNS che dispone di due abbonamenti, uno per le notifiche e-mail per l'utente e un altro per una coda Amazon Simple Queue Service da cui un'applicazione worker è in grado di leggere per elaborare le richieste e inviare e-mail ai clienti interessati.

Per visualizzare l'argomento

1. Apri la pagina degli argomenti nella console Amazon SNS.
2. Trova l'argomento creato dall'applicazione. Il nome inizia con awseb e contiene. NewSignupTopic
3. Scegli l'argomento per visualizzare le sottoscrizioni.

L'applicazione (`app.js`) definisce due itinerari. Il percorso principale (`/`) restituisce una pagina Web renderizzata da un modello Embedded JavaScript (EJS) con un modulo che l'utente compila per registrare il proprio nome e indirizzo e-mail. Con l'inserimento del modulo viene inviata una richiesta POST con i dati al percorso `/signup` che scrive una voce nella tabella DynamoDB e pubblica un messaggio nell'argomento Amazon SNS per notificare il proprietario dell'iscrizione.

L'applicazione di esempio include i file di configurazione che creano la tabella DynamoDB, l'argomento Amazon SNS e la coda Amazon SQS utilizzati dall'applicazione. Ciò consente di creare un nuovo ambiente e di verificare la funzionalità immediatamente, ma ha lo svantaggio di vincolare la tabella DynamoDB all'ambiente. Per un ambiente di produzione, è necessario creare la tabella DynamoDB al di fuori dell'ambiente per evitare di perderla quando termini l'ambiente o ne aggiorni la configurazione.

Creazione di una tabella DynamoDB

Per utilizzare una tabella DynamoDB esterna con un'applicazione in esecuzione in Elastic Beanstalk, crea innanzitutto una tabella in DynamoDB. Quando crei una tabella esternamente a Elastic Beanstalk, essa sarà indipendente da Elastic Beanstalk e dagli ambienti Elastic Beanstalk e non verrà arrestata da Elastic Beanstalk.

Creare una tabella con le impostazioni seguenti:

- Table name (Nome tabella) – **`nodejs-tutorial`**
- Primary key (Chiave primaria) – **`email`**
- Tipo chiave principale: String (Stringa)

Per creare una tabella DynamoDB

1. Apri la pagina delle tabelle nella console di gestione DynamoDB.
2. Scegliere Create table (Crea tabella).
3. Digita un Table name (Nome tabella) e la Primary key (Chiave primaria).
4. Scegli il tipo di chiave primaria.
5. Scegli Create (Crea).

Aggiornamento dei file di configurazione dell'applicazione

Aggiorna i file di configurazione nella sorgente dell'applicazione per utilizzare la tabella nodejs-tutorial invece di crearne una nuova.

Aggiornamento dell'applicazione di esempio per l'uso negli ambienti di produzione

1. Modifica la directory di lavoro corrente nella directory dell'applicazione `nodejs-example-dynamo`.

   ```
   ~$ cd nodejs-example-dynamo
   ```

2. Apri `.ebextensions/options.config` e modifica i valori delle impostazioni seguenti:

   - NewSignupEmail— Il tuo indirizzo email.
   - STARTUP_SIGNUP_TABLE: nodejs-tutorial

   Example.ebextensions/options.config

   ```
   option_settings:
     aws:elasticbeanstalk:customoption:
       NewSignupEmail: you@example.com
     aws:elasticbeanstalk:application:environment:
       THEME: "flatly"
       AWS_REGION: '`{"Ref" : "AWS::Region"}`'
       STARTUP_SIGNUP_TABLE: nodejs-tutorial
       NEW_SIGNUP_TOPIC: '`{"Ref" : "NewSignupTopic"}`'
     aws:elasticbeanstalk:container:nodejs:
       ProxyServer: nginx
     aws:elasticbeanstalk:container:nodejs:staticfiles:
   ```

```
    /static: /static
  aws:autoscaling:asg:
    Cooldown: "120"
  aws:autoscaling:trigger:
    Unit: "Percent"
    Period: "1"
    BreachDuration: "2"
    UpperThreshold: "75"
    LowerThreshold: "30"
    MeasureName: "CPUUtilization"
```

Ciò vale per le seguenti configurazioni per l'applicazione:

- L'indirizzo e-mail che l'argomento Amazon SNS impiega per le notifiche è impostato sul tuo indirizzo o su quello che inserisci nel file `options.config`.
- La tabella nodejs-tutorial verrà utilizzata al posto di quella creata da `.ebextensions/create-dynamodb-table.config`.

3. Remove `.ebextensions/create-dynamodb-table.config`.

```
~/nodejs-tutorial$ rm .ebextensions/create-dynamodb-table.config
```

La prossima volta che distribuisci l'applicazione, la tabella creata da questo file di configurazione verrà eliminata.

4. Implementa l'applicazione aggiornata nell'ambiente Elastic Beanstalk con il comando eb deploy.

```
~/nodejs-example-dynamo$ eb deploy
```

5. Quando la creazione dell'ambiente è completa, utilizza il comando eb open per aprire l'URL dell'ambiente nel browser predefinito.

```
~/nodejs-example-dynamo$ eb open
```

Durante la distribuzione, Elastic Beanstalk aggiorna la configurazione dell'argomento Amazon SNS ed elimina la tabella DynamoDB creata quando hai distribuito la prima versione dell'applicazione.

Ora, quando termini l'ambiente, la tabella nodejs-tutorial non sarà eliminata. In questo modo è possibile eseguire le implementazioni blu/verdi, modificare i file di configurazione oppure arrestare il sito Web senza il rischio di perdite di dati.

Apri il sito in un browser e verifica che il modulo funzioni come previsto. Crea alcune voci, quindi controlla la console DynamoDB per verificare la tabella.

Per visualizzare la tabella

1. Apri la pagina delle tabelle nella console DynamoDB.
2. Trova la tabella nodejs-tutorial.
3. Seleziona la tabella, quindi Items (Voci) e Start search (Avvia ricerca) per visualizzare tutti gli elementi della tabella.

Puoi anche notare che Elastic Beanstalk ha eliminato la tabella creata in precedenza.

Configurazione dell'ambiente per l'elevata disponibilità

Infine, configura il gruppo Auto Scaling dell'ambiente con un conteggio delle istanze minime più elevato. Esegui almeno due istanze in qualsiasi momento per evitare che i server Web nell'ambiente rappresentino una singola possibilità di errore e per consentire di distribuire le modifiche senza interrompere il funzionamento del servizio.

Configurazione del gruppo Auto Scaling dell'ambiente per elevata disponibilità

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Auto Scaling Group (Gruppo Auto Scaling), impostare Min instances (Istanze min) su **2**.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Pulizia

Dopo aver finito di utilizzare il codice demo, puoi chiudere l'ambiente. Elastic Beanstalk elimina AWS tutte le risorse correlate, come istanze Amazon EC2, istanze di database, sistemi di bilanciamentodel carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Puoi anche eliminare le tabelle DynamoDB esterne che hai creato.

Per eliminare una tabella DynamoDB

1. Apri la pagina delle tabelle nella console DynamoDB.
2. Seleziona una tabella.
3. Seleziona Actions (Operazioni), quindi Delete table (Elimina tabella).
4. Scegli Elimina.

Fasi successive

L'applicazione di esempio utilizza i file di configurazione per configurare le impostazioni del software e creare AWS risorse come parte dell'ambiente. Per ulteriori informazioni sui file di configurazione e sul loro uso, consulta Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

L'applicazione di esempio per questo tutorial usa il framework Web Express per Node.js. Per ulteriori informazioni su Express, consulta la documentazione ufficiale alla pagina expressjs.com.

Infine, se intendi utilizzare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Aggiungere un'istanza Amazon RDS DB all'ambiente Node.js Elastic Beanstalk

Questo argomento fornisce istruzioni per creare un Amazon RDS utilizzando la console Elastic Beanstalk. Puoi utilizzare un'istanza database Amazon Relational Database Service (Amazon RDS) per archiviare i dati raccolti e modificati dall'applicazione. Il database può essere accoppiato all'ambiente e gestito da Elastic Beanstalk oppure può essere creato e gestito esternamente da un altro servizio. In queste istruzioni il database è accoppiato all'ambiente e gestito da Elastic Beanstalk. Per ulteriori informazioni sull'integrazione di un Amazon RDS con Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Sections

- Aggiunta di un'istanza database all'ambiente
- Download di un driver
- Connessione a un database

Aggiunta di un'istanza database all'ambiente

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'aggiunta di un'istanza database richiede circa 10 minuti. Quando l'aggiornamento dell'ambiente è completo, il nome host dell'istanza database e altre informazioni di connessione sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente:

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Per ulteriori informazioni sulla configurazione di un'istanza di database accoppiata a un ambiente Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Download di un driver

Aggiungi il driver di database al file `package.json` del progetto in `dependencies`.

Example **`package.json`**: Express con MySQL

```
{
  "name": "my-app",
  "version": "0.0.1",
```

```
  "private": true,
  "dependencies": {
    "ejs": "latest",
    "aws-sdk": "latest",
    "express": "latest",
    "body-parser": "latest",
    "mysql": "latest"
  },
  "scripts": {
    "start": "node app.js"
  }
}
```

Pacchetti di driver comuni per Node.js

- MySQL: mysql
- PostgreSQL: node-postgres
- SQL Server: node-mssql
- Oracle: node-oracledb

Connessione a un database

Elastic Beanstalk fornisce informazioni di connessione per le istanze database collegate nelle proprietà dell'ambiente. Utilizza `process.env.VARIABLE` per leggere le proprietà e configurare una connessione di database.

Example app.js: connessione al database MySQL

```
var mysql = require('mysql');

var connection = mysql.createConnection({
  host     : process.env.RDS_HOSTNAME,
  user     : process.env.RDS_USERNAME,
  password : process.env.RDS_PASSWORD,
  port     : process.env.RDS_PORT
});

connection.connect(function(err) {
  if (err) {
    console.error('Database connection failed: ' + err.stack);
    return;
```

```
  }

  console.log('Connected to database.');
});

connection.end();
```

Per ulteriori informazioni sulla creazione di una stringa di connessione utilizzando node-mysql, vedi npmjs. org/package/mysql.

Strumenti e risorse Node.js

Di seguito sono elencate alcune risorse che ti consentono di ottenere ulteriori informazioni relative allo sviluppo di applicazioni Node.js:

Risorsa	Description
GitHub	Installa l' AWS SDK per Node.js utilizzando GitHub.
AWS SDK per Node.js (anteprima per sviluppatori)	Punto di riferimento per il codice di esempio, la documentazione, gli strumenti e risorse aggiuntive.

Distribuzione di applicazioni PHP con Elastic Beanstalk

Puoi distribuire un'applicazione PHP in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o la console Elastic Beanstalk. Per scoprire come, inizia con il tutorial, quindi consulta gli esempi avanzati. QuickStart

Risorse per la distribuzione di applicazioni PHP su Elastic Beanstalk

- QuickStart per PHP— Step-by-step istruzioni per distribuire un'applicazione PHP Hello World utilizzando l'EB CLI.
- Utilizzo della piattaforma PHP di Elastic Beanstalk— Come utilizzare le funzionalità della piattaforma specifiche per PHP.
- Esempi avanzati di — Tutorial pratici per framework e applicazioni PHP comuni, oltre a come aggiungere un database Amazon RDS al tuo ambiente.

Per maggiori informazioni sullo sviluppo con PHP in, consulta le seguenti risorse: AWS

- GitHub— Installa l' AWS SDK for GitHub PHP utilizzando.
- PHP Developer Center: strumenti, documenti e codice di esempio su cui sviluppare applicazioni PHP. AWS
- AWS SDK FAQs for PHP: ottieni le risposte alle domande più frequenti.

QuickStart: distribuisci un'applicazione PHP su Elastic Beanstalk

Nel seguente tutorial, imparerai come creare e distribuire un'applicazione PHP di esempio in un AWS Elastic Beanstalk ambiente utilizzando l'EB CLI.

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account

- Prerequisiti
- Fase 1: Creare un'applicazione PHP
- Passaggio 2: Esegui l'applicazione localmente
- Passaggio 3: inizializza e distribuisci l'applicazione PHP
- Passaggio 4: sfoglia la tua applicazione cloud
- Passaggio 5: Aggiorna e ridistribuisci l'applicazione
- Eliminazione
- Fasi successive

Vostro AWS account

Se non sei già un AWS cliente, devi creare un AWS account per utilizzare Elastic Beanstalk.

Crea un AWS account

Iscriviti per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

- Interfaccia a riga di comando Elastic Beanstalk - Per l'installazione, vedere. Installa EB CLI con lo script di configurazione (consigliato)
- PHP - Installa PHP sul tuo computer locale seguendo le istruzioni di installazione e configurazione sul sito web di PHP.

Fase 1: Creare un'applicazione PHP

Per questo avvio rapido, creerai un'applicazione PHP Hello World.

Crea una directory di progetto.

```
~$ mkdir eb-php
```

```
~$ cd eb-php
```

Quindi, crea un `index.php` file nella directory del progetto e aggiungi il codice seguente.

Example **`index.php`**

```
<?php
  echo "Hello from a PHP application running in Elastic Beanstalk!";
?>
```

Passaggio 2: Esegui l'applicazione localmente

Usa il seguente comando per eseguire l'applicazione localmente.

```
~$ php -S localhost:5000
```

Apri un browser per http://localhost:5000.

Dovresti vedere il tuo messaggio di saluto nel browser e registrare i messaggi nel tuo terminale.

Arresta il server locale entrando`Control+c`, in modo da poter distribuire Elastic Beanstalk.

Passaggio 3: inizializza e distribuisci l'applicazione PHP

Successivamente, distribuirai l'applicazione in un ambiente utilizzando la console Elastic Beanstalk o l'EB CLI. Per questo tutorial, utilizzerai l'EB CLI con l'opzione interattiva per inizializzare un ambiente.

Per inizializzare l'ambiente e creare un ambiente

1. Eseguire il seguente comando init.

   ```
   ~$ eb init -i
   ```

   Il comando init crea un'applicazione in modo interattivo. Il nome dell'applicazione verrà impostato per impostazione predefinita sulla cartella locale che è. `eb-php`

   Per tutte le richieste, ad eccezione dell'accesso SSH, accettate le impostazioni predefinite per creare un ambiente con l'ultima versione della piattaforma PHP. Per la risoluzione dei problemi, puoi configurare l'accesso SSH rieseguendo il `eb init -i` comando in un secondo momento oppure connetterti utilizzando Amazon EC2 Instance Connect o Session Manager.

2. Crea un ambiente e distribuisci la tua applicazione

   Esegui il comando seguente per creare un ambiente denominato`blue-env`.

   ```
   ~$ eb create blue-env
   ```

   Quando esegui il eb create comando per la prima volta, Elastic Beanstalk crea automaticamente un file zip dell'applicazione, chiamato pacchetto sorgente. Successivamente, Elastic Beanstalk crea un ambiente con una o più istanze Amazon EC2, quindi distribuisce l'applicazione nell'ambiente.

   La distribuzione dell'applicazione su Elastic Beanstalk potrebbe richiedere fino a cinque minuti.

Passaggio 4: sfoglia la tua applicazione cloud

Una volta completato il processo di creazione dell'ambiente, l'applicazione dovrebbe essere in esecuzione e in ascolto delle richieste sulla porta 5000. Connect all'applicazione con il seguente comando:

```
~$ eb open
```

Il `eb open` comando apre una scheda del browser in un sottodominio personalizzato creato per l'applicazione.

Passaggio 5: Aggiorna e ridistribuisci l'applicazione

Dopo aver creato un'applicazione e averla distribuita in un ambiente, è possibile distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è più rapida perché non richiede il provisioning o il riavvio di istanze Amazon EC2.

Aggiorna il codice PHP per includere il valore REQUEST_TIME dall'ambiente server:

```
<?php
  echo "Hello from a PHP application running in Elastic Beanstalk!";

  $timestamp = $_SERVER['REQUEST_TIME'];
  echo '<br/>Request time: ' . date('Y/m/d H:i:s', $timestamp);
```

```
?>
```

Ridistribuisci il codice PHP su Elastic Beanstalk con il seguente comando:

```
~$ eb deploy
```

Quando esegui eb deploy, la CLI EB integra i contenuti della tua directory di progetto in un bundle e li distribuisce al tuo ambiente.

Al termine della distribuzione, aggiorna la pagina o riconnettiti all'applicazione con. `eb open` Dovresti vedere i tuoi aggiornamenti. In caso contrario, risolvi i problemi eseguendo nuovamente il server locale per verificare le modifiche.

> **Complimenti!**
>
> Hai creato, distribuito e aggiornato un'applicazione PHP con Elastic Beanstalk!

Eliminazione

Dopo aver finito di utilizzare il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina AWS tutte le risorse correlate, come istanze Amazon EC2, istanze di database, sistemi di bilanciamentodel carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

In alternativa, è possibile terminare l'ambiente con l'EB CLI con il seguente comando:

```
~$ eb terminate
```

Fasi successive

Puoi esplorare il tuo ambiente applicativo utilizzando la console Elastic Beanstalk. Per maggiori informazioni, consulta Esplora il tuo ambiente.

Per esempi avanzati di utilizzo di PHP, consultaEsempi avanzati di PHP in Elastic Beanstalk.

Utilizzo della piattaforma PHP di Elastic Beanstalk

AWS Elastic Beanstalk fornisce e supporta vari rami della piattaforma per diverse versioni di PHP. Le piattaforme supportano applicazioni web PHP che funzionano in modo autonomo o con Composer. Consulta PHP nel documento AWS Elastic Beanstalk Piattaforme per un elenco completo dei rami della piattaforma supportati.

Elastic Beanstalk offre opzioni di configurazione che puoi utilizzare per personalizzare il software in esecuzione sulle istanze Amazon EC2 nel tuo ambiente Elastic Beanstalk. È possibile configurare le variabili di ambiente necessarie per l'applicazione, abilitare la rotazione dei log su Amazon S3, mappare le cartelle nell'origine dell'applicazione che contengono i file statici sui percorsi serviti dal server proxy e configurare le impostazioni di inizializzazione PHP comuni.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

Se utilizzi Composer, puoi includere un file `composer.json` nel bundle di origine per installare pacchetti durante la distribuzione.

Per la configurazione e le impostazioni avanzate di PHP che non vengono fornite come opzioni di configurazione, è possibile utilizzare i file di configurazione per fornire un file `INI` che può estendere e ignorare le impostazioni di default applicate da Elastic Beanstalk o installare altre estensioni.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori

informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Argomenti relativi alla piattaforma PHP

- Installazione dell' AWS SDK for PHP
- Considerazioni per PHP 8.1 su Amazon Linux 2
- Configurazione dell'ambiente PHP
- Namespace per la configurazione
- Installazione delle dipendenze dell'applicazione PHP Elastic Beanstalk
- Aggiornamento di Composer su Elastic Beanstalk
- Estendere php.ini nella configurazione di Elastic Beanstalk

Installazione dell' AWS SDK for PHP

Se devi gestire AWS le risorse dall'interno dell'applicazione, installa il. AWS SDK per PHP Ad esempio, con l'SDK for PHP, puoi utilizzare Amazon DynamoDB (DynamoDB) per archiviare le informazioni sulla sessione e sugli utenti senza creare un database relazionale.

Per installare l'SDK for PHP con Composer

```
$ composer require aws/aws-sdk-php
```

Per ulteriori informazioni, consulta la home page. AWS SDK for PHP Per istruzioni, consulta Installare il AWS SDK for PHP.

Considerazioni per PHP 8.1 su Amazon Linux 2

Leggi questa sezione se stai usando la ramificazione della piattaforma PHP 8.1 su Amazon Linux 2.

Considerazioni per PHP 8.1 su Amazon Linux 2

> **Note**
>
> Le informazioni contenute in questo argomento si applicano solo alle ramificazioni della piattaforma PHP 8.1 basati su Amazon Linux 2. Non si applica ai rami della piattaforma PHP

basati su AL2023. Inoltre non si applica alla ramificazione della piattaforma PHP 8.0 Amazon Linux 2.

Elastic Beanstalk memorizza i pacchetti RPM relativi a PHP 8.1 per la ramificazione della piattaforma PHP 8.1 su Amazon Linux 2 sulle istanze EC2 in una directory locale, anziché nel repository Amazon Linux. È possibile utilizzare rpm -i per installare pacchetti. A partire dalla versione 3.5.0 della piattaforma PHP 8.1, Elastic Beanstalk archivia i pacchetti RPM relativi a PHP 8.1 nella seguente directory EC2 locale.

`/opt/elasticbeanstalk/RPMS`

L'esempio seguente installa il pacchetto php-debuginfo.

```
$rpm -i /opt/elasticbeanstalk/RPMS/php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
```

La versione nel nome del pacchetto varierà in base alla versione effettiva elencata nella directory EC2 locale `/opt/elasticbeanstalk/RPMS`. Usa la stessa sintassi per installare altri pacchetti RPM PHP 8.1.

Espandi la sezione seguente per visualizzare un elenco dei pacchetti RPM che forniamo.

Pacchetti RPM

L'elenco seguente mostra i pacchetti RMP forniti dalla piattaforma PHP 8.1 Elastic Beanstalk su Amazon Linux 2. Questi si trovano nella directory locale `/opt/elasticbeanstalk/RPMS`.

I numeri di versione 8.1.8-1 e 3.7.0-1 nei nomi dei pacchetti elencati rappresentano solo un esempio.

- `php-8.1.8-1.amzn2.x86_64.rpm`
- `php-bcmath-8.1.8-1.amzn2.x86_64.rpm`
- `php-cli-8.1.8-1.amzn2.x86_64.rpm`
- `php-common-8.1.8-1.amzn2.x86_64.rpm`
- `php-dba-8.1.8-1.amzn2.x86_64.rpm`
- `php-dbg-8.1.8-1.amzn2.x86_64.rpm`
- `php-debuginfo-8.1.8-1.amzn2.x86_64.rpm`
- `php-devel-8.1.8-1.amzn2.x86_64.rpm`
- `php-embedded-8.1.8-1.amzn2.x86_64.rpm`

- `php-enchant-8.1.8-1.amzn2.x86_64.rpm`
- `php-fpm-8.1.8-1.amzn2.x86_64.rpm`
- `php-gd-8.1.8-1.amzn2.x86_64.rpm`
- `php-gmp-8.1.8-1.amzn2.x86_64.rpm`
- `php-intl-8.1.8-1.amzn2.x86_64.rpm`
- `php-ldap-8.1.8-1.amzn2.x86_64.rpm`
- `php-mbstring-8.1.8-1.amzn2.x86_64.rpm`
- `php-mysqlnd-8.1.8-1.amzn2.x86_64.rpm`
- `php-odbc-8.1.8-1.amzn2.x86_64.rpm`
- `php-opcache-8.1.8-1.amzn2.x86_64.rpm`
- `php-pdo-8.1.8-1.amzn2.x86_64.rpm`
- `php-pear-1.10.13-1.amzn2.noarch.rpm`
- `php-pgsql-8.1.8-1.amzn2.x86_64.rpm`
- `php-process-8.1.8-1.amzn2.x86_64.rpm`
- `php-pspell-8.1.8-1.amzn2.x86_64.rpm`
- `php-snmp-8.1.8-1.amzn2.x86_64.rpm`
- `php-soap-8.1.8-1.amzn2.x86_64.rpm`
- `php-sodium-8.1.8-1.amzn2.x86_64.rpm`
- `php-xml-8.1.8-1.amzn2.x86_64.rpm`
- `php-pecl-imagick-3.7.0-1.amzn2.x86_64.rpm`
- `php-pecl-imagick-debuginfo-3.7.0-1.amzn2.x86_64.rpm`
- `php-pecl-imagick-devel-3.7.0-1.amzn2.noarch.rpm`

È possibile utilizzare i pacchetti PEAR e PECL per installare estensioni comuni. Per ulteriori informazioni su PEAR, consulta il sito web PEAR PHP Extension and Application Repository. Per ulteriori informazioni su PECL, consulta il sito Web PECL extension.

I seguenti comandi di esempio installano le estensioni Memcached.

```
$pecl install memcache
```

Oppure puoi inoltre utilizzare quanto segue:

```
$pear install pecl/memcache
```

I seguenti comandi di esempio installano le estensioni Redis.

```
$pecl install redis
```

Oppure puoi inoltre utilizzare quanto segue:

```
$pear install pecl/redis
```

Configurazione dell'ambiente PHP

Puoi utilizzare la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3, configurare variabili che l'applicazione è in grado di leggere dall'ambiente e modificare le impostazioni PHP.

Per configurare l'ambiente PHP nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Impostazioni PHP

- Proxy server (Server proxy): il server proxy da utilizzare nelle istanze dell'ambiente. Per impostazione predefinita, viene utilizzato nginx.
- Document root (Root del documento): la cartella che contiene la pagina predefinita del tuo sito. Se la pagina di benvenuto non è nella radice del bundle di origine, specifica la cartella che lo contiene rispetto al percorso principale. Ad esempio, `/public` se la pagina di benvenuto si trova in una cartella chiamata `public`.
- Memory limit (Limite di memoria): la quantità massima di memoria che uno script può allocare. Ad esempio, `512M`.
- Zlib output compression (Compressione Zlib dell'output): impostare su `On` per comprimere le risposte.

- Allow URL fopen (Consenti URL fopen): impostare su `Off` per evitare che gli script scarichino file da sedi remote.
- Display errors (Visualizza errori): impostare su `On` per visualizzare messaggi di errore interni per il debug.
- Max execution time (Tempo massimo di esecuzione): tempo massimo, espresso in secondi, durante il quale a uno script è consentita l'esecuzione prima di essere terminato dall'ambiente.

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Enable log file rotation to Amazon S3 (Abilita rotazione file di log in Amazon S3): specifica se i file di log per le istanze Amazon EC2 dell'applicazione debbano essere copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Proprietà dell'ambiente

La sezione Environment Properties (Proprietà ambiente) consente di specificare le impostazioni di configurazione dell'ambiente nelle istanze Amazon EC2 che eseguono l'applicazione. Queste impostazioni sono passate come coppie chiave-valore all'applicazione.

Il codice dell'applicazione può accedere alle proprietà dell'ambiente utilizzando `$_SERVER` o la funzione `get_cfg_var`.

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

Per ulteriori informazioni, consultare Variabili di ambiente e altre impostazioni software.

Namespace per la configurazione

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

I seguenti namespace configurano sia il servizio proxy che le opzioni specifiche di PHP:

- `aws:elasticbeanstalk:environment:proxy:staticfiles`— configura il proxy dell'ambiente per servire file statici. È possibile definire i mapping dei percorsi virtuali alle directory dell'applicazione.
- `aws:elasticbeanstalk:environment:proxy`— specificare il server proxy dell'ambiente.
- `aws:elasticbeanstalk:container:php:phpini`— configura le opzioni specifiche di PHP. Questo spazio dei nomi include`composer_options`, che non è disponibile nella console Elastic Beanstalk. Questa opzione imposta le opzioni personalizzate da utilizzare durante l'installazione delle dipendenze utilizzando Composer tramite il comando. `composer.phar install` Per ulteriori informazioni su questo comando, incluse le opzioni disponibili, vedete install sul sito getcomposer.org.

Il seguente file di configurazione di esempio specifica un'opzione di file statici che esegue il mapping di una directory denominata `staticimages` al percorso `/images` e mostra le impostazioni per ciascuna delle opzioni disponibili nello spazio dei nomi `aws:elasticbeanstalk:container:php:phpini`:

Example.ebextensions/php-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
    memory_limit: 128M
    zlib.output_compression: "Off"
    allow_url_fopen: "On"
```

```
    display_errors: "Off"
    max_execution_time: 60
    composer_options: vendor/package
```

> **Note**
>
> Lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles` non è definito sui rami della piattaforma PHP AMI Amazon Linux (precedente Amazon Linux 2).

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Installazione delle dipendenze dell'applicazione PHP Elastic Beanstalk

Questo argomento descrive come configurare l'applicazione per installare altri pacchetti PHP necessari. L'applicazione potrebbe avere delle dipendenze da altri pacchetti PHP. È possibile configurare l'applicazione per installare queste dipendenze nelle istanze Amazon Elastic Compute Cloud (Amazon EC2) dell'ambiente. In alternativa, è possibile includere le dipendenze dell'applicazione nel bundle di origine e distribuirle con l'applicazione. Nella sezione seguente vengono illustrati entrambi i metodi.

Utilizzo di un file Composer per installare le dipendenze sulle istanze

Utilizza un file `composer.json` nella radice dell'origine del progetto per utilizzare Composer per installare i pacchetti di cui l'applicazione ha bisogno sulle istanze Amazon EC2 dell'ambiente.

Example composer.json

```
{
    "require": {
        "monolog/monolog": "1.0.*"
    }
}
```

Quando è presente un file `composer.json`, Elastic Beanstalk esegue `composer.phar install` per installare le dipendenze. È possibile aggiungere opzioni al comando impostando l'opzione `composer_options` nello spazio dei nomi `aws:elasticbeanstalk:container:php:phpini`.

Inclusione delle dipendenze nel bundle di origine

Se l'applicazione ha un numero elevato di dipendenze, installarle potrebbe richiedere molto tempo. Ciò può aumentare le operazioni di distribuzione e dimensionamento, poiché le dipendenze vengono installate in ogni nuova istanza.

Per evitare che ciò influisca negativamente sui tempi di installazione, utilizza Composer nel tuo ambiente di sviluppo per risolvere le dipendenze e installarle nella cartella `vendor`.

Come includere dipendenze nel tuo bundle di origine dell'applicazione

1. Esegui il comando seguente:

   ```
   % composer install
   ```

2. Includi la cartella `vendor` generata nella radice del bundle di origine dell'applicazione.

Quando Elastic Beanstalk individua una cartella `vendor` sull'istanza, ignora il file `composer.json` (anche se esiste). L'applicazione usa quindi le dipendenze dalla cartella `vendor`.

Aggiornamento di Composer su Elastic Beanstalk

Questo argomento descrive come configurare Elastic Beanstalk per mantenere aggiornato Composer. Potrebbe essere necessario aggiornare Composer se viene visualizzato un errore quando si tenta di installare pacchetti con un file Composer o se non si riesce a utilizzare la versione più recente della piattaforma. Tra un aggiornamento della piattaforma e l'altro, puoi aggiornare Composer nelle istanze del tuo ambiente utilizzando i file di configurazione nella tua cartella. `.ebextensions`

È possibile aggiornare automaticamente Composer con la seguente configurazione.

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0
```

La seguente impostazione dell'opzione imposta la variabile di `COMPOSER_HOME` ambiente, che configura la posizione della cache di Composer.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
```

```
    value: /home/webapp/composer-home
```

Puoi combinarli entrambi nello stesso file di configurazione nella tua `.ebextensions` cartella.

Example.ebextensions/composer.config

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0

option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

> **Note**
>
> A causa degli aggiornamenti all'installazione di Composer nella versione della piattaforma del 22 febbraio 2024 e nella versione della AL2023 piattaforma del 28 febbraio 2024, AL2 l'aggiornamento automatico di Composer potrebbe non riuscire se `COMPOSER_HOME` impostato al momento dell'esecuzione dell'aggiornamento automatico.
> I seguenti comandi combinati non verranno eseguiti: `export COMPOSER_HOME=/home/webapp/composer-home && /usr/bin/composer.phar self-update 2.7.0` Tuttavia, l'esempio precedente funzionerà. Nell'esempio precedente, l'impostazione dell'opzione for non `COMPOSER_HOME` verrà passata all'`01updateComposer`esecuzione e non verrà impostata durante l'esecuzione del comando di aggiornamento automatico.

> **Important**
>
> Se ometti il numero di versione dal comando `composer.phar self-update`, Composer verrà aggiornato all'ultima versione disponibile ogni volta che distribuisci il codice sorgente e quando le nuove istanze vengono assegnate da Auto Scaling. Questo potrebbe causare il fallimento del dimensionamento di operazioni e distribuzioni se viene rilasciata una versione di Composer non compatibile con la tua applicazione.

Per ulteriori informazioni sulle piattaforme PHP di Elastic Beanstalk, inclusa la versione di Composer, consultare le versioni della piattaforma PHP nel documento Piattaforme AWS Elastic Beanstalk .

Estendere php.ini nella configurazione di Elastic Beanstalk

Utilizza un file di configurazione con un blocco `files` da aggiungere a un file `.ini` per `/etc/php.d/` sulle istanze nel tuo ambiente. Il file di configurazione principale, `php.ini`, preleva le impostazioni dai file in questa cartella in ordine alfabetico. Molte estensioni sono abilitate per impostazione predefinita dai file in questa cartella.

Example.ebextensions/mongo.config

```
files:
  "/etc/php.d/99mongo.ini":
    mode: "000755"
    owner: root
    group: root
    content: |
      extension=mongo.so
```

Esempi avanzati di PHP in Elastic Beanstalk

Per iniziare a utilizzare le applicazioni PHP AWS Elastic Beanstalk, è necessario un pacchetto di sorgenti dell'applicazione da caricare come prima versione dell'applicazione da distribuire in un ambiente.

Ti consigliamo QuickStart per PHP di iniziare con una semplice applicazione PHP distribuita con l'EB CLI.

Esempi PHP avanzati

- Aggiungere un'istanza Amazon RDS DB al tuo ambiente PHP Elastic Beanstalk
- Distribuzione di un'applicazione Laravel su Elastic Beanstalk
- Distribuzione di un'applicazione CakePHP su Elastic Beanstalk
- Distribuzione di un'applicazione Symfony su Elastic Beanstalk
- Distribuzione di un'applicazione PHP a elevata disponibilità con un database Amazon RDS esterno in Elastic Beanstalk
- Implementazione di un WordPress sito Web ad alta disponibilità con un database Amazon RDS esterno su Elastic Beanstalk
- Distribuzione di un sito Web Drupal a elevata disponibilità con un database Amazon RDS esterno in Elastic Beanstalk

Aggiungere un'istanza Amazon RDS DB al tuo ambiente PHP Elastic Beanstalk

Questo argomento fornisce istruzioni per creare un Amazon RDS utilizzando la console Elastic Beanstalk. Puoi utilizzare un'istanza database Amazon Relational Database Service (Amazon RDS) per archiviare i dati raccolti e modificati dall'applicazione. Il database può essere accoppiato all'ambiente e gestito da Elastic Beanstalk oppure può essere creato e gestito esternamente da un altro servizio. In queste istruzioni il database è accoppiato all'ambiente e gestito da Elastic Beanstalk. Per ulteriori informazioni sull'integrazione di un Amazon RDS con Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Sections

- Aggiunta di un'istanza database all'ambiente
- Download di un driver
- Connessione a un database con PDO o My SQLi
- Connessione a un database con Symfony

Aggiunta di un'istanza database all'ambiente

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'aggiunta di un'istanza database richiede circa 10 minuti. Quando l'aggiornamento dell'ambiente è completo, il nome host dell'istanza database e altre informazioni di connessione sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente:

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Per ulteriori informazioni sulla configurazione di un'istanza di database accoppiata a un ambiente Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Download di un driver

Per usare PHP Data Objects (PDO) per la connessione al database, installa il driver corrispondente al motore di database scelto.

- MySQL – `PDO_MYSQL`
- PostgreSQL – `PDO_PGSQL`
- Oracle: `PDO_OCI`

- SQL Server – PDO_SQLSRV

Per ulteriori informazioni, consulta http://php.net/manual/en/pdo.installation.php.

Connessione a un database con PDO o My SQLi

Puoi utilizzare `$_SERVER[`VARIABLE`]` per leggere le informazioni di connessione dall'ambiente.

Per un PDO, crea un nome di origine dati (DSN) dall'host, dalla porta e dal nome. Trasferisci il DSN al costruttore per il PDO con il nome utente e la password del database.

Example Connessione a un database RDS con PDO - MySQL

```
<?php
$dbhost = $_SERVER['RDS_HOSTNAME'];
$dbport = $_SERVER['RDS_PORT'];
$dbname = $_SERVER['RDS_DB_NAME'];
$charset = 'utf8' ;

$dsn = "mysql:host={$dbhost};port={$dbport};dbname={$dbname};charset={$charset}";
$username = $_SERVER['RDS_USERNAME'];
$password = $_SERVER['RDS_PASSWORD'];

$pdo = new PDO($dsn, $username, $password);
?>
```

Per gli altri driver, sostituisci `mysql` con il nome del tuo driver: `pgsql`, `oci` o `sqlsrv`.

Per MySQLi, passa il nome host, il nome utente, la password, il nome del database e la porta al `mysqli` costruttore.

Example Connessione a un database RDS con mysqli_connect()

```
$link = new mysqli($_SERVER['RDS_HOSTNAME'], $_SERVER['RDS_USERNAME'],
 $_SERVER['RDS_PASSWORD'], $_SERVER['RDS_DB_NAME'], $_SERVER['RDS_PORT']);
```

Connessione a un database con Symfony

Per Symfony 3.2 e versioni successive, puoi utilizzare `%env(PROPERTY_NAME)%` per impostare i parametri di database in un file di configurazione in base alle proprietà dell'ambiente stabilite da Elastic Beanstalk.

Example app/config/parameters.yml

```
parameters:
    database_driver:   pdo_mysql
    database_host:     '%env(RDS_HOSTNAME)%'
    database_port:     '%env(RDS_PORT)%'
    database_name:     '%env(RDS_DB_NAME)%'
    database_user:     '%env(RDS_USERNAME)%'
    database_password: '%env(RDS_PASSWORD)%'
```

Per ulteriori informazioni, consultare la sezione sui parametri esterni (Symfony 3.4).

Per le versioni precedenti di Symfony, le variabili di ambiente sono accessibili solo se iniziano con `SYMFONY__`. Questo significa che le proprietà di ambiente definite da Elastic Beanstalk non sono accessibili e che occorre definire delle proprietà di ambiente personalizzate per trasmettere le informazioni di connessione a Symfony.

Per connetterti a un database con Symfony 2, crea una proprietà dell'ambiente per ciascun parametro. Successivamente, utilizza `%property.name%` per accedere alla variabile trasformata per Symfony in un file di configurazione. Ad esempio, si può accedere a una proprietà di ambiente denominata `SYMFONY__DATABASE__USER` come `database.user`.

```
    database_user:     "%database.user%"
```

Per ulteriori informazioni, consultare la sezione sui parametri esterni (Symfony 2.8).

Distribuzione di un'applicazione Laravel su Elastic Beanstalk

Laravel è un framework open source model-view-controller (MVC) per PHP. Questo tutorial illustra il processo di generazione di un'applicazione Laravel, la sua distribuzione in un AWS Elastic Beanstalk ambiente e la configurazione per la connessione a un'istanza di database Amazon Relational Database Service (Amazon RDS).

Sections

- Prerequisiti
- Avvio di un ambiente Elastic Beanstalk
- Installazione di Laravel e creazione di un sito Web
- Distribuzione dell'applicazione

- Configurazione delle impostazioni di Composer
- Aggiunta di un database all'ambiente
- Pulizia
- Fasi successive

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata per Windows di Ubuntu e Bash.

Laravel 6 richiede PHP 7.2 o versioni successive. Richiede anche le estensioni PHP riportate nella sezione dei requisiti del server nella documentazione ufficiale di Laravel. Segui le istruzioni per installare PHP e Composer.

Per informazioni sul supporto e sulla manutenzione di Laravel, vedere l'argomento relativo alle policy di supporto nella documentazione ufficiale di Laravel.

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma PHP e accetta le impostazioni predefinite e il codice di esempio.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced

2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.

- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon S3.

Installazione di Laravel e creazione di un sito Web

Composer è in grado di installare Laravel e di creare un progetto funzionante con un unico comando:

```
~$ composer create-project --prefer-dist laravel/laravel eb-laravel
```

Composer installa Laravel e le relative dipendenze e genera un progetto predefinito.

Se riscontri problemi durante l'installazione di Laravel, vai all'argomento di installazione nella documentazione ufficiale: 6.x. https://laravel.com/docs/

Distribuzione dell'applicazione

Crea un bundle di origine che contiene i file creati da Composer. Il comando seguente crea un bundle di origine denominato `laravel-default.zip`. Esclude i file nella cartella `vendor`, che occupano molto spazio e non sono necessari per la distribuzione dell'applicazione in Elastic Beanstalk.

```
~/eb-laravel$ zip ../laravel-default.zip -r * .[^.]* -x "vendor/*"
```

Carica il bundle di origine in Elastic Beanstalk per distribuire Laravel nell'ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

> **Note**
>
> Per ottimizzare ulteriormente il bundle di origine, inizializza un repository Git e utilizza il comando `git archive` per creare il bundle di origine. Il progetto predefinito di Laravel include un file `.gitignore` che indica a Git di escludere la cartella `vendor` e gli altri file non necessari per la distribuzione.

Configurazione delle impostazioni di Composer

Al termine della distribuzione, fai clic sull'URL per aprire la tua applicazione Laravel nel browser:

Di cosa si tratta? Per impostazione predefinita, Elastic Beanstalk fornisce la radice del progetto al percorso root del sito Web. In questo caso, tuttavia, la pagina predefinita (`index.php`) è a un livello inferiore nella cartella `public`. Puoi eseguire una verifica aggiungendo `/public` all'URL. Ad esempio, `http://laravel.us-east-2.elasticbeanstalk.com/public`.

Per distribuire l'applicazione Laravel nel percorso root, utilizza la console Elastic Beanstalk per configurare la radice del documento per il sito Web.

Per configurare la radice del documento del sito Web

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. In Document root (Radice documento), immettere **`/public`**.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.
7. Al termine dell'aggiornamento, fai clic sull'URL per riaprire il tuo sito all'interno del browser.

Fino qui, tutto bene. Nella fase successiva potrai aggiungere un database all'ambiente e configurare Laravel in modo che si connetta a esso.

Aggiunta di un database all'ambiente

Avvia un'istanza database RDS nel tuo ambiente Elastic Beanstalk. Puoi usare database MySQL o PostgreSQL con SQLServer Laravel su Elastic Beanstalk. Per questo esempio, utilizziamo MySQL.

Per aggiungere un'istanza database RDS al tuo ambiente Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. In Engine (Motore), selezionare mysql.
6. Digitare username (Nome utente) e password (Password) principali. Elastic Beanstalk fornirà questi valori alla tua applicazione utilizzando le proprietà dell'ambiente.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

La creazione di un'istanza di database richiede circa 10 minuti. Per ulteriori informazioni sui database associati a un ambiente Elastic Beanstalk, consultare Aggiunta di un database all'ambiente Elastic Beanstalk.

Nel frattempo, è possibile aggiornare il codice sorgente per leggere le informazioni di connessione dall'ambiente. Elastic Beanstalk fornisce i dettagli di connessione tramite variabili di ambiente, ad esempio `RDS_HOSTNAME`, a cui puoi accedere dalla tua applicazione.

La configurazione del database Laravel è memorizzata in un file denominato `database.php` nella cartella `config` del tuo codice di progetto. Individua la voce `mysql` e modifica le variabili `host`, `database`, `username` e `and password` per leggere i valori corrispondenti da Elastic Beanstalk:

Example~/Eb- .php laravel/config/database

```
...
    'connections' => [

        'sqlite' => [
```

```
            'driver' => 'sqlite',
            'database' => env('DB_DATABASE', database_path('database.sqlite')),
            'prefix' => '',
        ],

        'mysql' => [
            'driver' => 'mysql',
            'host' => env('RDS_HOSTNAME', '127.0.0.1'),
            'port' => env('RDS_PORT', '3306'),
            'database' => env('RDS_DB_NAME', 'forge'),
            'username' => env('RDS_USERNAME', 'forge'),
            'password' => env('RDS_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'strict' => true,
            'engine' => null,
        ],
...
```

Per verificare che la connessione al database sia configurata correttamente, aggiungi il codice a `index.`php per connetterti al database e aggiungi un codice per la risposta predefinita:

Example~/Eb- laravel/public/index .php

```
...
if(DB::connection()->getDatabaseName())
{
   echo "Connected to database ".DB::connection()->getDatabaseName();
}
$response->send();
...
```

Quando l'istanza database ha terminato l'avvio, raggruppa e distribuisci l'applicazione aggiornata nel tuo ambiente.

Per aggiornare l'ambiente Elastic Beanstalk

1. Creare un nuovo bundle di origine:

   ```
   ~/eb-laravel$ zip ../laravel-v2-rds.zip -r * .[^.]* -x "vendor/*"
   ```

2. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
3. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
4. Selezionare Upload and Deploy (Carica e distribuisci).
5. Selezionare Browse (Sfoglia) e caricare `laravel-v2-rds.zip`.
6. Selezionare Deploy (Distribuisci).

La distribuzione di una nuova versione dell'applicazione richiede meno di un minuto. Al termine della distribuzione, aggiorna la pagina Web di nuovo per verificare che la connessione al database sia riuscita:

Pulizia

Dopo aver finito di lavorare con il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina AWS tutte le risorse correlate, come istanze Amazon EC2, istanze di database, sistemi di bilanciamentodel carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Inoltre, puoi terminare le risorse di database create al di fuori dell'ambiente Elastic Beanstalk. Quando termini un'istanza database Amazon RDS, puoi acquisire uno snapshot e successivamente ripristinare i dati in un'altra istanza.

Per terminare l'istanza database RDS

1. Apri la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegli l'istanza database.
4. Scegli Actions (Operazioni), quindi Delete (Elimina).
5. Scegliere se creare una snapshot, quindi selezionare Delete (Elimina).

Fasi successive

Per ulteriori informazioni su Laravel, visita il sito Web ufficiale di Laravel su laravel.com.

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

In questo tutorial, hai utilizzato la console Elastic Beanstalk per configurare le opzioni Composer. Per rendere questa configurazione parte della tua origine delle applicazioni, puoi utilizzare un file di configurazione come il seguente.

Example.ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Per ulteriori informazioni, consultare Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

L'esecuzione di un'istanza database Amazon RDS nell'ambiente Elastic Beanstalk è perfetta per le attività di sviluppo e test, ma vincola il ciclo di vita del database al tuo ambiente. Per istruzioni

sulla connessione a un database in esecuzione al di fuori del tuo ambiente, consulta Aggiungere un'istanza Amazon RDS DB al tuo ambiente PHP Elastic Beanstalk.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Distribuzione di un'applicazione CakePHP su Elastic Beanstalk

CakePHP è un framework open source, MVC per PHP. Questo tutorial ti guiderà attraverso il processo di creazione di un progetto CakePHP, la distribuzione in un ambiente Elastic Beanstalk e la configurazione per connetterlo a un'istanza database Amazon RDS.

Sections

- Prerequisiti
- Avvio di un ambiente Elastic Beanstalk
- Installazione di CakePHP e creazione di un sito Web
- Distribuzione dell'applicazione
- Aggiunta di un database all'ambiente
- Pulizia
- Fasi successive

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata con Windows di Ubuntu e Bash.

CakePHP 4 richiede PHP 7.4 o successivo. Richiede anche le estensioni PHP elencate nella documentazione ufficiale di installazione di CakePHP . È necessario installare sia PHP che Composer.

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma PHP e accetta le impostazioni predefinite e il codice di esempio.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di

bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.

- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *`subdomain region`*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon S3.

Installazione di CakePHP e creazione di un sito Web

Composer è in grado di installare CakePHP e di creare un progetto funzionante con un unico comando:

```
~$ composer create-project --prefer-dist cakephp/app eb-cake
```

Composer installa CakePHP e circa 20 dipendenze e genera un progetto predefinito.

In caso di problemi durante l'installazione di CakePHP, visita l'argomento relativo all'installazione nella documentazione ufficiale: http://book.cakephp.org/4.0/en/installation.html

Distribuzione dell'applicazione

Crea un bundle di origine che contiene i file creati da Composer. Il comando seguente crea un bundle di origine denominato `cake-default.zip`. Esclude i file nella cartella `vendor`, che occupano molto spazio e non sono necessari per la distribuzione dell'applicazione in Elastic Beanstalk.

```
eb-cake zip ../cake-default.zip -r * .[^.]* -x "vendor/*"
```

Carica il bundle di origine in Elastic Beanstalk per distribuire CakePHP nell'ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).

6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

> **Note**
>
> Per ottimizzare ulteriormente il bundle di origine, inizializza un repository Git e utilizza il comando `git archive` per creare il bundle di origine. Il progetto predefinito di Symfony include un file `.gitignore` che indica a Git di escludere la cartella `vendor` e gli altri file non necessari per la distribuzione.

Una volta completato il processo, fai clic sull'URL per aprire la tua applicazione CakePHP nel browser.

Fino qui, tutto bene. Nella fase successiva potrai aggiungere un database all'ambiente e configurare CakePHP in modo che si connetta a esso.

Aggiunta di un database all'ambiente

Avvia un'istanza database Amazon RDS nel tuo ambiente Elastic Beanstalk. Puoi usare database MySQL o PostgreSQL con SQLServer CakePHP su Elastic Beanstalk. Per questo esempio, utilizziamo PostgreSQL.

Per aggiungere un'istanza database Amazon RDS al tuo ambiente Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. In Database, scegliere Edit (Modifica).
5. In DB engine (Motore DB), scegliere postgres.
6. Digitare username (Nome utente) e password (Password) principali. Elastic Beanstalk fornirà questi valori alla tua applicazione utilizzando le proprietà dell'ambiente.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

La creazione di un'istanza di database richiede circa 10 minuti. Nel frattempo, è possibile aggiornare il codice sorgente per leggere le informazioni di connessione dall'ambiente. Elastic Beanstalk fornisce

i dettagli di connessione tramite variabili di ambiente, ad esempio `RDS_HOSTNAME`, a cui puoi accedere dalla tua applicazione.

La configurazione del database CakePHP è memorizzata in un file denominato `app.php` nella cartella `config` del tuo codice di progetto. Apri il file e aggiungi un codice che legga le variabili di ambiente da `$_SERVER` e le assegna a variabili locali. Inserisci le righe evidenziate nell'esempio qui di seguito, dopo la prima linea (`<?php`):

Example~/Eb- .php cake/config/app

```
<?php
if (!defined('RDS_HOSTNAME')) {
  define('RDS_HOSTNAME', $_SERVER['RDS_HOSTNAME']);
  define('RDS_USERNAME', $_SERVER['RDS_USERNAME']);
  define('RDS_PASSWORD', $_SERVER['RDS_PASSWORD']);
  define('RDS_DB_NAME', $_SERVER['RDS_DB_NAME']);
}
return [
...
```

La connessione al database viene configurata ulteriormente nel file `app.php`. Trova la sezione seguente e modifica la configurazione di origine dati predefinita con il nome del driver corrispondente al motore di database (`Mysql`, `Sqlserver` o `Postgres`) e imposta le variabili `host`, `username`, `password` e `database` per leggere i valori corrispondenti da Elastic Beanstalk:

Example~/Eb- cake/config/app .php

```
...
     /**
     * Connection information used by the ORM to connect
     * to your application's datastores.
     * Drivers include Mysql Postgres Sqlite Sqlserver
     * See vendor\cakephp\cakephp\src\Database\Driver for complete list
     */
    'Datasources' => [
        'default' => [
            'className' => 'Cake\Database\Connection',
            'driver' => 'Cake\Database\Driver\Postgres',
            'persistent' => false,
            'host' => RDS_HOSTNAME,
            /*
```

```
             * CakePHP will use the default DB port based on the driver selected
             * MySQL on MAMP uses port 8889, MAMP users will want to uncomment
             * the following line and set the port accordingly
             */
            //'port' => 'non_standard_port_number',
            'username' => RDS_USERNAME,
            'password' => RDS_PASSWORD,
            'database' => RDS_DB_NAME,
            /*
             * You do not need to set this flag to use full utf-8 encoding (internal
 default since CakePHP 3.6).
             */
            //'encoding' => 'utf8mb4',
            'timezone' => 'UTC',
            'flags' => [],
            'cacheMetadata' => true,
            'log' => false,
...
```

Quando l'istanza database ha terminato l'avvio, raggruppa e distribuisci l'applicazione aggiornata nel tuo ambiente:

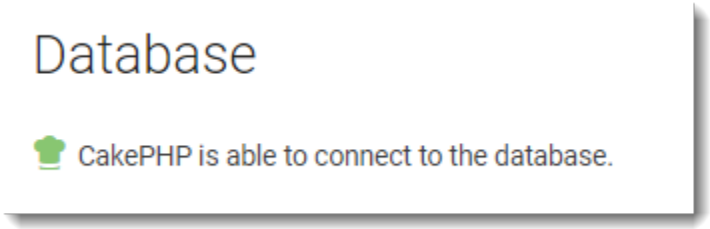
Per aggiornare l'ambiente Elastic Beanstalk

1. Creare un nuovo bundle di origine:

   ```
   ~/eb-cake$ zip ../cake-v2-rds.zip -r * .[^.]* -x "vendor/*"
   ```

2. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
3. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
4. Selezionare Upload and Deploy (Carica e distribuisci).
5. Selezionare Browse (Sfoglia) e carica `cake-v2-rds.zip`.
6. Selezionare Deploy (Distribuisci).

La distribuzione di una nuova versione dell'applicazione richiede meno di un minuto. Al termine della distribuzione, aggiorna la pagina Web di nuovo per verificare che la connessione al database sia riuscita:



Pulizia

Dopo aver finito di lavorare con il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina AWS tutte le risorse correlate, come istanze Amazon EC2, istanze di database, sistemi di bilanciamentodel carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Inoltre, puoi terminare le risorse di database create al di fuori dell'ambiente Elastic Beanstalk. Quando termini un'istanza database Amazon RDS, puoi acquisire uno snapshot e successivamente ripristinare i dati in un'altra istanza.

Per terminare l'istanza database RDS

1. Apri la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegli l'istanza database.
4. Scegli Actions (Operazioni), quindi Delete (Elimina).
5. Scegliere se creare una snapshot, quindi selezionare Delete (Elimina).

Fasi successive

Per ulteriori informazioni su CakePHP, leggi il manuale all'indirizzo book.cakephp.org.

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

L'esecuzione di un'istanza database Amazon RDS nell'ambiente Elastic Beanstalk è perfetta per le attività di sviluppo e test, ma vincola il ciclo di vita del database al tuo ambiente. Per istruzioni sulla connessione a un database in esecuzione al di fuori del tuo ambiente, consulta Aggiungere un'istanza Amazon RDS DB al tuo ambiente PHP Elastic Beanstalk.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Distribuzione di un'applicazione Symfony su Elastic Beanstalk

Symfony è un framework open source per lo sviluppo di applicazioni Web PHP dinamiche. Questo tutorial illustra il processo di generazione di un'applicazione Symfony e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

Sections

- Prerequisiti
- Avvio di un ambiente Elastic Beanstalk
- Installazione di Symfony e creazione di un sito Web
- Distribuzione dell'applicazione
- Configurazione delle impostazioni di Composer
- Pulizia
- Fasi successive

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows è possibile installare il sottosistema Windows per Linux per ottenere una versione integrata con Windows di Ubuntu e Bash.

Symfony 4.4.9 richiede PHP 7.1.3 o versioni successive. Richiede anche le estensioni PHP elencate nell'argomento dei requisiti tecnici nella documentazione ufficiale di installazione di Symfony. In questo tutorial viene utilizzato PHP 7.2 e la versione della piattaforma Elastic Beanstalk corrispondente. Prima di procedere, è necessario installare sia PHP che Composer.

Per informazioni sul supporto e sulla manutenzione di Symfony, consulta l'argomento relativo alle versioni di Symfony sul sito Web di Symfony. Per ulteriori informazioni sugli aggiornamenti relativi al supporto della versione PHP per Symfony 4.4.9, consulta Note di rilascio di Symfony 4.4.9 sul sito Web di Symfony.

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma PHP e accetta le impostazioni predefinite e il codice di esempio.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- EC2 istanza: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per eseguire app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza dell'istanza: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa consente al traffico HTTP proveniente dal sistema di bilanciamento del carico di raggiungere l' EC2 istanza su cui è in esecuzione la tua app web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza Load Balancer: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. `subdomain region`.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon S3.

Installazione di Symfony e creazione di un sito Web

Composer è in grado di installare Symfony e di creare un progetto funzionante con un unico comando:

```
~$ composer create-project symfony/website-skeleton eb-symfony
```

Composer installa Symfony e le relative dipendenze e genera un progetto predefinito.

In caso di problemi durante l'installazione di Symfony, consulta l'argomento relativo all'installazione nella documentazione ufficiale di Symfony.

Distribuzione dell'applicazione

Passare alla directory del progetto.

```
~$ cd eb-symfony
```

Crea un bundle di origine che contiene i file creati da Composer. Il comando seguente crea un bundle di origine denominato `symfony-default.zip`. Esclude i file nella cartella `vendor`, che occupano molto spazio e non sono necessari per la distribuzione dell'applicazione in Elastic Beanstalk.

```
eb-symfony$ zip ../symfony-default.zip -r * .[^.]* -x "vendor/*"
```

Carica il bundle di origine in Elastic Beanstalk per distribuire Symfony nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

> **Note**
>
> Per ottimizzare ulteriormente il bundle di origine, inizializza un repository Git e utilizza il comando `git archive` per creare il bundle di origine. Il progetto predefinito di Symfony include un file `.gitignore` che indica a Git di escludere la cartella `vendor` e gli altri file non necessari per la distribuzione.

Configurazione delle impostazioni di Composer

Al termine della distribuzione, fai clic sull'URL per aprire la tua applicazione Symfony nel browser.

Di cosa si tratta? Per impostazione predefinita, Elastic Beanstalk fornisce la radice del progetto al percorso principale del sito Web. In questo caso, tuttavia, la pagina predefinita (`app.php`) è a un livello inferiore nella cartella `web`. Puoi eseguire una verifica aggiungendo `/public` all'URL. Ad esempio, `http://symfony.us-east-2.elasticbeanstalk.com/public`.

Per distribuire l'applicazione Symfony nel percorso root, utilizza la console Elastic Beanstalk per configurare la radice del documento del sito Web.

Per configurare la radice del documento del sito Web

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. In Document root (Radice documento), immettere **`/public`**.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.
7. Al termine dell'aggiornamento, fai clic sull'URL per riaprire il tuo sito all'interno del browser.

Pulizia

Dopo aver finito di utilizzare il codice demo, puoi chiudere l'ambiente. Elastic Beanstalk elimina tutte le risorse AWS correlate, come istanze EC2 Amazon, istanze di database, sistemi di bilanciamento del carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Fasi successive

Per ulteriori informazioni su Symfony, consulta la presentazione di Symfony su symfony.com.

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

In questo tutorial, hai utilizzato la console Elastic Beanstalk per configurare le opzioni Composer. Per rendere questa configurazione parte della tua origine delle applicazioni, puoi utilizzare un file di configurazione come il seguente.

Example.ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Per ulteriori informazioni, consultare Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Symfony utilizza i propri file di configurazione per configurare le connessioni del database. Per istruzioni sulla connessione a un database con Symfony, consulta Connessione a un database con Symfony.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Distribuzione di un'applicazione PHP a elevata disponibilità con un database Amazon RDS esterno in Elastic Beanstalk

Questo tutorial illustra il processo di avvio di un'istanza DB RDS esterna e la configurazione di un ambiente ad alta disponibilità che esegue un'applicazione PHP per connettersi ad essa. AWS Elastic Beanstalk L'esecuzione di un'istanza database esterna a Elastic Beanstalk separa il database dal ciclo di vita dell'ambiente. Ciò consente di connettersi allo stesso database da più ambienti, sostituire un database per un altro o eseguire una distribuzione blu/verde senza influire sul database.

Il tutorial fa riferimento a un'applicazione PHP di esempio che utilizza un database MySQL per archiviare i dati di testo fornite dall'utente. L'applicazione di esempio utilizza i file di configurazione per configurare le impostazioni PHP e creare una tabella nel database che l'applicazione può utilizzare. Mostra inoltre come utilizzare un file Composer per installare i pacchetti durante la distribuzione.

Sections

- Prerequisiti
- Avvio di un'istanza database in Amazon RDS
- Creazione di un ambiente Elastic Beanstalk
- Configurazione di gruppi di sicurezza, proprietà dell'ambiente e dimensionamento
- Distribuzione di un'applicazione di esempio
- Pulizia
- Fasi successive

Prerequisiti

Prima di iniziare, scarica il pacchetto sorgente dell'applicazione di esempio da: -app-1.3.zip GitHub eb-demo-php-simple

Le procedure descritte in questo tutorial per le attività Amazon Relational Database Service (Amazon RDS) presuppongono che si stiano avviando risorse in un Amazon Virtual Private Cloud (Amazon VPC) predefinito. Tutti i nuovi account includono un VPC predefinito in ciascuna regione. Se non hai un VPC predefinito, le procedure variano. Consulta Utilizzo di Elastic Beanstalk con Amazon RDS per le istruzioni relative alle piattaforme EC2-Classic e alle piattaforme VPC personalizzate.

Avvio di un'istanza database in Amazon RDS

Per utilizzare un database esterno con un'applicazione in esecuzione in Elastic Beanstalk, avvia innanzitutto un'istanza database con Amazon RDS. Un'istanza avviata con Amazon RDS è completamente indipendente da Elastic Beanstalk e dagli ambienti Elastic Beanstalk e non verrà terminata o monitorata da Elastic Beanstalk.

Utilizza la console Amazon RDS per avviare un'istanza database Multi-AZ MySQL. La scelta di un'implementazione Multi-AZ garantisce che il database eseguirà il failover e continuerà a essere disponibile se l'istanza database di origine è inutilizzabile.

Per avviare un'istanza database RDS in un VPC predefinito

1. Aprire la console di RDS.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere Create database (Crea database).
4. Scegliere Standard Create (Creazione standard).

> **Important**
>
> Non selezionare Easy Create (Creazione rapida). Se viene selezionata questa opzione, non è possibile configurare le impostazioni necessarie per avviare questo database RDS.

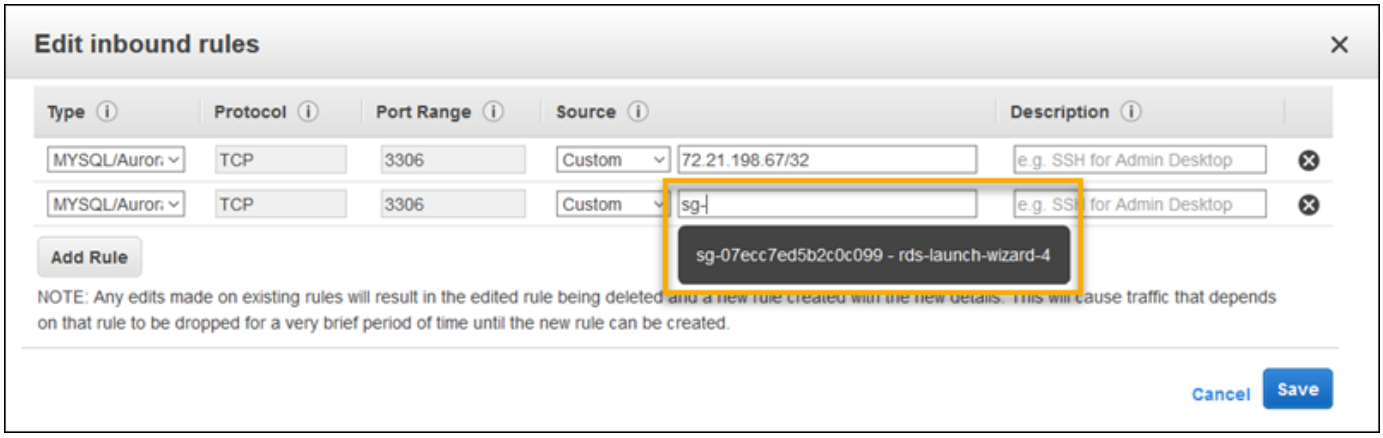
5. In Additional configuration (Configurazione aggiuntiva), digitare **ebdb** per Initial database name (Nome database iniziale).
6. Controllare le impostazioni di default e regolare queste impostazioni in base alle proprie esigenze specifiche. Prestare particolare attenzione alle seguenti opzioni:
   - DB instance class (Classe di istanze database): scegliere una dimensione di istanza con capacità di memoria e potenza di CPU sufficienti per il tuo carico di lavoro.
   - Implementazione Multi-AZ: per un'elevata disponibilità, impostala su Crea un nodo Replica/Reader Aurora in una AZ diversa.
   - Master username (Nome utente master) e Master password (Password master): nome utente e password del database. Prendi nota di queste impostazioni perché ti serviranno in seguito.
7. Verificare le impostazioni predefinite per le opzioni rimanenti e scegliere Create database (Crea database).

Quindi, modifica il gruppo di sicurezza associato all'istanza database per consentire il traffico in entrata sulla porta appropriata. Questo è lo stesso gruppo di sicurezza che collegherai successivamente all'ambiente Elastic Beanstalk, perciò la regola che aggiungi concederà l'autorizzazione di ingresso ad altre risorse nello stesso gruppo di sicurezza.

Per modificare le regole in entrata del gruppo di sicurezza associato all'istanza RDS

1. Aprire la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Nella sezione Connectivity (Connettività), prendere nota dei valori relativi a Subnets (Sottoreti), Security groups (Gruppi di sicurezza) ed Endpoint mostrati in questa pagina. In questo modo è possibile utilizzare queste informazioni in un secondo momento.
5. In Security (Sicurezza) viene visualizzato il gruppo di sicurezza associato all'istanza database. Aprire il collegamento per visualizzare il gruppo di sicurezza nella console Amazon EC2.
6. Nei dettagli del gruppo di sicurezza, scegliere Inbound (In entrata).

7. Scegli Modifica.
8. Selezionare Add Rule (Aggiungi regola).
9. In Type (Tipo), scegliere il motore di database utilizzato dall'applicazione.
10. In Source (Origine), digitare **`sg-`** per visualizzare un elenco dei gruppi di sicurezza disponibili. Scegliere il gruppo di sicurezza associato al gruppo Auto Scaling utilizzato con l'ambiente Elastic Beanstalk. In questo modo le istanze Amazon EC2 dell'ambiente possono accedere al database.



11. Scegli Save (Salva).

La creazione di un'istanza database richiede circa 10 minuti. Nel frattempo, crea il tuo ambiente Elastic Beanstalk.

Creazione di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma PHP e accetta le impostazioni predefinite e il codice di esempio. Dopo aver avviato l'ambiente, puoi configurare l'ambiente per la connessione al database, quindi distribuire l'applicazione di esempio da cui hai scaricato. GitHub

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).

5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.

- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene. L'istanza database di RDS avviata è al di fuori del tuo ambiente, per cui sei responsabile della gestione del suo ciclo di vita.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon S3.

Configurazione di gruppi di sicurezza, proprietà dell'ambiente e dimensionamento

Aggiungi il gruppo di sicurezza dell'istanza database all'ambiente in esecuzione. Con questa procedura Elastic Beanstalk predispone di nuovo tutte le istanze nell'ambiente con il gruppo di sicurezza aggiuntivo collegato.

Per aggiungere un gruppo di sicurezza al tuo ambiente

- Esegui una delle seguenti operazioni:
  - Per aggiungere un gruppo di sicurezza tramite la console Elastic Beanstalk
    a. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

b. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

c. Nel pannello di navigazione, seleziona Configuration (Configurazione).

d. Nella categoria di configurazione Instances (Istanze), scegliere Edit (Modifica).

e. In EC2 security groups (Gruppi di sicurezza EC2), scegliere il gruppo di sicurezza da collegare alle istanze, in aggiunta a quello creato da Elastic Beanstalk.

f. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

g. Leggere l'avviso, quindi selezionare Confirm (Conferma).

- Per aggiungere un gruppo di sicurezza tramite un file di configurazione, utilizzare il file di esempio `securitygroup-addexisting.config`.

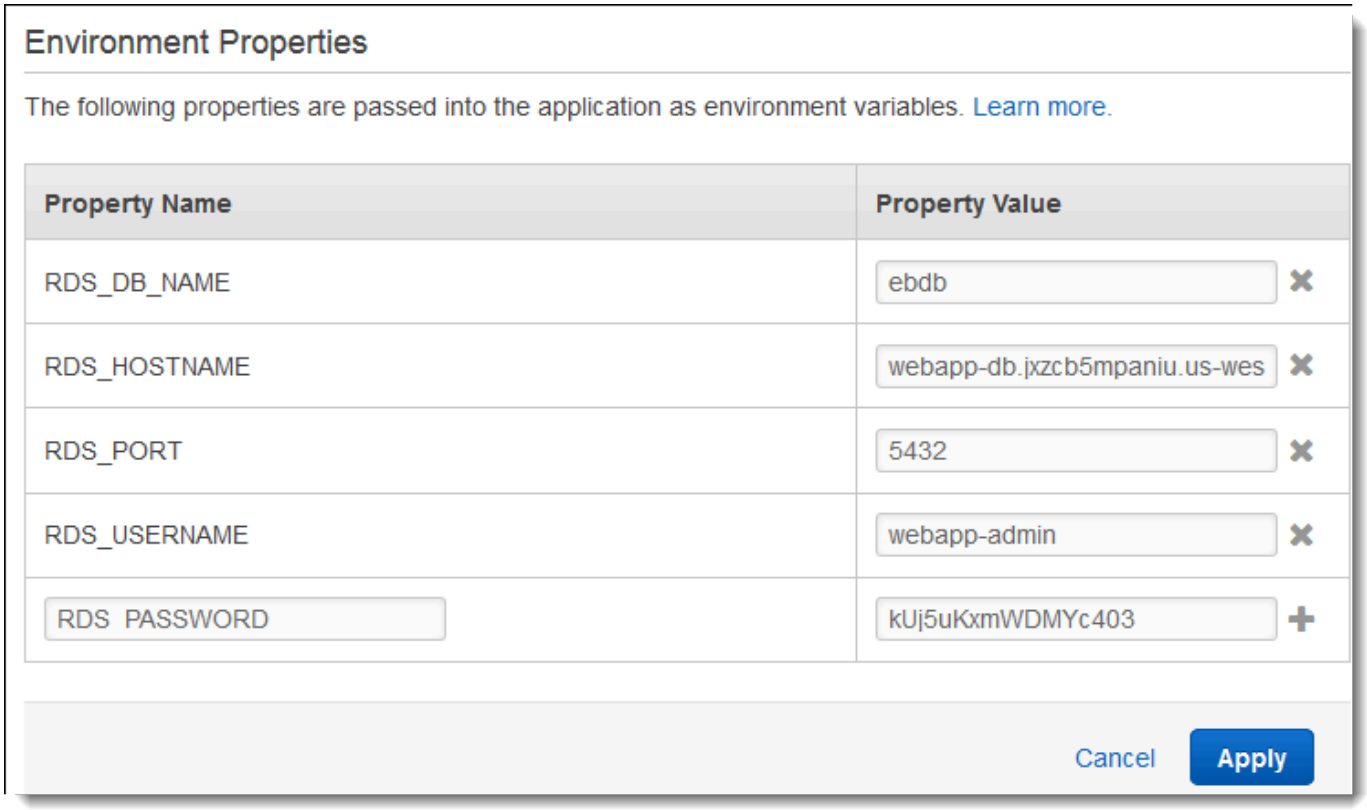
Quindi, utilizza le proprietà dell'ambiente per far passare le informazioni di connessione al tuo ambiente. L'applicazione di esempio utilizza un set di proprietà predefinito che corrisponde a quello configurato da Elastic Beanstalk quando viene effettuato il provisioning di un database all'interno del tuo ambiente.

Per configurare le proprietà dell'ambiente per un'istanza database Amazon RDS

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Nella sezione Environment properties (Proprietà ambiente), definire le variabili che l'applicazione deve leggere per formare una stringa di collegamento. Per la compatibilità con gli ambienti che dispongono di un'istanza database RDS integrata, utilizza i seguenti nomi e valori. Tutti i valori, ad eccezione della password, sono disponibili nella console RDS.

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e

Nome proprietà	Descrizione	Valore proprietà
		sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.



6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Infine, configura il gruppo Auto Scaling dell'ambiente con un conteggio delle istanze minime più elevato. Esegui almeno due istanze in qualsiasi momento per evitare che i server Web nell'ambiente rappresentino una singola possibilità di errore e per consentire di distribuire le modifiche senza interrompere il funzionamento del servizio.

Configurazione del gruppo Auto Scaling dell'ambiente per elevata disponibilità

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Auto Scaling Group (Gruppo Auto Scaling), impostare Min instances (Istanze min) su **2**.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Distribuzione di un'applicazione di esempio

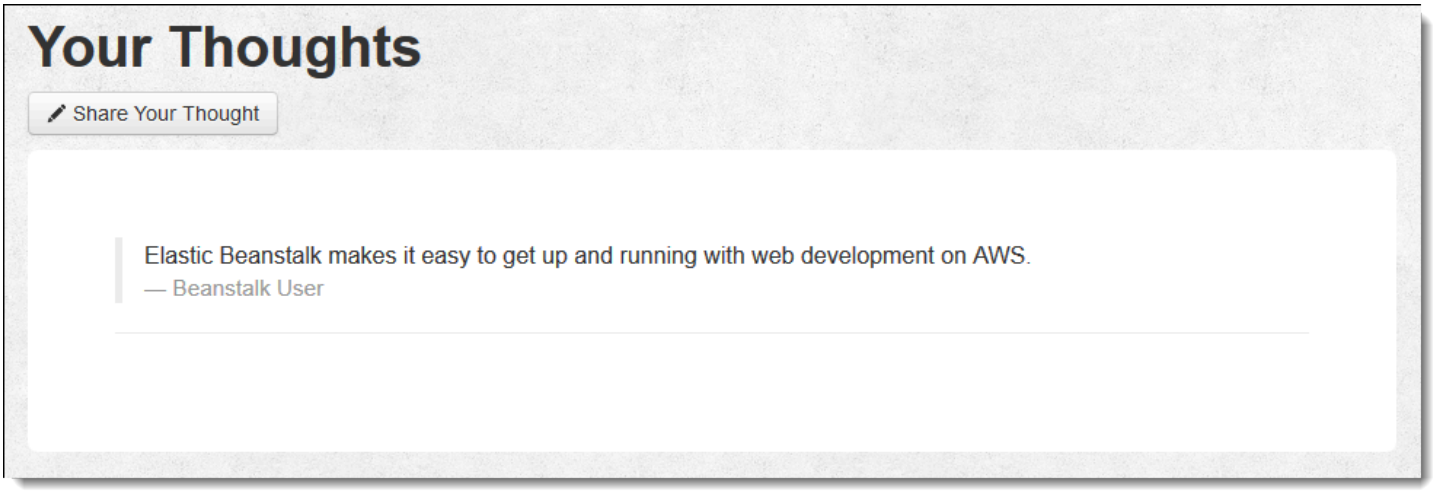
Ora l'ambiente è pronto per eseguire l'applicazione di esempio e connettersi ad Amazon RDS. Distribuisci l'applicazione di esempio nel tuo ambiente.

> **Note**
>
> Scarica il pacchetto sorgente da GitHub, se non l'hai già fatto: -app-1.3.zip eb-demo-php-simple

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Il sito raccoglie i commenti degli utenti e utilizza un database MySQL per archiviare i dati. Per aggiungere un commento, scegli Share Your Thought (Condividi la tua opinione), immetti un commento, quindi seleziona Submit Your Thought (Invia la tua opinione). L'applicazione web scrive il commento sul database, in modo che qualsiasi istanza nell'ambiente possa leggerlo; in questo modo, inoltre, il commento non andrà perso se le istanze diventano inutilizzabili.



Pulizia

Dopo aver finito di lavorare con il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina AWS tutte le risorse correlate, come istanze Amazon EC2, istanze di database, sistemi di bilanciamentodel carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Inoltre, puoi terminare le risorse di database create al di fuori dell'ambiente Elastic Beanstalk. Quando termini un'istanza database Amazon RDS, puoi acquisire uno snapshot e successivamente ripristinare i dati in un'altra istanza.

Per terminare l'istanza database RDS

1. Apri la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegli l'istanza database.

4. Scegli Actions (Operazioni), quindi Delete (Elimina).
5. Scegliere se creare una snapshot, quindi selezionare Delete (Elimina).

Fasi successive

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

L'applicazione di esempio utilizza i file di configurazione per configurare le impostazioni PHP e creare una tabella nel database, se non esiste già. È inoltre possibile utilizzare un file di configurazione per configurare le impostazioni del gruppo di sicurezza delle istanze durante la creazione dell'ambiente per evitare gli aggiornamenti di configurazione onerosi in termini di tempo. Per ulteriori informazioni, consultare Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Per sviluppo e test, puoi usare la funzionalità di Elastic Beanstalk per l'aggiunta di un'istanza database gestita direttamente nell'ambiente. Per istruzioni sulla configurazione di un database nell'ambiente, consultare Aggiunta di un database all'ambiente Elastic Beanstalk.

Se è necessario un database ad alte prestazioni, prendi in considerazione Amazon Aurora. Amazon Aurora è un motore di database compatibile con MySQL che offre caratteristiche di database commerciali a basso costo. Per connettere l'applicazione a un altro database, ripeti la procedura di configurazione del gruppo di sicurezza e aggiorna le proprietà dell'ambiente relative a RDS.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Implementazione di un WordPress sito Web ad alta disponibilità con un database Amazon RDS esterno su Elastic Beanstalk

Questo tutorial descrive come avviare un'istanza database Amazon RDS esterna a AWS Elastic Beanstalk, quindi come configurare un ambiente ad alta disponibilità che esegue un WordPress sito Web per connettersi ad esso. Il sito Web utilizza Amazon Elastic File System (Amazon EFS) come storage condiviso per i file caricati.

L'esecuzione di un'istanza database esterna a Elastic Beanstalk separa il database dal ciclo di vita dell'ambiente. Ciò consente di connettersi allo stesso database da più ambienti, sostituire un database per un altro o eseguire una distribuzione blu/verde senza influire sul database.

> **Note**
>
> Per informazioni aggiornate sulla compatibilità delle versioni di PHP con WordPress le versioni, consulta Compatibilità e WordPress versioni di PHP sul sito Web. WordPress È necessario fare riferimento a queste informazioni prima di eseguire l'aggiornamento a una nuova versione di PHP per le proprie implementazioni. WordPress

Argomenti

- Prerequisiti
- Avvio di un'istanza database in Amazon RDS
- Scarica WordPress
- Avvio di un ambiente Elastic Beanstalk
- Configurazione di gruppi di sicurezza e proprietà dell'ambiente
- Configurazione e distribuzione dell'applicazione
- Installa WordPress
- Aggiornamento di chiavi e salt
- Rimozione delle limitazioni di accesso
- Configurazione del gruppo Auto Scaling
- Aggiorna WordPress
- Eliminazione
- Fasi successive

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata con Windows di Ubuntu e Bash.

VPC di default

Le procedure Amazon Relational Database Service (Amazon RDS) descritte in questo tutorial presuppongono che si stiano avviando risorse in un Amazon Virtual Private Cloud (Amazon VPC) predefinito. Tutti i nuovi account includono un VPC predefinito in ogni AWS regione. Se non hai un VPC predefinito, le procedure variano. Consulta le istruzioni Utilizzo di Elastic Beanstalk con Amazon RDS per le EC2 piattaforme VPC classiche e personalizzate.

AWS Regioni

L'applicazione di esempio utilizza Amazon EFS, che funziona solo nelle AWS regioni che supportano Amazon EFS. Per informazioni sulle AWS regioni supportate, consulta Amazon Elastic File System Endpoints and Quotas nel. Riferimenti generali di AWS

Avvio di un'istanza database in Amazon RDS

Quando avvii un'istanza con Amazon RDS, è completamente indipendente da Elastic Beanstalk e dagli ambienti Elastic Beanstalk, e non verrà terminata né monitorata da Elastic Beanstalk.

Nelle fasi seguenti, si utilizzerà la console Amazon RDS per:

- Lanciare un database con il motore MySQL.
- Attivare una implementazione Multi-AZ. Questo crea uno standby in una zona di disponibilità (AZ) diversa per fornire ridondanza dei dati, eliminare i I/O blocchi e ridurre al minimo i picchi di latenza durante i backup di sistema.

Per avviare un'istanza database RDS in un VPC predefinito

1. Aprire la console di RDS.

2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere Create database (Crea database).
4. Scegliere Standard Create (Creazione standard).

> **Important**
>
> Non selezionare Easy Create (Creazione rapida). Se viene selezionata questa opzione, non è possibile configurare le impostazioni necessarie per avviare questo database RDS.

5. In Additional configuration (Configurazione aggiuntiva), digitare **ebdb** per Initial database name (Nome database iniziale).
6. Controllare le impostazioni di default e regolare queste impostazioni in base alle proprie esigenze specifiche. Prestare particolare attenzione alle seguenti opzioni:
   - DB instance class (Classe di istanze database): scegliere una dimensione di istanza con capacità di memoria e potenza di CPU sufficienti per il tuo carico di lavoro.
   - Implementazione Multi-AZ: per un'elevata disponibilità, impostala su Crea un nodo Replica/Reader Aurora in una AZ diversa.
   - Master username (Nome utente master) e Master password (Password master): nome utente e password del database. Prendi nota di queste impostazioni perché ti serviranno in seguito.
7. Verificare le impostazioni predefinite per le opzioni rimanenti e scegliere Create database (Crea database).

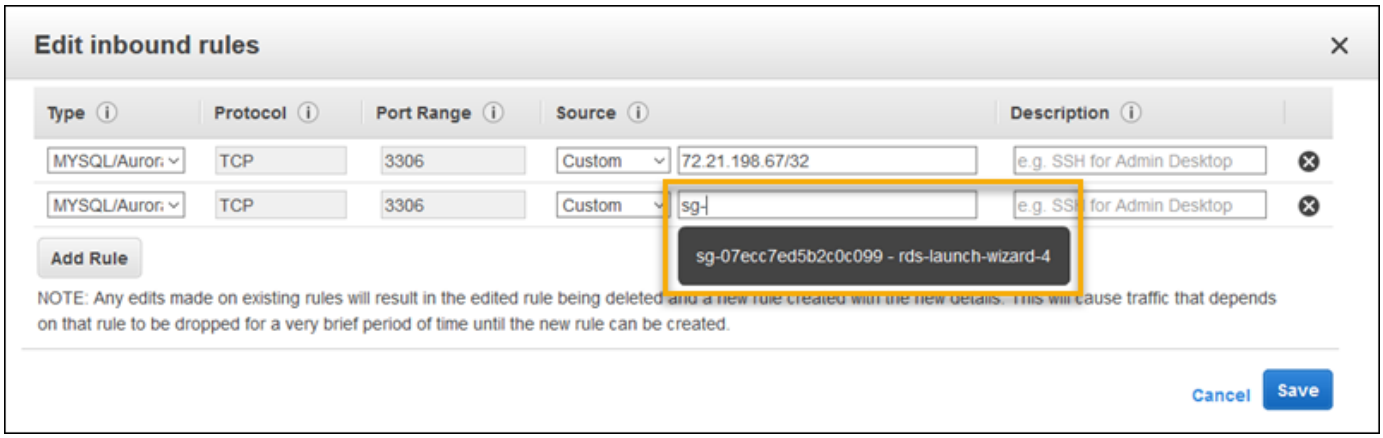
Dopo aver creato l'istanza DB, modificare il gruppo di sicurezza ad essa collegato in modo da consentire il traffico in ingresso sulla porta appropriata.

> **Note**
>
> Questo è lo stesso gruppo di sicurezza che collegherai successivamente all'ambiente Elastic Beanstalk, perciò la regola che aggiungi ora concederà l'autorizzazione di ingresso ad altre risorse nello stesso gruppo di sicurezza.

Per modificare le regole in entrata del gruppo di sicurezza associato all'istanza RDS

1. Aprire la console Amazon RDS.

2. Scegli Databases (Database).
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Nella sezione Connectivity (Connettività), prendere nota dei valori relativi a Subnets (Sottoreti), Security groups (Gruppi di sicurezza) ed Endpoint mostrati in questa pagina. In questo modo è possibile utilizzare queste informazioni in un secondo momento.
5. In Security (Sicurezza) viene visualizzato il gruppo di sicurezza associato all'istanza database. Apri il link per visualizzare il gruppo di sicurezza nella EC2 console Amazon.
6. Nei dettagli del gruppo di sicurezza, scegliere Inbound (In entrata).
7. Scegli Modifica.
8. Selezionare Add Rule (Aggiungi regola).
9. In Type (Tipo), scegliere il motore di database utilizzato dall'applicazione.
10. In Source (Origine), digitare **`sg-`** per visualizzare un elenco dei gruppi di sicurezza disponibili. Scegliere il gruppo di sicurezza associato al gruppo Auto Scaling utilizzato con l'ambiente Elastic Beanstalk. In questo modo EC2 le istanze Amazon nell'ambiente possono avere accesso al database.



11. Scegli Save (Salva).

La creazione di un'istanza database richiede circa 10 minuti. Nel frattempo, scarica WordPress e crea il tuo ambiente Elastic Beanstalk.

Scarica WordPress

Per prepararsi alla distribuzione WordPress AWS Elastic Beanstalk, è necessario copiare WordPress i file sul computer e fornire le informazioni di configurazione corrette.

Per creare un progetto WordPress

1. Scarica WordPress da wordpress.org.

```
~$curl https://wordpress.org/wordpress-6.2.tar.gz -o wordpress.tar.gz
```

2. Scarica i file di configurazione dal repository di esempio:

```
~$ wget https://github.com/aws-samples/eb-php-wordpress/releases/download/v1.1/eb-
php-wordpress-v1.zip
```

3. Estrai WordPress e modifica il nome della cartella.

```
~$ tar -xvf wordpress.tar.gz
~$ mv wordpress wordpress-beanstalk
~$ cd wordpress-beanstalk
```

4. Estrai i file di configurazione durante l' WordPress installazione.

```
~/wordpress-beanstalk$ unzip ../eb-php-wordpress-v1.zip
 creating: .ebextensions/
inflating: .ebextensions/dev.config
inflating: .ebextensions/efs-create.config
inflating: .ebextensions/efs-mount.config
inflating: .ebextensions/loadbalancer-sg.config
inflating: .ebextensions/wordpress.config
inflating: LICENSE
inflating: README.md
inflating: wp-config.php
```

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Dopo aver avviato l'ambiente, puoi configurarlo per connetterti al database, quindi distribuire il WordPress codice nell'ambiente.

Nelle fasi seguenti, si utilizzerà la console Elastic Beanstalk per:

- Creare un'applicazione Elastic Beanstalk utilizzando la piattaforma PHP gestita.
- Accettare le impostazioni predefinite e il codice di esempio.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

Risorse Elastic Beanstalk create

- EC2 istanza: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per eseguire app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza dell'istanza: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa consente al traffico HTTP proveniente dal sistema di bilanciamento del carico di raggiungere l' EC2 istanza su cui è in esecuzione la tua app web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza Load Balancer: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.

- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Poiché l'istanza Amazon RDS avviata è al di fuori del tuo ambiente, sei responsabile della gestione del suo ciclo di vita.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon S3.

Configurazione di gruppi di sicurezza e proprietà dell'ambiente

Aggiungi il gruppo di sicurezza dell'istanza database all'ambiente in esecuzione. Con questa procedura Elastic Beanstalk predispone di nuovo tutte le istanze nell'ambiente con il gruppo di sicurezza aggiuntivo collegato.

Per aggiungere un gruppo di sicurezza al tuo ambiente

- Esegui una delle seguenti operazioni:
  - Per aggiungere un gruppo di sicurezza tramite la console Elastic Beanstalk
    a. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
    b. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
    c. Nel pannello di navigazione, seleziona Configuration (Configurazione).
    d. Nella categoria di configurazione Instances (Istanze), scegliere Edit (Modifica).
    e. In Gruppi EC2 di sicurezza, scegli il gruppo di sicurezza da collegare alle istanze, oltre al gruppo di sicurezza dell'istanza creato da Elastic Beanstalk.
    f. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.
    g. Leggere l'avviso, quindi selezionare Confirm (Conferma).
  - Per aggiungere un gruppo di sicurezza tramite un file di configurazione, utilizzare il file di esempio `securitygroup-addexisting.config`.

Quindi, utilizza le proprietà dell'ambiente per far passare le informazioni di connessione al tuo ambiente.

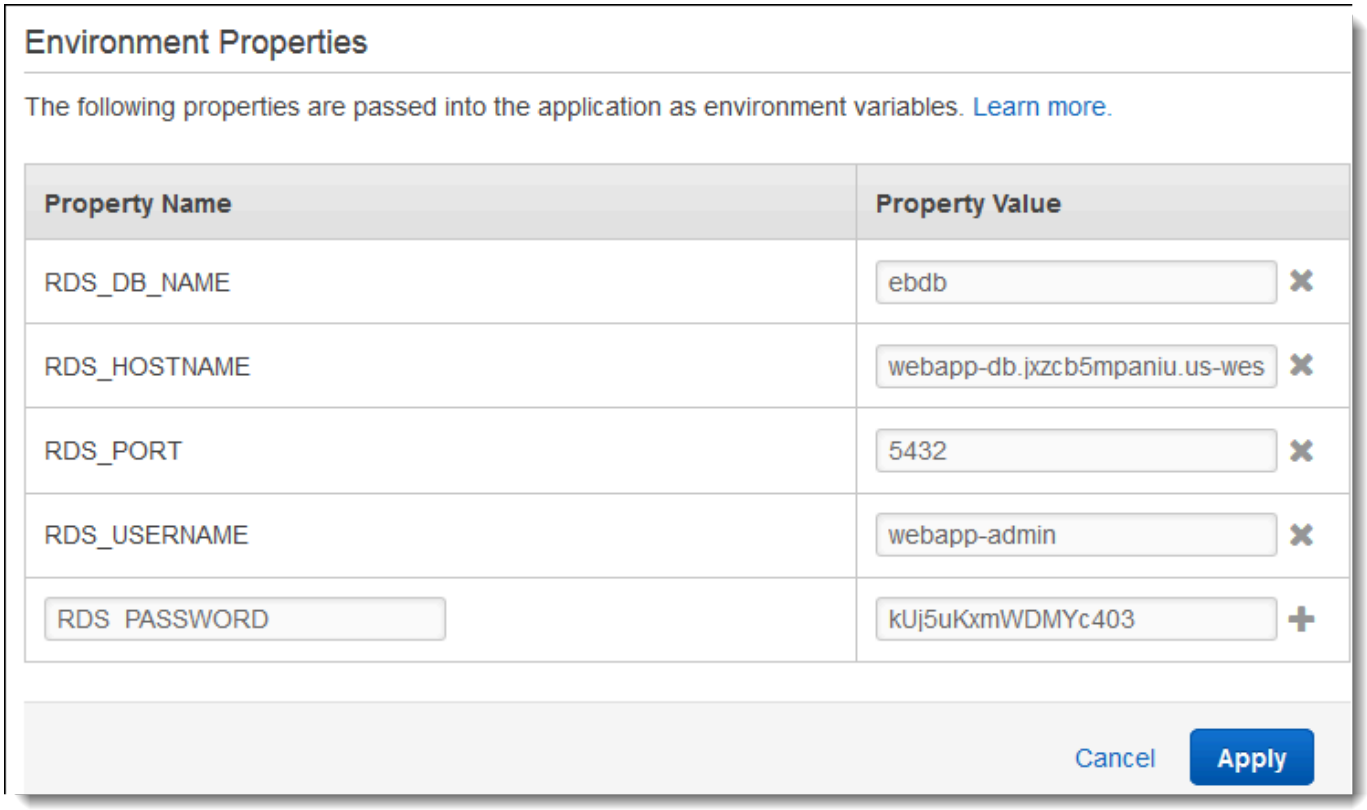
L' WordPress applicazione utilizza un set predefinito di proprietà che corrispondono a quelle configurate da Elastic Beanstalk quando si effettua il provisioning di un database all'interno del proprio ambiente.

Per configurare le proprietà dell'ambiente per un'istanza database Amazon RDS

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).

4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Nella sezione Environment properties (Proprietà ambiente), definire le variabili che l'applicazione deve leggere per formare una stringa di collegamento. Per la compatibilità con gli ambienti che dispongono di un'istanza database RDS integrata, utilizza i seguenti nomi e valori. Tutti i valori, ad eccezione della password, sono disponibili nella console RDS.

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.



6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurazione e distribuzione dell'applicazione

Verifica che la struttura della cartella `wordpress-beanstalk` sia corretta, come illustrato.

```
wordpress-beanstalk$ tree -aL 1
.
### .ebextensions
### index.php
### LICENSE
### license.txt
### readme.html
### README.md
### wp-activate.php
### wp-admin
### wp-blog-header.php
### wp-comments-post.php
### wp-config.php
### wp-config-sample.php
```

```
### wp-content
### wp-cron.php
### wp-includes
### wp-links-opml.php
### wp-load.php
### wp-login.php
### wp-mail.php
### wp-settings.php
### wp-signup.php
### wp-trackback.php
### xmlrpc.php
```

Il file `wp-config.php` personalizzato del repository del progetto usa le variabili di ambiente definite nella fase precedente per configurare la connessione di database. La cartella `.ebextensions` contiene i file di configurazione che consentono di creare risorse aggiuntive nell'ambiente Elastic Beanstalk.

I file di configurazione richiedono una modifica per poter funzionare con il tuo account. Sostituisci i valori segnaposto nei file con quelli appropriati IDs e crea un pacchetto sorgente.

Aggiornamento dei file di configurazione e creazione di un bundle di origine

1. Modifica i file di configurazione nel seguente modo.

   - `.ebextensions/dev.config`— Limita l'accesso all'ambiente per proteggerlo durante il processo di installazione. WordPress Sostituisci l'indirizzo IP segnaposto nella parte superiore del file con l'indirizzo IP pubblico del computer che utilizzerai per accedere al sito Web dell'ambiente e completare l'installazione. WordPress

     > **Note**
     >
     > In base alla rete, potrebbe essere necessario utilizzare un blocco dell'indirizzo IP.

   - `.ebextensions/efs-create.config`— Crea un file system EFS e punti di montaggio in ogni Availability Zone/subnet del tuo VPC. Identifica il tuo VPC e la sottorete predefiniti IDs nella console Amazon VPC.

2. Crea un bundle di origine contenente i file nella cartella di progetto. Il comando seguente crea un bundle di origine denominato `wordpress-beanstalk.zip`.

   ```
   ~/eb-wordpress$ zip ../wordpress-beanstalk.zip -r * .[^.]*
   ```

Carica il pacchetto sorgente su Elastic WordPress Beanstalk per distribuirlo nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Installa WordPress

Per completare l' WordPress installazione

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegliere l'URL dell'ambiente per aprire il sito in un browser. Verrai reindirizzato a una procedura guidata di WordPress installazione perché non hai ancora configurato il sito.
4. Esegui un'installazione standard. Il file `wp-config.php` è già presente nel codice sorgente e configurato per leggere le informazioni di connessione al database dall'ambiente, pertanto il sistema non dovrebbe richiedere la configurazione della connessione.

L'installazione richiede circa un minuto.

Aggiornamento di chiavi e salt

Il file WordPress di configurazione legge `wp-config.php` anche i valori delle chiavi e dei sali dalle proprietà dell'ambiente. Al momento, queste proprietà sono tutte impostate su `test` dal file `wordpress.config` nella cartella `.ebextensions`.

Il salt di hash può essere un qualsiasi valore che soddisfi i requisiti di proprietà dell'ambiente, che però non deve essere memorizzato nel controllo del codice sorgente. Utilizza la console Elastic Beanstalk per impostare queste proprietà direttamente nell'ambiente.

Aggiornamento delle proprietà di ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. In Software, selezionare Edit (Modifica).
5. Modificare le seguenti proprietà per `Environment properties`:
   - `AUTH_KEY`: il valore scelto per `AUTH_KEY`.
   - `SECURE_AUTH_KEY`: il valore scelto per `SECURE_AUTH_KEY`.
   - `LOGGED_IN_KEY`: il valore scelto per `LOGGED_IN_KEY`.
   - `NONCE_KEY`: il valore scelto per `NONCE_KEY`.
   - `AUTH_SALT`: il valore scelto per `AUTH_SALT`.
   - `SECURE_AUTH_SALT`: il valore scelto per `SECURE_AUTH_SALT`.
   - `LOGGED_IN_SALT`: il valore scelto per `LOGGED_IN_SALT`.
   - `NONCE_SALT`: il valore scelto per `NONCE_SALT`.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

> **Note**
>
> Impostando le proprietà direttamente nell'ambiente, si sovrascrivono i valori in `wordpress.config`.

Rimozione delle limitazioni di accesso

Il progetto di esempio include il file di configurazione `loadbalancer-sg.config`. Crea un gruppo di sicurezza e lo assegna al sistema di bilanciamento del carico dell'ambiente, utilizzando l'indirizzo IP configurato in `dev.config`. Limita l'accesso HTTP sulla porta 80 alle connessioni dalla rete. In caso contrario, una parte esterna potrebbe connettersi al tuo sito prima che tu abbia installato WordPress e configurato il tuo account amministratore.

Ora che hai installato WordPress, rimuovi il file di configurazione per aprire il sito al mondo.

Per rimuovere le limitazioni e aggiornare l'ambiente

1. Elimina il file `.ebextensions/loadbalancer-sg.config` dalla directory del progetto.

```
~/wordpress-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Crea un bundle di origine.

```
~/eb-wordpress$ zip ../wordpress-beanstalk-v2.zip -r * .[^.]*
```

Carica il pacchetto sorgente su Elastic WordPress Beanstalk per distribuirlo nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Configurazione del gruppo Auto Scaling

Infine, configura il gruppo Auto Scaling dell'ambiente con un conteggio delle istanze minime più elevato. Esegui almeno due istanze in qualsiasi momento per evitare che il server Web nel tuo ambiente diventi un singolo punto di errore. Ciò consente anche di distribuire le modifiche senza rendere inattivo il tuo sito.

Configurazione del gruppo Auto Scaling dell'ambiente per elevata disponibilità

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).

4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Auto Scaling Group (Gruppo Auto Scaling), impostare Min instances (Istanze min) su **2**.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Per supportare i caricamenti di contenuto in più istanze, il progetto di esempio utilizza Amazon EFS per creare un file system condiviso. Crea un post sul sito e carica i contenuti per memorizzarli sul file system condiviso. Visualizza il post e aggiorna la pagina più volte per accedere a entrambe le istanze e verificare che il file system condiviso sia in fase di elaborazione.

Aggiorna WordPress

Per eseguire l'aggiornamento a una nuova versione di WordPress, esegui il backup del sito e distribuiscilo in un nuovo ambiente.

> **Important**
>
> Non utilizzate la funzionalità di aggiornamento inclusa WordPress né aggiornate i file sorgente per utilizzare una nuova versione. Entrambe queste azioni possono far sì che il post URLs restituisca errori 404 anche se questi sono ancora presenti nel database e nel file system.

Per eseguire l'aggiornamento WordPress

1. Nella console di WordPress amministrazione, utilizza lo strumento di esportazione per esportare i post in un file XML.
2. Distribuisci e installa la nuova versione di WordPress Elastic Beanstalk seguendo gli stessi passaggi utilizzati per installare la versione precedente. Per evitare tempi di inattività, crea un ambiente con la nuova versione.
3. Nella nuova versione, installa lo strumento WordPress Importer nella console di amministrazione e usalo per importare il file XML contenente i tuoi post. Se i post sono stati creati da un utente amministratore in una vecchia versione, assegnali all'utente amministratore del nuovo sito anziché cercare di importare l'utente admin.
4. Se hai distribuito la nuova versione in un ambiente distinto, esegui uno scambio di CNAME per reindirizzare gli utenti dal vecchio sito al nuovo.

Eliminazione

Dopo aver finito di lavorare con il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina tutte le risorse AWS correlate, come istanze EC2 Amazon, istanze di database, sistemi di bilanciamento del carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Inoltre, puoi terminare le risorse di database create al di fuori dell'ambiente Elastic Beanstalk. Quando termini un'istanza database Amazon RDS, puoi acquisire uno snapshot e successivamente ripristinare i dati in un'altra istanza.

Per terminare l'istanza database RDS

1. Apri la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegli l'istanza database.
4. Scegli Actions (Operazioni), quindi Delete (Elimina).
5. Scegliere se creare una snapshot, quindi selezionare Delete (Elimina).

Fasi successive

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

L'applicazione di esempio utilizza i file di configurazione per configurare le impostazioni PHP e creare una tabella nel database, se non esiste già. È inoltre possibile utilizzare un file di configurazione per configurare le impostazioni del gruppo di sicurezza delle istanze durante la creazione dell'ambiente per evitare gli aggiornamenti di configurazione onerosi in termini di tempo. Per ulteriori informazioni, consultare Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Per sviluppo e test, puoi usare la funzionalità di Elastic Beanstalk per l'aggiunta di un'istanza database gestita direttamente nell'ambiente. Per istruzioni sulla configurazione di un database nell'ambiente, consultare Aggiunta di un database all'ambiente Elastic Beanstalk.

Se è necessario un database ad alte prestazioni, prendi in considerazione Amazon Aurora. Amazon Aurora è un motore di database compatibile con MySQL che offre caratteristiche di database commerciali a basso costo. Per connettere l'applicazione a un altro database, ripeti la procedura di configurazione del gruppo di sicurezza e aggiorna le proprietà dell'ambiente relative a RDS.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Distribuzione di un sito Web Drupal a elevata disponibilità con un database Amazon RDS esterno in Elastic Beanstalk

Questo tutorial illustra il processo di avvio di un'istanza DB RDS esterna a. AWS Elastic Beanstalk Successivamente, descrive come configurare un ambiente a elevata disponibilità che esegue un sito Web Drupal per connettersi. Il sito Web utilizza Amazon Elastic File System (Amazon EFS) come storage condiviso per i file caricati. L'esecuzione di un'istanza DB esterna a Elastic Beanstalk disaccoppia il database dal ciclo di vita dell'ambiente e consente di connettersi allo stesso database da più ambienti, sostituire un database con un altro o eseguire una distribuzione senza influire sul database. blue/green

Sections

- Prerequisiti
- Avvio di un'istanza database in Amazon RDS
- Avvio di un ambiente Elastic Beanstalk
- Configurazione delle impostazioni di sicurezza e delle proprietà dell'ambiente
- Configurazione e distribuzione dell'applicazione
- Installa Drupal
- Aggiornamento della configurazione Drupal e rimozione delle restrizioni di accesso

- Configurazione del gruppo Auto Scaling
- Pulizia
- Fasi successive

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata con Windows di Ubuntu e Bash.

Le procedure descritte in questo tutorial per le attività Amazon Relational Database Service (Amazon RDS) presuppongono che si stiano avviando risorse in un Amazon Virtual Private Cloud (Amazon VPC) predefinito. Tutti i nuovi account includono un VPC predefinito in ciascuna regione. Se non hai un VPC predefinito, le procedure variano. Consulta le istruzioni Utilizzo di Elastic Beanstalk con Amazon RDS per le EC2 piattaforme VPC classiche e personalizzate.

L'applicazione di esempio utilizza Amazon EFS. Funziona solo nelle AWS regioni che supportano Amazon EFS. Per informazioni sulle AWS regioni di supporto, consulta Amazon Elastic File System Endpoints and Quotas nel. Riferimenti generali di AWS

Se la piattaforma dell'ambiente Elastic Beanstalk utilizza PHP 7.4 o versioni precedenti, si consiglia di utilizzare Drupal versione 8.9.13 per questo tutorial. Per le piattaforme installate con PHP 8.0 o versioni successive, si consiglia di utilizzare Drupal 9.1.5.

Per maggiori informazioni sulle versioni di Drupal e sulle versioni PHP supportate, consulta Requisiti PHP sul sito Web di Drupal. Le versioni principali consigliate da Drupal sono elencate sul sito Web https://www.drupal. org/project/drupal.

Avvio di un'istanza database in Amazon RDS

Per utilizzare un database esterno con un'applicazione in esecuzione in Elastic Beanstalk, avvia innanzitutto un'istanza database con Amazon RDS. Un'istanza avviata con Amazon RDS è completamente indipendente da Elastic Beanstalk e dagli ambienti Elastic Beanstalk e non verrà terminata o monitorata da Elastic Beanstalk.

Utilizza la console Amazon RDS per avviare un'istanza database Multi-AZ MySQL. La scelta di un'implementazione Multi-AZ garantisce che il database eseguirà il failover e continuerà a essere disponibile se l'istanza database di origine è inutilizzabile.

Per avviare un'istanza database RDS in un VPC predefinito

1. Aprire la console di RDS.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere Create database (Crea database).
4. Scegliere Standard Create (Creazione standard).

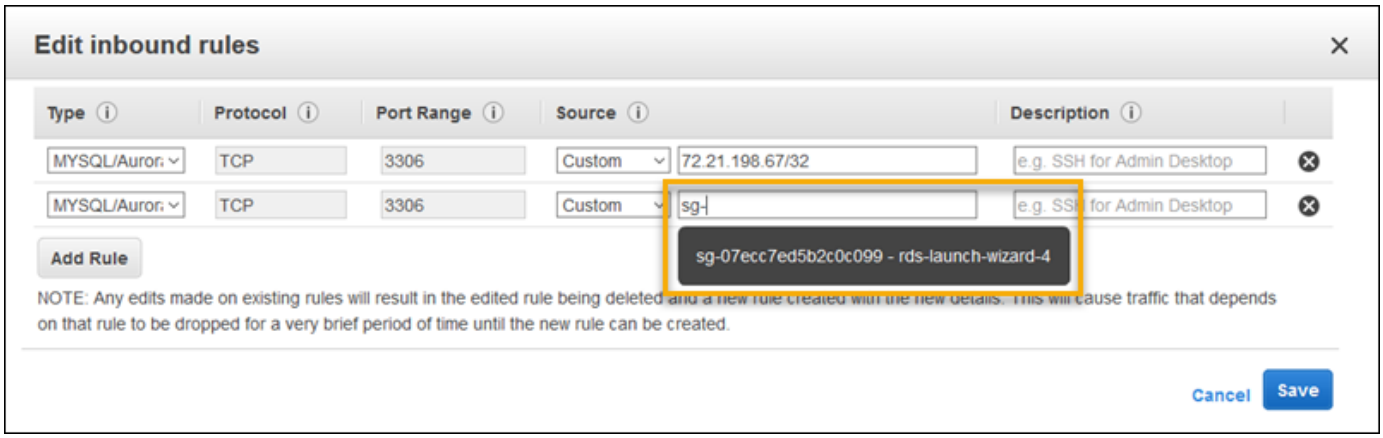
   > **Important**
   >
   > Non selezionare Easy Create (Creazione rapida). Se viene selezionata questa opzione, non è possibile configurare le impostazioni necessarie per avviare questo database RDS.

5. In Additional configuration (Configurazione aggiuntiva), digitare **ebdb** per Initial database name (Nome database iniziale).
6. Controllare le impostazioni di default e regolare queste impostazioni in base alle proprie esigenze specifiche. Prestare particolare attenzione alle seguenti opzioni:
   - DB instance class (Classe di istanze database): scegliere una dimensione di istanza con capacità di memoria e potenza di CPU sufficienti per il tuo carico di lavoro.
   - Implementazione Multi-AZ: per un'elevata disponibilità, impostala su Crea un nodo Replica/Reader Aurora in una AZ diversa.
   - Master username (Nome utente master) e Master password (Password master): nome utente e password del database. Prendi nota di queste impostazioni perché ti serviranno in seguito.
7. Verificare le impostazioni predefinite per le opzioni rimanenti e scegliere Create database (Crea database).

Quindi, modifica il gruppo di sicurezza associato all'istanza database per consentire il traffico in entrata sulla porta appropriata. Questo è lo stesso gruppo di sicurezza che collegherai successivamente all'ambiente Elastic Beanstalk, perciò la regola che aggiungi concederà l'autorizzazione di ingresso ad altre risorse nello stesso gruppo di sicurezza.

Per modificare le regole in entrata del gruppo di sicurezza associato all'istanza RDS

1. Aprire la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Nella sezione Connectivity (Connettività), prendere nota dei valori relativi a Subnets (Sottoreti), Security groups (Gruppi di sicurezza) ed Endpoint mostrati in questa pagina. In questo modo è possibile utilizzare queste informazioni in un secondo momento.
5. In Security (Sicurezza) viene visualizzato il gruppo di sicurezza associato all'istanza database. Apri il link per visualizzare il gruppo di sicurezza nella EC2 console Amazon.
6. Nei dettagli del gruppo di sicurezza, scegliere Inbound (In entrata).
7. Scegli Modifica.
8. Selezionare Add Rule (Aggiungi regola).
9. In Type (Tipo), scegliere il motore di database utilizzato dall'applicazione.
10. In Source (Origine), digitare **`sg-`** per visualizzare un elenco dei gruppi di sicurezza disponibili. Scegliere il gruppo di sicurezza associato al gruppo Auto Scaling utilizzato con l'ambiente Elastic Beanstalk. In questo modo EC2 le istanze Amazon nell'ambiente possono avere accesso al database.



11. Scegli Save (Salva).

La creazione di un'istanza database richiede circa 10 minuti. Nel frattempo, avvia il tuo ambiente Elastic Beanstalk.

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma PHP e accetta le impostazioni predefinite e il codice di esempio. Dopo aver avviato l'ambiente, puoi configurarlo affinché si connetta al database, quindi distribuisci il codice Drupal nell'ambiente.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- EC2 istanza: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per eseguire app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza dell'istanza: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa consente al traffico HTTP proveniente dal sistema di bilanciamento del carico di raggiungere l' EC2 istanza su cui è in esecuzione la tua app web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di

bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.

- Gruppo di sicurezza Load Balancer: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene. L'istanza database di RDS avviata è al di fuori del tuo ambiente, per cui sei responsabile della gestione del suo ciclo di vita.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consultare Utilizzo di Elastic Beanstalk con Amazon S3.

Configurazione delle impostazioni di sicurezza e delle proprietà dell'ambiente

Aggiungi il gruppo di sicurezza dell'istanza database all'ambiente in esecuzione. Con questa procedura Elastic Beanstalk predispone di nuovo tutte le istanze nell'ambiente con il gruppo di sicurezza aggiuntivo collegato.

Per aggiungere un gruppo di sicurezza al tuo ambiente

- Esegui una delle seguenti operazioni:
  - Per aggiungere un gruppo di sicurezza tramite la console Elastic Beanstalk
    a. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
    b. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
    c. Nel pannello di navigazione, seleziona Configuration (Configurazione).
    d. Nella categoria di configurazione Instances (Istanze), scegliere Edit (Modifica).
    e. In Gruppi EC2 di sicurezza, scegli il gruppo di sicurezza da collegare alle istanze, oltre al gruppo di sicurezza dell'istanza creato da Elastic Beanstalk.
    f. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.
    g. Leggere l'avviso, quindi selezionare Confirm (Conferma).
  - Per aggiungere un gruppo di sicurezza tramite un file di configurazione, utilizzare il file di esempio `securitygroup-addexisting.config`.

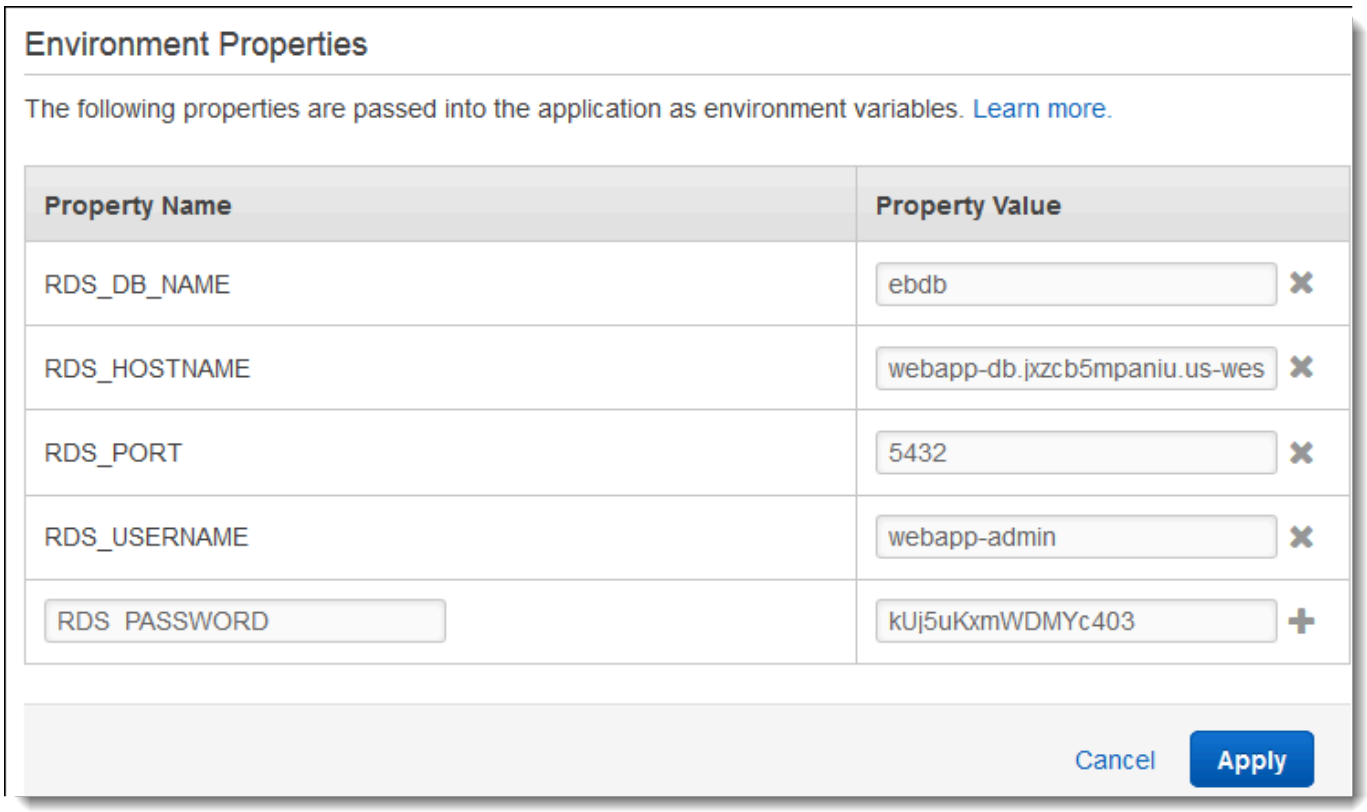
Quindi, utilizza le proprietà dell'ambiente per far passare le informazioni di connessione al tuo ambiente. L'applicazione di esempio utilizza un set di proprietà predefinito che corrisponde a quello configurato da Elastic Beanstalk quando viene effettuato il provisioning di un database all'interno del tuo ambiente.

Per configurare le proprietà dell'ambiente per un'istanza database Amazon RDS

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Nella sezione Environment properties (Proprietà ambiente), definire le variabili che l'applicazione deve leggere per formare una stringa di collegamento. Per la compatibilità con gli ambienti che dispongono di un'istanza database RDS integrata, utilizza i seguenti nomi e valori. Tutti i valori, ad eccezione della password, sono disponibili nella console RDS.

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configura tion (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configura tion (Configurazione) della console Amazon RDS: Master username (Nome utente master).

Nome proprietà	Descrizione	Valore proprietà
RDS_PASSWORD	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.



6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Dopo avere installato Drupal, è necessario connettersi all'istanza con SSH per recuperare alcuni dettagli di configurazione. Assegna una chiave SSH alle istanze dell'ambiente.

Per configurare SSH

1. Se non hai mai creato una coppia di chiavi, apri la pagina delle coppie di chiavi della EC2 console Amazon e segui le istruzioni per crearne una.
2. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
3. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

4. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
5. In Security (Sicurezza), scegliere Edit (Modifica).
6. Per la coppia di EC2 chiavi, scegli la tua coppia di chiavi.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurazione e distribuzione dell'applicazione

Per creare un progetto Drupal per Elastic Beanstalk, scarica il codice sorgente di Drupal e combinalo con i file nel repository aws-samples/ su. eb-php-drupal GitHub

Per creare un progetto Drupal

1. Esegui il comando seguente per scaricare Drupal da www.drupal.org/download. Per saperne di più sui download, consulta il sito web di Drupal.

   Se la piattaforma del tuo ambiente Elastic Beanstalk utilizza PHP 7.4 o versioni precedenti, ti consigliamo di scaricare Drupal versione 8.9.13 per questo tutorial. Per scaricarlo, puoi emettere il comando seguente.

   ```
   ~$ curl https://ftp.drupal.org/files/projects/drupal-8.9.13.tar.gz -o drupal.tar.gz
   ```

   Se la tua piattaforma utilizza PHP 8.0 o versioni successive, ti consigliamo di scaricare Drupal 9.1.5. Puoi utilizzare questo comando per scaricarlo.

   ```
   ~$ curl https://ftp.drupal.org/files/projects/drupal-9.1.5.tar.gz -o drupal.tar.gz
   ```

   Per maggiori informazioni sulle versioni di Drupal e sulle versioni PHP supportate, consulta Requisiti PHP nella documentazione ufficiale di Drupal. Le versioni principali consigliate da Drupal sono riportate sul sito Web di Drupal.
2. Utilizza il comando seguente per scaricare i file di configurazione dal repository di esempio:

   ```
   ~$ wget https://github.com/aws-samples/eb-php-drupal/releases/download/v1.1/eb-php-drupal-v1.zip
   ```
3. Estrai Drupal e modifica il nome della cartella.

   Se è stato scaricato Drupal 8.9.13:

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-8.9.13 drupal-beanstalk
~$ cd drupal-beanstalk
```

Se è stato scaricato Drupal 9.1.5:

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-9.1.5 drupal-beanstalk
~$ cd drupal-beanstalk
```

4. Estrai i file di configurazione dall'installazione di Drupal.

```
~/drupal-beanstalk$ unzip ../eb-php-drupal-v1.zip
 creating: .ebextensions/
 inflating: .ebextensions/dev.config
 inflating: .ebextensions/drupal.config
 inflating: .ebextensions/efs-create.config
 inflating: .ebextensions/efs-filesystem.template
 inflating: .ebextensions/efs-mount.config
 inflating: .ebextensions/loadbalancer-sg.config
 inflating: LICENSE
 inflating: README.md
 inflating: beanstalk-settings.php
```

Verifica che la struttura della cartella `drupal-beanstalk` sia corretta, come illustrato.

```
drupal-beanstalk$ tree -aL 1
.
### autoload.php
### beanstalk-settings.php
### composer.json
### composer.lock
### core
### .csslintrc
### .ebextensions
### .ebextensions
### .editorconfig
### .eslintignore
### .eslintrc.json
### example.gitignore
```

```
### .gitattributes
### .htaccess
### .ht.router.php
### index.php
### LICENSE
### LICENSE.txt
### modules
### profiles
### README.md
### README.txt
### robots.txt
### sites
### themes
### update.php
### vendor
### web.config
```

Il file `beanstalk-settings.php` del repository del progetto usa le variabili di ambiente definite nella fase precedente per configurare la connessione di database. La cartella `.ebextensions` contiene i file di configurazione che consentono di creare risorse aggiuntive nell'ambiente Elastic Beanstalk.

I file di configurazione richiedono una modifica per poter funzionare con il tuo account. Sostituisci i valori segnaposto nei file con quelli appropriati e crea un pacchetto sorgente. IDs

Aggiornamento dei file di configurazione e creazione di un bundle di origine

1. Modifica i file di configurazione nel seguente modo.
   - `.ebextensions/dev.config`: limita l'accesso all'ambiente soltanto al tuo indirizzo IP per garantire la protezione durante il processo di installazione di Drupal. Sostituisci l'indirizzo IP del segnaposto nella parte superiore del file con il tuo indirizzo IP pubblico.
   - `.ebextensions/efs-create.config`: crea un file system EFS e punti di montaggio in ciascuna zona di disponibilità/sottorete nel VPC. Identifica il tuo VPC e la sottorete predefiniti IDs nella console Amazon VPC.
2. Crea un bundle di origine contenente i file nella cartella di progetto. Il comando seguente crea un bundle di origine denominato `drupal-beanstalk.zip`. Esclude i file nella cartella `vendor`, che occupano molto spazio e non sono necessari per la distribuzione dell'applicazione in Elastic Beanstalk.

```
~/eb-drupal$ zip ../drupal-beanstalk.zip -r * .[^.]* -x "vendor/*"
```

Carica il bundle di origine in Elastic Beanstalk per distribuire Drupal nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Installa Drupal

Completamento dell'installazione di Drupal

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Scegliere l'URL dell'ambiente per aprire il sito in un browser. Viene visualizzata una procedura guidata di installazione di Drupal in quanto il sito non è stato ancora configurato.
4. Eseguire un'installazione standard con le seguenti impostazioni per il database:
   - Nome del database: il DB Name (Nome DB) mostrato nella console Amazon RDS.
   - Nome utente e password del database: valori Master Username (Nome utente master) e Master Password (Password master) inseriti al momento della creazione del database.
   - Opzioni avanzate > Host: Endpoint dell'istanza database mostrata nella console Amazon RDS.

L'installazione richiede circa un minuto.

Aggiornamento della configurazione Drupal e rimozione delle restrizioni di accesso

Tramite il processo di installazione di Drupal è stato creato un file denominato `settings.php` nella cartella `sites/default` nell'istanza. Questo file è necessario nel codice sorgente per evitare di reimpostare il sito per le distribuzioni successive, ma il file attualmente contiene segreti di cui non vuoi eseguire il commit nell'origine. Stabilisci una connessione all'istanza dell'applicazione per recuperare le informazioni dal file di impostazioni.

Per connettersi all'istanza dell'applicazione con SSH

1. Apri la pagina delle istanze della EC2 console Amazon.
2. Scegliere l'istanza dell'applicazione. È quella che ha lo stesso nome dell'ambiente Elastic Beanstalk.
3. Scegli Connetti.
4. Seguire le istruzioni per connettere l'istanza a SSH. Il comando è simile a quello riportato di seguito.

```
$ ssh -i ~/.ssh/mykey ec2-user@ec2-00-55-33-222.us-west-2.compute.amazonaws.com
```

Ottieni l'ID della directory di sincronizzazione dall'ultima riga del file di impostazioni.

```
[ec2-user ~]$ tail -n 1 /var/app/current/sites/default/settings.php
$config_directories['sync'] = 'sites/default/files/
config_4ccfX2sPQm79p1mk5IbUq9S_FokcENO4mxyC-L18-4g_xKj_7j9ydn31kDOYOgnzMu071Tvc4Q/
sync';
```

Il file contiene anche la chiave hash corrente dei siti, ma puoi ignorare il valore corrente e usarne uno personalizzato.

Assegna il percorso della directory di sincronizzazione e la chiave hash alle proprietà dell'ambiente. Il file di impostazioni personalizzate nel repository del progetto legge queste proprietà per configurare il sito durante la distribuzione, in aggiunta alle proprietà di connessione di database impostate in precedenza.

Proprietà di configurazione di Drupal

- `SYNC_DIR`: percorso della directory di sincronizzazione.
- `HASH_SALT`: qualsiasi valore di stringa che soddisfi i requisiti delle proprietà dell'ambiente.

Per configurare le variabili di ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri verso il basso fino alle variabili di ambiente Runtime.
6. Seleziona Aggiungi variabile di ambiente.
7. Per Sorgente seleziona Testo normale.

> **Note**
>
> I valori Secrets Manager e SSM Parameter Store nel menu a discesa servono a configurare le variabili di ambiente come segreti per archiviare dati sensibili, come credenziali e chiavi API. Per ulteriori informazioni, consulta Utilizzo di Elastic Gestione dei segreti AWS AWS Systems Manager Beanstalk con e Parameter Store.

8. Immettere le coppie del nome della variabile di ambiente e del valore della variabile di ambiente.
9. Se è necessario aggiungere altre variabili, ripetere i passaggi da 6 a 8.
10. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Infine, il progetto di esempio include un file di configurazione (`loadbalancer-sg.config`) che crea un gruppo di sicurezza e lo assegna al sistema di bilanciamento del carico dell'ambiente, usando l'indirizzo IP configurato in `dev.config` per limitare l'accesso HTTP sulla porta 80 alle connessioni dalla tua rete. In caso contrario, una terza parte potrebbe connettersi al tuo sito prima del completamento dell'installazione di Drupal e della configurazione del tuo account amministratore.

Per aggiornare la configurazione di Drupal e rimuovere le restrizioni di accesso

1. Elimina il file `.ebextensions/loadbalancer-sg.config` dalla directory del progetto.

```
~/drupal-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Copia il file `settings.php` personalizzato nella cartella dei siti.

```
~/drupal-beanstalk$ cp beanstalk-settings.php sites/default/settings.php
```

3. Crea un bundle di origine.

```
~/eb-drupal$ zip ../drupal-beanstalk-v2.zip -r * .[^.]* -x "vendor/*"
```

Carica il bundle di origine in Elastic Beanstalk per distribuire Drupal nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Configurazione del gruppo Auto Scaling

Infine, configura il gruppo Auto Scaling dell'ambiente con un conteggio delle istanze minime più elevato. Esegui almeno due istanze in qualsiasi momento per evitare che i server Web nell'ambiente rappresentino una singola possibilità di errore e per consentire di distribuire le modifiche senza interrompere il funzionamento del servizio.

Configurazione del gruppo Auto Scaling dell'ambiente per elevata disponibilità

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione Capacity (Capacità), scegliere Edit (Modifica).
5. Nella sezione Auto Scaling Group (Gruppo Auto Scaling), impostare Min instances (Istanze min) su **2**.

6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Per supportare i caricamenti di contenuto in più istanze, il progetto di esempio utilizza Amazon Elastic File System per creare un file system condiviso. Crea un post sul sito e carica i contenuti per memorizzarli sul file system condiviso. Visualizza il post e aggiorna la pagina più volte per accedere a entrambe le istanze e verificare che il file system condiviso sia in fase di elaborazione.

Pulizia

Dopo aver finito di utilizzare il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina tutte le risorse AWS correlate, come istanze EC2 Amazon, istanze di database, sistemi di bilanciamento del carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Inoltre, puoi terminare le risorse di database create al di fuori dell'ambiente Elastic Beanstalk. Quando termini un'istanza database Amazon RDS, puoi acquisire uno snapshot e successivamente ripristinare i dati in un'altra istanza.

Per terminare l'istanza database RDS

1. Apri la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegli l'istanza database.
4. Scegli Actions (Operazioni), quindi Delete (Elimina).
5. Scegliere se creare una snapshot, quindi selezionare Delete (Elimina).

Fasi successive

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

L'applicazione di esempio utilizza i file di configurazione per configurare le impostazioni PHP e creare una tabella nel database, se non esiste già. È inoltre possibile utilizzare un file di configurazione per configurare le impostazioni del gruppo di sicurezza delle istanze durante la creazione dell'ambiente per evitare gli aggiornamenti di configurazione onerosi in termini di tempo. Per ulteriori informazioni, consultare Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Per sviluppo e test, puoi usare la funzionalità di Elastic Beanstalk per l'aggiunta di un'istanza database gestita direttamente nell'ambiente. Per istruzioni sulla configurazione di un database nell'ambiente, consultare Aggiunta di un database all'ambiente Elastic Beanstalk.

Se è necessario un database ad alte prestazioni, prendi in considerazione Amazon Aurora. Amazon Aurora è un motore di database compatibile con MySQL che offre caratteristiche di database commerciali a basso costo. Per connettere l'applicazione a un altro database, ripeti la procedura di configurazione del gruppo di sicurezza e aggiorna le proprietà dell'ambiente relative a RDS.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Distribuzione di applicazioni Python con Elastic Beanstalk

Questo capitolo fornisce istruzioni per configurare e distribuire l'applicazione web Python su. AWS Elastic Beanstalk Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni Web Python utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, è possibile continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure è possibile utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Segui i passaggi indicati nelle step-by-step istruzioni QuickStart per Python per creare e distribuire un'applicazione web Python Hello World con l'EB CLI.

Argomenti

- QuickStart: distribuisci un'applicazione Python su Elastic Beanstalk
- Configurazione dell'ambiente di sviluppo Python per Elastic Beanstalk
- Utilizzo della piattaforma Python di Elastic Beanstalk
- Implementazione di un'applicazione Flask in Elastic Beanstalk
- Distribuzione di un'applicazione Django in Elastic Beanstalk
- Aggiungere un'istanza DB Amazon RDS al tuo ambiente Python Elastic Beanstalk
- Risorse e strumenti Python

QuickStart: distribuisci un'applicazione Python su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione Python e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account

- Prerequisiti
- Fase 1: Creare un'applicazione Python
- Fase 2: Esegui l'applicazione localmente
- Passaggio 3: distribuisci la tua applicazione Python con l'EB CLI
- Fase 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti a un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Framework Python e Flask

Conferma di avere una versione di Python funzionante con `pip` installata eseguendo i seguenti comandi.

```
~$ python3 --version
Python 3.N.N
>~$ python3 -m pip --version
pip X.Y.Z from ... (python 3.N.N)
```

Se uno dei comandi precedenti restituisce «Python was not found «, esegui i seguenti comandi che use `python` invece di. `python3` La configurazione degli alias e dei collegamenti simbolici può variare in base al sistema operativo e alle singole personalizzazioni, pertanto il `python3` comando potrebbe non funzionare sul computer.

```
~$ python --version
Python 3.N.N
>~$ python -m pip --version
pip X.Y.Z from ... (python 3.N.N)
```

Se non hai Python installato sul tuo computer locale, puoi scaricarlo dalla pagina dei download di Python sul sito Web di Python. Per un elenco delle versioni del linguaggio Python supportate da Elastic Beanstalk, consulta Piattaforme Python supportate nella guida Piattaforme.AWS Elastic Beanstalk Il sito Web per i download di Python fornisce un collegamento alla Python Developer's Guide, dove troverai le istruzioni di installazione e configurazione.

> **Note**
>
> Il `pip` pacchetto Python è incluso di default in Python 3.4 o versioni successive.

Se l'output indica che disponi di una versione supportata di Python, ma non lo è`pip`, consulta la pagina di installazione sul sito Web pip.pypa.io. Fornisce indicazioni per installare pip in un ambiente Python che non lo dispone.

Conferma se Flask è installato eseguendo il seguente comando:

```
~$ pip list | grep Flask
```

Se Flask non è installato, puoi installarlo con il seguente comando:

```
~$ pip install Flask
```

Fase 1: Creare un'applicazione Python

Crea una directory di progetto.

```
~$ mkdir eb-python
~$ cd eb-python
```

Crea un esempio di «Hello Elastic Beanstalk!» Applicazione Python che distribuirai utilizzando Elastic Beanstalk.

Crea un file di testo denominato `application.py` nella directory appena creata con i seguenti contenuti.

Example~/eb- .py python/application

```
from flask import Flask
application = Flask(__name__)

@application.route('/')
def hello_elastic_beanstalk():
        return 'Hello Elastic Beanstalk!'
```

Crea un file di testo denominato `requirements.txt` con la riga seguente. Questo file contiene i `pip` pacchetti necessari per l'esecuzione dell'applicazione.

Example~/eb- .txt python/requirements

```
Flask
```

Fase 2: Esegui l'applicazione localmente

Esegui il comando seguente per eseguire l'applicazione localmente.

```
~/eb-python$ export FLASK_APP=application.py && flask run --port 5000
```

Dovresti vedere un output simile al seguente

```
Serving Flask app 'application.py'
Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a
 production WSGI server instead.
Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [01/Jan/1970 00:00:00] "GET / HTTP/1.1" 200 -
```

Accedi a `http://localhost:5000` nel tuo browser web. Il browser Web dovrebbe visualizzare «Hello Elastic Beanstalk!».

Passaggio 3: distribuisci la tua applicazione Python con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire la tua applicazione Python

1. Inizializza il repository della CLI EB con il comando eb init.

   ```
   ~/eb-python$ eb init -p python-3.9 python-tutorial --region us-east-2
   ```

   Questo comando crea un'applicazione denominata `python-tutorial` e configura il repository locale per creare ambienti con la versione della piattaforma Python fornita.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

   ```
   ~/eb-python$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   2) [ Create new KeyPair ]
   ```

Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

```
~/eb-python$ eb create python-env
```

Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-python$ eb open
```

Complimenti! Hai distribuito un'applicazione Python con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 5: rimozione

Puoi chiudere il tuo ambiente quando finisci di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-python$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.

- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. `subdomain region`.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

> **Prova altri tutorial**
>
> Se desideri provare altri tutorial con diverse applicazioni di esempio, consulta i seguenti tutorial:

- Implementazione di un'applicazione Flask in Elastic Beanstalk
- Distribuzione di un'applicazione Django in Elastic Beanstalk

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire applicazioni Python localmente, vedi. Configurazione dell'ambiente di sviluppo Python per Elastic Beanstalk

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Configurazione dell'ambiente di sviluppo Python per Elastic Beanstalk

Questo argomento fornisce istruzioni per configurare un ambiente di sviluppo Python per testare l'applicazione localmente prima di distribuirla. AWS Elastic Beanstalk Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows è possibile installare il sottosistema Windows per Linux per ottenere una versione integrata per Windows di Ubuntu e Bash.

Sections

- Prerequisiti
- Uso di un ambiente virtuale

- Configurazione di un progetto Python per Elastic Beanstalk

Prerequisiti

L'elenco seguente fornisce i prerequisiti comuni per lavorare con Elastic Beanstalk e le tue applicazioni Python:

- Linguaggio Python: installa la versione del linguaggio Python inclusa nella versione della piattaforma Elastic Beanstalk Python scelta. Per un elenco delle nostre versioni del linguaggio Python supportate, consulta Piattaforme Python supportate nella guida Piattaforme.AWS Elastic Beanstalk Se non hai già configurato Python sulla tua macchina di sviluppo, consulta la pagina dei download di Python sul sito Web di Python.
- **`pip`**utility — L'**`pip`**utilità è il programma di installazione dei pacchetti di Python. Installa ed elenca le dipendenze per il tuo progetto, in modo che Elastic Beanstalk sappia come configurare l'ambiente dell'applicazione. Per ulteriori informazioni in merito*`pip`*, consulta la pagina pip sul sito Web pip.pypa.io.
- (Facoltativo) Elastic Beanstalk Command Line Interface (EB CLI): l'EB CLI può impacchettare l'applicazione con i file di distribuzione necessari. Può anche creare un ambiente Elastic Beanstalk e distribuirvi l'applicazione. Puoi anche effettuare distribuzioni tramite la console Elastic Beanstalk, quindi l'EB CLI non è strettamente necessaria.
- Un'**`SSH`**installazione funzionante: puoi connetterti alle istanze in esecuzione con il protocollo SSH per esaminare o eseguire il debug di una distribuzione.
- **`virtualenv`**pacchetto: questo `virtualenv` strumento crea un ambiente di sviluppo e test per l'applicazione. Elastic Beanstalk può replicare questo ambiente senza installare pacchetti aggiuntivi non richiesti dall'applicazione. Per ulteriori informazioni, consulta il sito web virtualenv. Dopo aver installato Python, puoi installare il `virtualenv` pacchetto con il seguente comando:

```
$ pip install virtualenv
```

Uso di un ambiente virtuale

Dopo aver installato i prerequisiti, imposta un ambiente virtuale con `virtualenv` per l'installazione delle dipendenze dell'applicazione. Utilizzando un ambiente virtuale, è possibile distinguere esattamente quali pacchetti sono necessari all'applicazione in modo che i pacchetti richiesti vengano installati sulle EC2 istanze che eseguono l'applicazione.

Per configurare un ambiente virtuale

1. Apri una finestra a riga di comando e digita:

```
$ virtualenv /tmp/eb_python_app
```

   *eb_python_app*Sostituitelo con un nome che abbia senso per l'applicazione (usare il nome dell'applicazione è una buona idea). Il comando `virtualenv` crea un ambiente virtuale per l'utente nella directory specificata e stampa i risultati delle operazioni:

```
Running virtualenv with interpreter /usr/bin/python
New python executable in /tmp/eb_python_app/bin/python3.12
Also creating executable in /tmp/eb_python_app/bin/python
Installing setuptools, pip...done.
```

2. Quando l'ambiente virtuale è pronto, avvialo eseguendo lo script `activate` che si trova nella directory `bin` dell'ambiente. Ad esempio, per avviare l'ambiente eb_python_app creato nella fase precedente, digita:

```
$ source /tmp/eb_python_app/bin/activate
```

   L'ambiente virtuale stamperà il proprio nome, (ad esempio: `(eb_python_app)`) all'inizio di ogni prompt dei comandi, per ricordarti che sei in un ambiente Python virtuale.

3. Per interrompere l'utilizzo dell'ambiente virtuale e tornare all'interprete Python predefinito del sistema con tutte le relative librerie installate, eseguire il comando `deactivate`.

```
(eb_python_app) $ deactivate
```

> **Note**
>
> Una volta creato, è possibile riavviare l'ambiente virtuale in qualsiasi momento eseguendo di nuovo lo script `activate`.

Configurazione di un progetto Python per Elastic Beanstalk

È possibile utilizzare l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) per preparare le applicazioni Python alla distribuzione con Elastic Beanstalk.

Per configurare un'applicazione Python per la distribuzione con Elastic Beanstalk

1. Dall'interno del tuo ambiente virtuale, torna alla parte superiore della struttura ad albero della directory del progetto (`python_eb_app`) e digita:

   ```
   pip freeze >requirements.txt
   ```

   Questo comando copia i nomi e le versioni dei pacchetti installati nell'ambiente virtuale su `requirements.txt`. Ad esempio, se il pacchetto PyYAML versione 3.11 è installato nell'ambiente virtuale, il file conterrà la riga:

   ```
   PyYAML==3.11
   ```

   Ciò consente a Elastic Beanstalk di replicare l'ambiente Python dell'applicazione utilizzando gli stessi pacchetti e le stesse versioni che hai utilizzato per sviluppare e testare l'applicazione.
2. Configura il repository della CLI EB con il comando eb init. Segui le istruzioni per scegliere regione, piattaforma e altre opzioni.

Per impostazione predefinita, Elastic Beanstalk cerca un file denominato `application.py` per avviare l'applicazione. Se questo non esiste nel progetto Python creato, sono necessari alcuni aggiustamenti dell'ambiente della tua applicazione. Dovrai anche impostare le variabili di ambiente per consentire il caricamento dei moduli della tua applicazione. Per ulteriori informazioni, consultare Utilizzo della piattaforma Python di Elastic Beanstalk.

Utilizzo della piattaforma Python di Elastic Beanstalk

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Python su Elastic Beanstalk.

AWS Elastic Beanstalk supporta una serie di rami di piattaforma per diverse versioni del linguaggio di programmazione Python. Vedi Python nel documento AWS Elastic Beanstalk Piattaforme per un elenco completo.

Le applicazioni web Python possono essere eseguite dietro un server proxy con WSGI. Elastic Beanstalk fornisce Gunicorn come server WSGI predefinito.

È possibile aggiungere un `Procfile` al bundle di origine per specificare e configurare il server WSGI per l'applicazione. Per informazioni dettagliate, vedi the section called “Procfile”.

È possibile utilizzare i file `Pipfile.lock` e `Pipfile` creati da Pipenv per specificare le dipendenze dei pacchetti Python e altri requisiti. Per informazioni dettagliate sulla definizione delle dipendenze, consulta the section called “Specifica delle dipendenze”.

Elastic Beanstalk offre opzioni di configurazione che puoi utilizzare per personalizzare il software che viene eseguito EC2 sulle istanze nell'ambiente Elastic Beanstalk. Puoi configurare le variabili di ambiente necessarie per la tua applicazione, accedere alla rotazione dei log su Amazon S3 e mappare le cartelle nel codice sorgente dell'applicazione che contengono i file statici ai percorsi serviti dal server proxy.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Per i pacchetti Python disponibili in `pip`, puoi includere un file di requisiti nella radice del codice sorgente dell'applicazione. Elastic Beanstalk installa tutti i pacchetti delle dipendenze specificati in un file dei requisiti durante la distribuzione. Per informazioni dettagliate, vedi the section called “Specifica delle dipendenze”.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Configurazione dell'ambiente Python

Le impostazioni della piattaforma Python ti consentono di ottimizzare il comportamento delle tue istanze Amazon. EC2 Puoi modificare la configurazione dell'istanza Amazon dell'ambiente Elastic Beanstalk utilizzando EC2 la console Elastic Beanstalk.

Usa la console Elastic Beanstalk per configurare AWS X-Ray le impostazioni dei processi Python, abilitare, abilitare la rotazione dei log su Amazon S3 e configurare le variabili che l'applicazione può leggere dall'ambiente.

Per configurare l'ambiente Python nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Monitoring (Monitoraggio), scegli Edit (Modifica).

Impostazioni Python

- Proxy server (Server proxy): il server proxy da utilizzare nelle istanze dell'ambiente. Per impostazione predefinita, viene utilizzato nginx.
- WSGI Path (Percorso WSGI): nome del file principale dell'applicazione o il relativo percorso. Ad esempio, `application.py` o `django/wsgi.py`.
- NumProcesses— Il numero di processi da eseguire su ciascuna istanza dell'applicazione.
- NumThreads— Il numero di thread da eseguire in ogni processo.

AWS X-Ray impostazioni

- Demone X-Ray: esegue il demone per elaborare i AWS X-Ray dati di traccia da. SDK AWS X-Ray per Python

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Abilita la rotazione dei file di registro su Amazon S3: specifica se i file di log per le EC2 istanze Amazon dell'applicazione vengono copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Per impostazione predefinita, il server proxy in un ambiente Python serve qualsiasi file in una cartella denominata `static` nel percorso `/static`. Ad esempio, se il codice sorgente dell'applicazione contiene un file con nome `logo.png` in una cartella chiamata `static`, il server proxy lo servirà agli utenti su *`subdomain`*`.elasticbeanstalk.com/static/logo.png`. È possibile configurare mappature aggiuntive come indicato in questa sezione.

Proprietà dell'ambiente

Puoi utilizzare le proprietà dell'ambiente per fornire informazioni alla tua applicazione e configurare variabili di ambiente. Ad esempio, puoi creare una proprietà di ambiente denominata `CONNECTION_STRING` che specifica una stringa di connessione che la tua applicazione può utilizzare per connettersi a un database.

All'interno dell'ambiente Python in esecuzione in Elastic Beanstalk, questi valori sono accessibili tramite il dizionario `os.environ` di Python. Per ulteriori informazioni, consulta http://docs.python.org/library/os.html.

Puoi utilizzare un codice simile al seguente per accedere alle chiavi e ai parametri:

```
import os
endpoint = os.environ['API_ENDPOINT']
```

Le proprietà dell'ambiente sono inoltre in grado di fornire informazioni a un framework. Ad esempio, è possibile creare una proprietà denominata `DJANGO_SETTINGS_MODULE` per configurare Django in modo che utilizzi un modulo specifico di impostazioni. A seconda dell'ambiente, il valore potrebbe essere `development.settings`, `production.settings` e così via.

Per ulteriori informazioni, consultare Variabili di ambiente e altre impostazioni software.

Spazio dei nomi di configurazione di Python

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

La piattaforma Python definisce le opzioni negli spazi dei nomi `aws:elasticbeanstalk:environment:proxy`, `aws:elasticbeanstalk:environment:proxy:staticfiles` e `aws:elasticbeanstalk:container:python`.

Il seguente file di configurazione di esempio specifica le impostazioni delle opzioni di configurazione per creare una proprietà di ambiente denominata `DJANGO_SETTINGS_MODULE`, scegliere il server proxy Apache, specificare due opzioni di file statici che mappano una directory denominata `statichtml` sul percorso `/html` e una directory denominata `staticimages` sul percorso `/images` e specificare le impostazioni aggiuntive nello spazio dei nomi `aws:elasticbeanstalk:container:python`. Questo spazio dei nomi contiene le opzioni che consentono di specificare il percorso dello script WSGI nel tuo codice sorgente e il numero di thread e processi in esecuzione in WSGI.

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    DJANGO_SETTINGS_MODULE: production.settings
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
  aws:elasticbeanstalk:container:python:
    WSGIPath: ebdjango.wsgi:application
    NumProcesses: 3
    NumThreads: 20
```

> **Note**
>
> - Se stai utilizzando una versione di piattaforma Python in AMI Amazon Linux (precedente ad Amazon Linux 2), sostituisci il valore `WSGIPath` con `ebdjango/wsgi.py`. Il valore nell'esempio funziona con il server Gunicorn WSGI, che non è supportato nelle versioni della piattaforma AMI Amazon Linux.
> - Inoltre, queste versioni della piattaforma precedenti utilizzano uno spazio dei nomi diverso per la configurazione dei file statici, ovvero `aws:elasticbeanstalk:container:python:staticfiles`. Ha gli stessi nomi di opzioni e semantica dello spazio dei nomi dei file statici standard.

I file di configurazione, supportano inoltre diverse chiavi per modificare ulteriormente il software delle istanze del tuo ambiente. Questo esempio utilizza la chiave pacchetti per installare Memcached con `yum` e i comandi dei container per eseguire comandi che consentono di configurare il server durante la distribuzione:

```
packages:
  yum:
    libmemcached-devel: '0.31'

container_commands:
  collectstatic:
    command: "django-admin.py collectstatic --noinput"
  01syncdb:
    command: "django-admin.py syncdb --noinput"
    leader_only: true
  02migrate:
    command: "django-admin.py migrate"
    leader_only: true
  03wsgipass:
    command: 'echo "WSGIPassAuthorization On" >> ../wsgi.conf'
  99customize:
    command: "scripts/customize.sh"
```

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

L'`python3`eseguibile

La versione dell'`python3`eseguibile disponibile nelle EC2 istanze negli ambienti Elastic Beanstalk Python non corrisponderà sempre alla stessa versione di Python utilizzata dalla piattaforma. Ad esempio, sulla piattaforma Python 3.12 AL2 023, punta `/usr/bin/python3` a Python 3.9. Questo perché Python 3.9 è il sistema Python su 023. AL2 Per ulteriori informazioni, consulta Python in AL2 023 nella Guida per l'utente di Amazon Linux 2023. È possibile accedere a un eseguibile corrispondente alla versione di Python utilizzata dalla piattaforma in una posizione con versione (ad esempio`/usr/bin/python3.12`) o nella `bin` directory dell'ambiente virtuale dell'applicazione (ad esempio). `/var/app/venv/staging-LQM1lest/bin/python3` La piattaforma utilizza l'eseguibile Python corretto che corrisponde al ramo della piattaforma.

Configurazione del server WSGI con un profilo su Elastic Beanstalk

È possibile aggiungere un `Procfile` al bundle di origine per specificare e configurare il server WSGI per l'applicazione. È possibile specificare comandi di avvio ed esecuzione personalizzati in. `Procfile`

Quando utilizzi un `Procfile`, vengono sovrascritte le opzioni dello spazio dei nomi `aws:elasticbeanstalk:container:python` impostate utilizzando i file di configurazione.

Nell'esempio seguente viene utilizzato un `Procfile` per specificare uWSGI come server e configurarlo.

Example Procfile

```
web: uwsgi --http :8000 --wsgi-file application.py --master --processes 4 --threads 2
```

Nell'esempio seguente viene utilizzato un `Procfile` per configurare Gunicorn, il server WSGI predefinito.

Example Procfile

```
web: gunicorn --bind :8000 --workers 3 --threads 2 project.wsgi:application
```

> **Note**
>
> - Se configuri un server WSGI diverso da Gunicorn, assicurati di specificarlo anche come dipendenza dell'applicazione, in modo che venga installato nelle istanze dell'ambiente.

Per informazioni dettagliate sulla specifica delle dipendenze, consulta the section called "Specifica delle dipendenze".

- La porta predefinita per il server WSGI è 8000. Se specifichi un numero di porta diverso nel comando `Procfile`, imposta anche la proprietà `PORT` dell'ambiente su questo numero di porta.

Specificazione delle dipendenze utilizzando un file dei requisiti su Elastic Beanstalk

Questo argomento descrive come configurare l'applicazione per installare altri pacchetti Python richiesti. Un'applicazione Python tipica dispone di dipendenze da altri pacchetti Python di terze parti. Con la piattaforma Elastic Beanstalk Python, hai diversi modi per specificare i pacchetti Python da cui dipende la tua applicazione.

Uso di **`pip`** e **`requirements.txt`**

Lo strumento standard per l'installazione di pacchetti Python è `pip`. Dispone di una funzionalità che consente di specificare tutti i pacchetti di cui hai bisogno (nonché le loro versioni) in un singolo file di requisiti. Per ulteriori informazioni sul file di requisiti, consulta Formato del file di requisiti sul sito Web della documentazione pip.

Crea un file denominato `requirements.txt` e inseriscilo nella directory di livello superiore del tuo bundle di origine. Di seguito è riportato un file `requirements.txt` di esempio per Django.

```
Django==2.2
mysqlclient==2.0.3
```

Nel tuo ambiente di sviluppo, puoi utilizzare il comando `pip freeze` per generare il tuo file di requisiti.

```
~/my-app$ pip freeze > requirements.txt
```

Per verificare che il tuo file di requisiti contenga solo i pacchetti che sono effettivamente utilizzati dall'applicazione, usa un ambiente virtuale che contenga solo quei pacchetti installati. Al di fuori di un ambiente virtuale, l'output di `pip freeze` includerà tutti i pacchetti `pip` installati sul computer di sviluppo, inclusi quelli forniti con il sistema operativo.

> **Note**
>
> Nelle versioni della piattaforma Python in AMI Amazon Linux, Elastic Beanstalk non supporta Pipenv o Pipfiles a livello nativo. Se utilizzi Pipenv per gestire le dipendenze dell'applicazione, esegui il comando seguente per generare un file `requirements.txt`.
>
> ```
> ~/my-app$ pipenv lock -r > requirements.txt
> ```
>
> Per ulteriori informazioni, consulta Generating a requirements.txt nella documentazione di Pipenv.

Uso di Pipenv e **`Pipfile`**

Pipenv è uno strumento di creazione pacchetti Python moderno. Combina l'installazione del pacchetto con la creazione e la gestione di un file delle dipendenze e di un virtualenv per l'applicazione. Per ulteriori informazioni, consulta Pipenv: Python Dev Workflow for Humans.

Pipenv mantiene due file:

- `Pipfile`: questo file contiene vari tipi di dipendenze e requisiti.
- `Pipfile.lock`: questo file contiene uno snapshot della versione che abilita le build deterministiche.

Puoi crearli nell'ambiente di sviluppo e includerli nella directory di primo livello del bundle di origine che implementi in Elastic Beanstalk. Per ulteriori informazioni su questi due file, consulta Esempio di Pipfile e PipFile.lock.

Nell'esempio seguente viene utilizzato Pipenv per installare Django e il framework REST Django. Questi comandi creano i file `Pipfile` e `Pipfile.lock`.

```
~/my-app$ pipenv install django
~/my-app$ pipenv install djangorestframework
```

Priorità

Se includi più di un file dei requisiti tra quelli descritti qui, Elastic Beanstalk ne utilizzerà uno solo. L'elenco seguente mostra la precedenza, in ordine decrescente.

1. `requirements.txt`
2. `Pipfile.lock`
3. `Pipfile`

> **Note**
>
> A partire dalla versione della piattaforma Amazon Linux 2 del 7 marzo 2023, se fornisci più di uno di questi file, Elastic Beanstalk emetterà un messaggio sulla console indicando quale dei file di dipendenza è stato utilizzato durante una implementazione.

La seguente procedura descrive la logica seguita da Elastic Beanstalk per installare le dipendenze durante l'implementazione di un'istanza.

- Se è presente un file `requirements.txt`, usiamo il comando `pip install -r requirements.txt`.
- A partire dalla versione della piattaforma Amazon Linux 2 del 7 marzo 2023, se non è presente alcun file `requirements.txt` ma ce n'è uno `Pipfile.lock`, utilizziamo il comando `pipenv sync`. Prima di quella versione, usavamo `pipenv install --ignore-pipfile`.
- Se non c'è né un file `requirements.txt` né un `Pipfile.lock`, ma è presente un `Pipfile`, usiamo il comando `pipenv install --skip-lock`.
- Se non viene trovato nessuno dei tre file dei requisiti, non installiamo alcuna dipendenza dall'applicazione.

Implementazione di un'applicazione Flask in Elastic Beanstalk

Questo tutorial illustra il processo di generazione di un'applicazione Flask e la sua distribuzione in un ambiente. AWS Elastic Beanstalk Flask è un framework di applicazione Web open source per Python.

In questo tutorial, verranno eseguite le operazioni seguenti:

- Configurazione di un ambiente virtuale Python con Flask
- Creazione di un'applicazione Flask
- Distribuzione del sito con la CLI EB
- Pulizia

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata con Windows di Ubuntu e Bash.

In questo tutorial utilizziamo Python 3.11 e la corrispondente versione della piattaforma Elastic Beanstalk. Installa Python seguendo la procedura descritta in Configurazione dell'ambiente di sviluppo Python per Elastic Beanstalk.

Il framework Flask verrà installato nell'ambito di questo tutorial.

Questo tutorial utilizza anche l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Configurazione di un ambiente virtuale Python con Flask

Crea una directory di progetto e l'ambiente virtuale per la tua applicazione, quindi installa Flask.

Per configurare l'ambiente di progetto

1. Crea una directory di progetto.

```
~$ mkdir eb-flask
~$ cd eb-flask
```

2. Crea e attiva un ambiente virtuale denominato `virt`:

```
~/eb-flask$ virtualenv virt
~$ source virt/bin/activate
(virt) ~/eb-flask$
```

Visualizzerai `(virt)` anteposto al prompt dei comandi, a indicare che sei in un ambiente virtuale. Utilizza l'ambiente virtuale per le restanti operazioni di questo tutorial.

3. Installa Flask con `pip install`:

```
(virt)~/eb-flask$ pip install flask==3.0.3
```

4. Visualizza le librerie installate con `pip freeze`:

```
(virt)~/eb-flask$ pip freeze
blinker==1.8.2
click==8.1.7
Flask==3.0.3
importlib_metadata==8.5.0
itsdangerous==2.2.0
Jinja2==3.1.4
MarkupSafe==2.1.5
Werkzeug==3.0.4
zipp==3.20.2
```

Questo comando elenca tutti i pacchetti installati nell'ambiente virtuale. Poiché ti trovi in un ambiente virtuale, i pacchetti installati a livello globale come la CLI EB non vengono visualizzati.

5. Salva l'output da `pip freeze` in un file denominato `requirements.txt`.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

Questo file indica a Elastic Beanstalk di installare le librerie durante la distribuzione. Per ulteriori informazioni, consulta Specificazione delle dipendenze utilizzando un file dei requisiti su Elastic Beanstalk.

Creazione di un'applicazione Flask

Successivamente, crea un'applicazione che distribuirai utilizzando Elastic Beanstalk. Creeremo un servizio web «Hello World». RESTful

In questa directory, crea un nuovo file di testo denominato `application.py` con il seguente contenuto:

Example~/eb-flask/application.py

```
from flask import Flask

# print a nice greeting.
def say_hello(username = "World"):
    return '<p>Hello %s!</p>\n' % username

# some bits of text for the page.
header_text = '''
    <html>\n<head> <title>EB Flask Test</title> </head>\n<body>'''
instructions = '''
    <p><em>Hint</em>: This is a RESTful web service! Append a username
    to the URL (for example: <code>/Thelonious</code>) to say hello to
    someone specific.</p>\n'''
home_link = '<p><a href="/">Back</a></p>\n'
footer_text = '</body>\n</html>'

# EB looks for an 'application' callable by default.
application = Flask(__name__)

# add a rule for the index page.
application.add_url_rule('/', 'index', (lambda: header_text +
    say_hello() + instructions + footer_text))

# add a rule when the page is accessed with a name appended to the site
# URL.
application.add_url_rule('/<username>', 'hello', (lambda username:
    header_text + say_hello(username) + home_link + footer_text))

# run the app.
if __name__ == "__main__":
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production app.
    application.debug = True
```

```
    application.run()
```

Questo esempio stampa un saluto personalizzato che varia in base al percorso utilizzato per accedere al servizio.

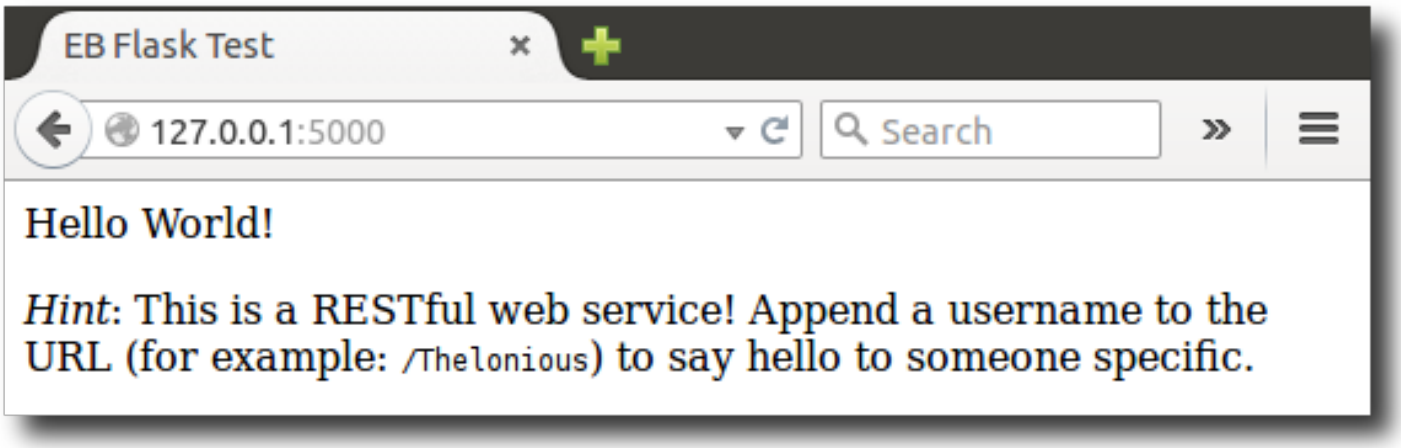
> **Note**
>
> Se aggiungi `application.debug = True` prima di eseguire l'applicazione, l'output di debug viene attivato in caso di problemi. È una buona prassi per lo sviluppo, ma è necessario rimuovere le istruzioni di debug dal codice di produzione in quanto l'output di debug può rivelare aspetti interni dell'applicazione.

Utilizzando `application.py` come nome di file e fornendo un oggetto `application` richiamabile (l'oggetto Flask, in questo caso), Elastic Beanstalk trova facilmente il codice dell'applicazione.

Esegui `application.py` con Python:

```
(virt) ~/eb-flask$ python application.py
 * Serving Flask app "application" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 313-155-123
```

Apri `http://127.0.0.1:5000/` nel tuo browser Web. L'applicazione dovrebbe essere in esecuzione e visualizzare la pagina di indice:



Seleziona il log del server per visualizzare l'output dalla tua richiesta. Puoi arrestare il server Web e tornare al tuo ambiente virtuale premendo CTRL+C.

Se invece hai ricevuto l'output di debug, correggi gli errori e accertati che l'applicazione venga eseguita localmente prima di configurarla per Elastic Beanstalk.

Distribuzione del sito con la CLI EB

Hai aggiunto tutto ciò di cui hai bisogno per distribuire l'applicazione su Elastic Beanstalk. La directory del tuo progetto dovrebbe ora apparire così:

```
~/eb-flask/
|-- virt
|-- application.py
`-- requirements.txt
```

La cartella `virt`, tuttavia, non è necessaria affinché l'applicazione sia eseguita su Elastic Beanstalk. Quando si esegue la distribuzione, Elastic Beanstalk crea un nuovo ambiente virtuale sulle istanze server e installa le librerie elencate in `requirements.txt`. Per ridurre al minimo le dimensioni del bundle di origine caricato durante la distribuzione, aggiungi un file .ebignore, che indica alla CLI EB di escludere la cartella `virt`.

Example~/eb-flask/.ebignore

```
virt
```

Quindi, potrai creare l'ambiente applicativo e distribuire l'applicazione configurata con Elastic Beanstalk.

Creazione di un ambiente e distribuzione dell'applicazione Flask

1. Inizializza il repository della CLI EB con il comando eb init:

```
~/eb-flask$ eb init -p python-3.11 flask-tutorial --region us-east-2
Application flask-tutorial has been created.
```

   Questo comando crea una nuova applicazione denominata `flask-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma Python 3.11.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi di default e connetterti all'istanza EC2 che esegue l'applicazione con SSH:

```
~/eb-flask$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

   Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci l'applicazione nello stesso con eb create:

```
~/eb-flask$ eb create flask-env
```

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- Istanza EC2: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per l'esecuzione di app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.

- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal sistema di bilanciamento del carico di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza del sistema di bilanciamento del carico: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

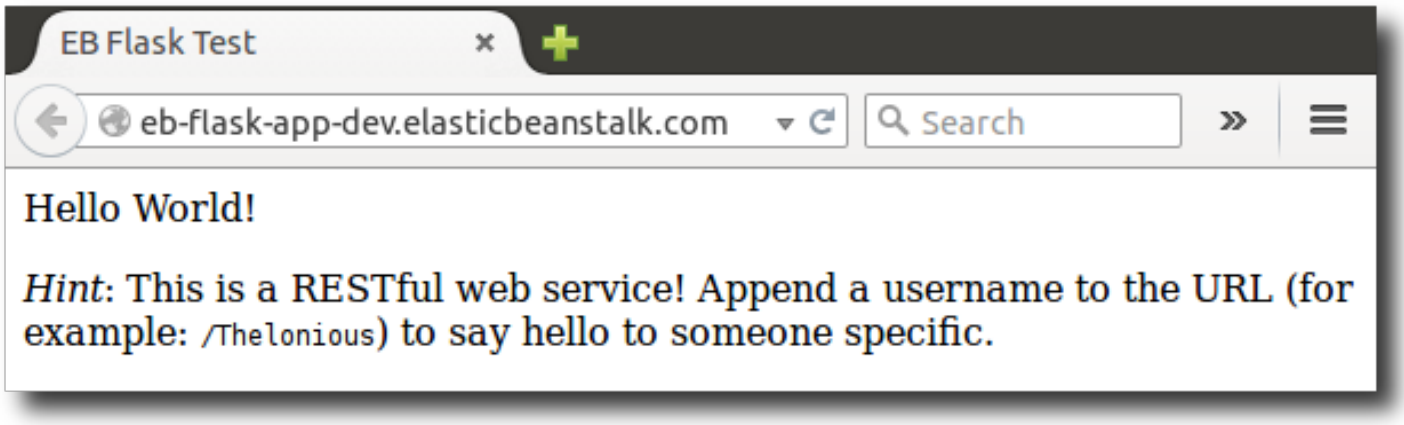
Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon S3.

Al termine del processo di creazione dell'ambiente, apri il sito Web con eb open:

```
~/eb-flask$ eb open
```

Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione. Dovresti visualizzare lo stesso sito Web Flask che hai creato e testato in locale.



Se non vedi l'applicazione in esecuzione, oppure viene visualizzato un messaggio di errore, consulta la sezione relativa alla risoluzione di problemi relativi alle distribuzioni per informazioni su come determinare la causa dell'errore.

Se vedi l'applicazione in esecuzione, significa che hai distribuito la tua prima applicazione Flask con Elastic Beanstalk.

Pulizia

Dopo aver finito di lavorare con il codice demo, puoi chiudere il tuo ambiente. Elastic Beanstalk elimina AWS tutte le risorse correlate, come istanze Amazon EC2, istanze di database, sistemi di bilanciamentodel carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

In alternativa, con la CLI EB:

```
~/eb-flask$ eb terminate flask-env
```

Fasi successive

Per ulteriori informazioni su Flask, visita flask.pocoo.org.

Se vuoi provare un altro framework Web Python, consulta Distribuzione di un'applicazione Django in Elastic Beanstalk.

Distribuzione di un'applicazione Django in Elastic Beanstalk

In questo tutorial viene descritta la distribuzione di un sito Web Django predefinito e autogenerato in un ambiente AWS Elastic Beanstalk in cui è in esecuzione Python. Questo tutorial illustra come ospitare un'app Web Python nel cloud utilizzando un ambiente Elastic Beanstalk.

In questo tutorial, verranno eseguite le operazioni seguenti:

- Configurazione di un ambiente virtuale Python e installazione di Django
- Creazione di un progetto Django
- Configurazione dell'applicazione Django per Elastic Beanstalk
- Distribuzione del sito con la CLI EB
- Aggiornamento di un'applicazione
- Eliminazione

Prerequisiti

Per seguire questo tutorial, è necessario aver prima installato tutti i prerequisiti comuni per Python, inclusi i seguenti pacchetti:

- Python 3.7 o versioni successive
- `pip`
- `virtualenv`
- `awsebcli`

Il framework Django viene installato nell'ambito di questo tutorial.

> **Note**
>
> La creazione di ambienti con CLI EB richiede un ruolo del servizio. Puoi creare un ruolo di servizio tramite la creazione di un ambiente nella console Elastic Beanstalk. Se non disponi di un ruolo del servizio, CLI EB tenta di crearne uno durante l'esecuzione di `eb create`.

Configurazione di un ambiente virtuale Python e installazione di Django

Crea un ambiente virtuale con `virtualenv` e utilizzalo per installare Django e le relative dipendenze. Utilizzando un ambiente virtuale, puoi sapere esattamente di quali pacchetti ha bisogno la tua applicazione, in modo che i pacchetti richiesti vengano installati sulle EC2 istanze Amazon su cui è in esecuzione l'applicazione.

La procedura seguente illustra i comandi da immettere per i sistemi basati su Unix e Windows, mostrati su schede separate.

Per configurare un ambiente virtuale

1. Crea un ambiente virtuale denominato `eb-virt`.

   Unix-based systems

   ```
   ~$ virtualenv ~/eb-virt
   ```

Windows

```
C:\> virtualenv %HOMEPATH%\eb-virt
```

2. Attiva l'ambiente virtuale.

   Unix-based systems

```
~$ source ~/eb-virt/bin/activate
(eb-virt) ~$
```

   Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate
(eb-virt) C:\>
```

   Visualizzerai `(eb-virt)` anteposto al prompt dei comandi, a indicare che sei in un ambiente virtuale.

> **Note**
>
> Il resto di queste istruzioni mostra il prompt dei comandi di Linux nella directory principale ~$. Su Windows questo è`C:\Users\`*USERNAME*>, *USERNAME* dov'è il tuo nome di accesso a Windows.

3. Utilizza `pip` per installare Django.

```
(eb-virt)~$ pip install django==2.2
```

> **Note**
>
> La versione Django che installi deve essere compatibile con la versione Python della configurazione Elastic Beanstalk Python scelta per distribuire l'applicazione. Per informazioni sulla distribuzione, consultare ??? in questo argomento.
> Per ulteriori informazioni sulle versioni correnti della piattaforma Python, consulta Python nel documento Piattaforme AWS Elastic Beanstalk .

Per la compatibilità delle versioni Django con Python, consulta la domanda relativa a quale versione di Python è necessario usare con Django?

4. Per verificare che Django sia installato, immetti quanto segue.

```
(eb-virt)~$ pip freeze
Django==2.2
...
```

Questo comando elenca tutti i pacchetti installati nell'ambiente virtuale. Successivamente, potrai utilizzare l'output di questo comando per configurare il progetto da utilizzare con Elastic Beanstalk.

Creazione di un progetto Django

Ora è possibile creare un progetto Django ed eseguirlo sul tuo computer, utilizzando l'ambiente virtuale.

> **Note**
>
> Questo tutorial utilizza SQLite, che è un motore di database incluso in Python. Il database viene distribuito con i tuoi file di progetto. Per gli ambienti di produzione, ti consigliamo di utilizzare Amazon Relational Database Service (Amazon RDS) e di separarlo dall'ambiente. Per ulteriori informazioni, consulta Aggiungere un'istanza DB Amazon RDS al tuo ambiente Python Elastic Beanstalk.

Per generare un'applicazione Django

1. Attiva il tuo ambiente virtuale.

   Unix-based systems

```
~$ source ~/eb-virt/bin/activate
(eb-virt) ~$
```

Windows

```
C:\>%HOMEPATH%\eb-virt\Scripts\activate
(eb-virt) C:\>
```

Visualizzerai il prefisso `(eb-virt)` anteposto al prompt dei comandi, a indicare che sei in un ambiente virtuale.

> **Note**
>
> Il resto di queste istruzioni mostra il prompt dei comandi di Linux ~$ nella directory principale e la directory principale Linux `~/`. Su Windows questi sono`C:\Users\`*USERNAME*>, *USERNAME* dov'è il tuo nome di accesso a Windows.

2. Utilizzare il comando `django-admin startproject` per creare un nuovo progetto Django denominato `ebdjango`:

```
(eb-virt)~$ django-admin startproject ebdjango
```

Questo comando crea un sito Django standard denominato ebdjango con la seguente struttura di directory:

```
~/ebdjango
  |-- ebdjango
  |   |-- __init__.py
  |   |-- settings.py
  |   |-- urls.py
  |   `-- wsgi.py
  `-- manage.py
```

3. Eseguire il sito Django in locale con `manage.py runserver`:

```
(eb-virt) ~$ cd ebdjango
```

```
(eb-virt) ~/ebdjango$ python manage.py runserver
```

4. Aprire `http://127.0.0.1:8000/` in un browser Web per visualizzare il sito:

5. Seleziona il log del server per visualizzare l'output dalla tua richiesta. Per arrestare il server Web e tornare all'ambiente virtuale, premere Ctrl+C.

```
Django version 2.2, using settings 'ebdjango.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[07/Sep/2018 20:14:09] "GET / HTTP/1.1" 200 16348
Ctrl+C
```

Configurazione dell'applicazione Django per Elastic Beanstalk

Ora che disponi di un sito con tecnologia Django sul computer locale, puoi configurarlo per la distribuzione con Elastic Beanstalk.

Per impostazione predefinita, Elastic Beanstalk cerca un file denominato `application.py` per avviare l'applicazione. Poiché questo non esiste nel progetto Django creato, è necessario apportare alcune modifiche all'ambiente dell'applicazione. Dovrai anche impostare le variabili di ambiente per consentire il caricamento dei moduli dell'applicazione.

Per configurare il sito per Elastic Beanstalk

1. Attiva il tuo ambiente virtuale.

   Unix-based systems

   ```
   ~/ebdjango$ source ~/eb-virt/bin/activate
   ```

   Windows

   ```
   C:\Users\USERNAME\ebdjango>%HOMEPATH%\eb-virt\Scripts\activate
   ```

2. Esegui `pip freeze`, quindi salva l'output in un file con nome `requirements.txt`:

   ```
   (eb-virt) ~/ebdjango$ pip freeze > requirements.txt
   ```

   Elastic `requirements.txt` Beanstalk lo utilizza per determinare quale pacchetto installare EC2 sulle istanze che eseguono l'applicazione.

3. Crea una directory denominata `.ebextensions`.

```
(eb-virt) ~/ebdjango$ mkdir .ebextensions
```

4. Nella directory `.ebextensions`, aggiungi un file di configurazione denominato `django.config` con il seguente testo:

   Example~/ebdjango/.ebextensions/django.config

```
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: ebdjango.wsgi:application
```

   L'impostazione `WSGIPath` specifica la posizione dello script WSGI utilizzato da Elastic Beanstalk per avviare l'applicazione.

> **Note**
>
> Se stai utilizzando una versione di piattaforma Python in AMI Amazon Linux (precedente ad Amazon Linux 2), sostituisci il valore `WSGIPath` con `ebdjango/wsgi.py`. Il valore nell'esempio funziona con il server Gunicorn WSGI, che non è supportato nelle versioni della piattaforma AMI Amazon Linux.

5. Disattiva l'ambiente virtuale con il comando `deactivate`.

```
(eb-virt) ~/ebdjango$ deactivate
```

   Riattiva il tuo ambiente virtuale ogni volta che hai bisogno di aggiungere pacchetti alla tua applicazione o di eseguire l'applicazione in locale.

Distribuzione del sito con la CLI EB

Hai aggiunto tutto ciò di cui hai bisogno per distribuire l'applicazione su Elastic Beanstalk. La directory del progetto dovrebbe ora apparire così.

```
~/ebdjango/
|-- .ebextensions
|   `-- django.config
|-- ebdjango
|   |-- __init__.py
```

```
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
|-- db.sqlite3
|-- manage.py
`-- requirements.txt
```

Quindi, potrai creare l'ambiente applicativo e distribuire l'applicazione configurata con Elastic Beanstalk.

Subito dopo la distribuzione, potrai modificare la configurazione di Django per aggiungere il nome di dominio che Elastic Beanstalk ha assegnato all'applicazione per il valore `ALLOWED_HOSTS` di Django. Quindi ridistribuisci l'applicazione. Si tratta di un requisito di sicurezza di Django, progettato per evitare attacchi alle intestazioni di `Host` HTTP. Per ulteriori informazioni, consulta la pagina relativa alla convalida dell'intestazione host.

Creazione di un ambiente e distribuzione dell'applicazione Django

> **Note**
>
> Il tutorial utilizza la CLI EB come meccanismo di distribuzione, ma è anche possibile utilizzare la console Elastic Beanstalk per distribuire un file ZIP con i contenuti del progetto.

1. Inizializza il repository della CLI EB con il comando eb init.

   ```
   ~/ebdjango$ eb init -p python-3.7 django-tutorial
   Application django-tutorial has been created.
   ```

   Questo comando crea un'applicazione denominata `django-tutorial`. Configura inoltre il repository locale per creare ambienti con la versione più recente della piattaforma Python 3.7.

2. (Facoltativo) Esegui eb init nuovamente per configurare una coppia di key pair predefinita in modo da poter utilizzare SSH per connetterti all' EC2 istanza che esegue l'applicazione.

   ```
   ~/ebdjango$ eb init
   Do you want to set up SSH for your instances?
   (y/n): y
   Select a keypair.
   1) my-keypair
   ```

```
2) [ Create new KeyPair ]
```

Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create:

```
~/ebdjango$ eb create django-env
```

> **Note**
>
> Se visualizzi un messaggio di errore "service role required" (ruolo del servizio richiesto), esegui `eb create` interattivamente (senza specificare un nome dell'ambiente) ed CLI EB crea il ruolo per te.

Questo comando crea un ambiente Elastic Beanstalk con bilanciamento del carico denominato `django-env`. Per la creazione dell'ambiente sono necessari circa 5 minuti. Quando Elastic Beanstalk crea le risorse necessarie per l'esecuzione dell'applicazione, genera messaggi informativi che la CLI EB inoltra al terminale.

4. Al termine del processo di creazione dell'ambiente, puoi individuare il nome di dominio del tuo nuovo ambiente eseguendo eb status:

```
~/ebdjango$ eb status
Environment details for: django-env
  Application name: django-tutorial
  ...
  CNAME: eb-django-app-dev.elasticbeanstalk.com
  ...
```

Il nome di dominio del tuo ambiente è il valore della proprietà `CNAME`.

5. Aprire il file `settings.py` nella directory `ebdjango`. Individuare l'impostazione `ALLOWED_HOSTS` e quindi aggiungere al valore dell'impostazione il nome dominio dell'applicazione che hai trovato nella fase precedente. Se non trovi l'impostazione nel file, aggiungila a una nuova riga.

```
...
```

```
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

6. Salva il file, quindi distribuisci l'applicazione eseguendo eb deploy. Quando esegui eb deploy, la CLI EB integra i contenuti della tua directory di progetto in un bundle e li distribuisce al tuo ambiente.

```
~/ebdjango$ eb deploy
```

> **Note**
>
> Se usi Git con il progetto, consulta Utilizzo dell'interfaccia a riga di comando EB con Git.

7. Al termine del processo di aggiornamento dell'ambiente, apri il sito Web con eb open.

```
~/ebdjango$ eb open
```

Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione. Dovresti visualizzare lo stesso sito Web Django che hai creato e testato in locale.

Se non vedi l'applicazione in esecuzione, oppure viene visualizzato un messaggio di errore, consulta la sezione relativa alla risoluzione di problemi relativi alle distribuzioni per informazioni su come determinare la causa dell'errore.

Se vedi l'applicazione in esecuzione, significa che hai distribuito la tua prima applicazione Django con Elastic Beanstalk.

Aggiornamento di un'applicazione

Ora che disponi di un'applicazione in esecuzione su Elastic Beanstalk, puoi aggiornare e ridistribuire l'applicazione o la sua configurazione ed Elastic Beanstalk si occuperà dell'aggiornamento delle istanze e dell'avvio della nuova versione dell'applicazione.

Per questo esempio, abilitiamo la console di amministrazione di Django e configuriamo qualche altra impostazione.

Modifica delle impostazioni del sito

Per impostazione predefinita, il sito Web Django usa il fuso orario UTC per visualizzare l'ora. Per modificare questa opzione, è necessario specificare un fuso orario in `settings.py`.

Per modificare il fuso orario del sito

1. Modifica l'impostazione `TIME_ZONE` in `settings.py`.

   Example~/ .py ebdjango/ebdjango/settings

   ```
   ...
   # Internationalization
   LANGUAGE_CODE = 'en-us'
   TIME_ZONE = 'US/Pacific'
   USE_I18N = True
   USE_L10N = True
   USE_TZ = True
   ```

   Per un elenco dei fusi orari, visita questa pagina.

2. Distribuisci l'applicazione nell'ambiente Elastic Beanstalk.

   ```
   ~/ebdjango/$ eb deploy
   ```

Creazione di un amministratore del sito

È possibile creare un amministratore del sito per l'applicazione Django per accedere alla console di amministrazione direttamente dal sito Web. I dettagli di accesso dell'amministratore vengono archiviati in modo sicuro nell'immagine locale del database inclusa nel progetto predefinito generato da Django.

Per creare un amministratore del sito

1. Inizializza il database locale dell'applicazione Django:

   ```
   (eb-virt) ~/ebdjango$ python manage.py migrate
   Operations to perform:
     Apply all migrations: admin, auth, contenttypes, sessions
   Running migrations:
     Applying contenttypes.0001_initial... OK
     Applying auth.0001_initial... OK
     Applying admin.0001_initial... OK
     Applying admin.0002_logentry_remove_auto_add... OK
     Applying admin.0003_logentry_add_action_flag_choices... OK
     Applying contenttypes.0002_remove_content_type_name... OK
   ```

```
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

2. Esegui `manage.py createsuperuser` per creare un amministratore:

```
(eb-virt) ~/ebdjango$ python manage.py createsuperuser
Username: admin
Email address: me@mydomain.com
Password: ********
Password (again): ********
Superuser created successfully.
```

3. Per indicare a Django dove archiviare i file statici, definisci `STATIC_ROOT` in `settings.py`:

   Example~/ .py ebdjango/ebdjango/settings

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/
STATIC_URL = '/static/'
STATIC_ROOT = 'static'
```

4. Esegui `manage.py collectstatic` per popolare la `static` directory con risorse statiche (JavaScriptCSS e immagini) per il sito di amministrazione.

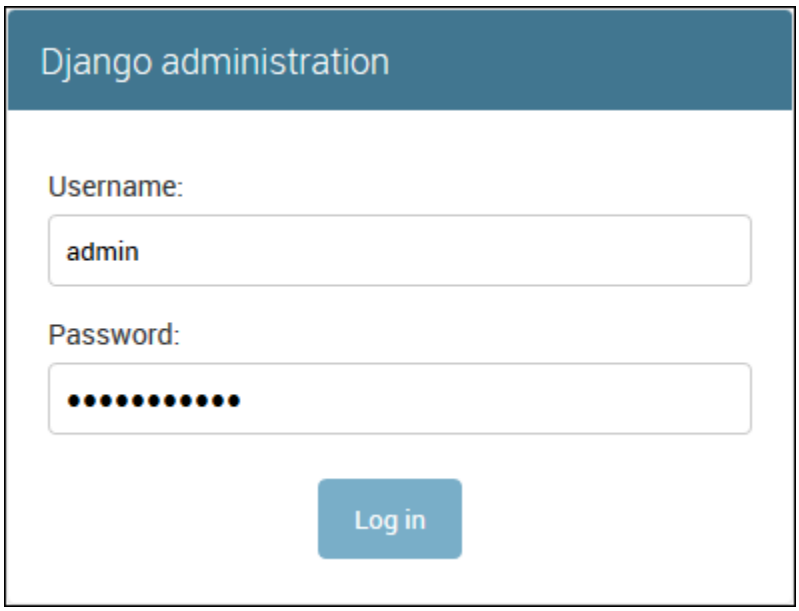
```
(eb-virt) ~/ebdjango$ python manage.py collectstatic
119 static files copied to ~/ebdjango/static
```

5. Distribuisci l'applicazione.

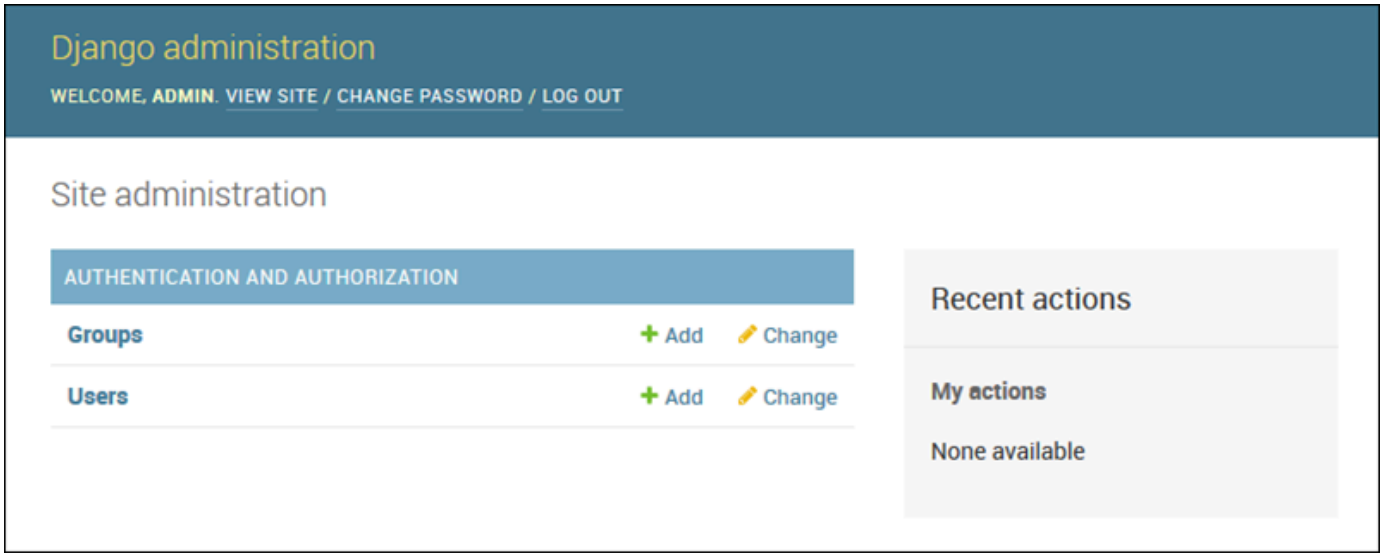
```
~/ebdjango$ eb deploy
```

6. Visualizza la console di amministrazione aprendo il sito nel tuo browser e aggiungendo `/admin/` all'URL del sito, ad esempio:

```
http://djang-env.p33kq46sfh.us-west-2.elasticbeanstalk.com/admin/
```



7. Effettua il login con il nome utente e la password che hai configurato nella fase 2.



È possibile utilizzare una procedura simile a quella locale updating/testing seguita daeb deploy. Elastic Beanstalk si occupa di attività quali l'aggiornamento dei server in tempo reale, consentendoti di concentrarti sullo sviluppo delle applicazioni invece che sull'amministrazione dei server.

Aggiunta di un file di configurazione di migrazione del database

È possibile aggiungere comandi al tuo script `.ebextensions` che vengono eseguiti quando il tuo sito è aggiornato. In questo modo è possibile generare automaticamente migrazioni di database.

Per aggiungere una fase di migrazione durante la migrazione dell'applicazione

1. Crea un nuovo file di configurazione denominato `db-migrate.config` con il seguente contenuto.

   Example~/ebdjango/.ebextensions/db-migrate.config

   ```
   container_commands:
     01_migrate:
       command: "source /var/app/venv/*/bin/activate && python3 manage.py migrate"
       leader_only: true
   option_settings:
     aws:elasticbeanstalk:application:environment:
       DJANGO_SETTINGS_MODULE: ebdjango.settings
   ```

   Questo file di configurazione attiva l'ambiente virtuale del server ed esegue il comando `manage.py migrate` durante il processo di distribuzione, prima di avviare l'applicazione. Poiché è in esecuzione prima di avviare l'applicazione, è necessario anche configurare la variabile di ambiente `DJANGO_SETTINGS_MODULE` in modo esplicito (in genere `wsgi.py` si occupa di questo per te durante l'avvio). Se specifichi `leader_only: true` nel comando sei sicuro che viene eseguito solo una volta durante la distribuzione a più istanze.

2. Distribuisci l'applicazione.

   ```
   ~/ebdjango$ eb deploy
   ```

Eliminazione

Per risparmiare ore di istanza e altre AWS risorse tra le sessioni di sviluppo, chiudi l'ambiente Elastic Beanstalk con. eb terminate

```
~/ebdjango$ eb terminate django-env
```

Questo comando termina l'ambiente e tutte le AWS risorse in esecuzione al suo interno. Tuttavia, non elimina l'applicazione, di conseguenza puoi sempre creare altri ambienti con la stessa configurazione eseguendo di nuovo eb create.

Se hai terminato con l'applicazione di esempio, puoi anche eliminare la cartella del progetto e l'ambiente virtuale:

```
~$ rm -rf ~/eb-virt
~$ rm -rf ~/ebdjango
```

Fasi successive

Per ulteriori informazioni su Django, tra cui un tutorial dettagliato, consulta la documentazione ufficiale.

Se vuoi provare un altro framework Web Python, consulta Implementazione di un'applicazione Flask in Elastic Beanstalk.

Aggiungere un'istanza DB Amazon RDS al tuo ambiente Python Elastic Beanstalk

Questo argomento fornisce istruzioni per creare un Amazon RDS utilizzando la console Elastic Beanstalk. Puoi utilizzare un'istanza database Amazon Relational Database Service (Amazon RDS) per archiviare i dati raccolti e modificati dall'applicazione. Il database può essere accoppiato all'ambiente e gestito da Elastic Beanstalk oppure può essere creato e gestito esternamente da un altro servizio. In queste istruzioni il database è accoppiato all'ambiente e gestito da Elastic Beanstalk. Per ulteriori informazioni sull'integrazione di un Amazon RDS con Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Sections

- Aggiunta di un'istanza database all'ambiente
- Download di un driver
- Connessione a un database

Aggiunta di un'istanza database all'ambiente

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'aggiunta di un'istanza database richiede circa 10 minuti. Quando l'aggiornamento dell'ambiente è completo, il nome host dell'istanza database e altre informazioni di connessione sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente:

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master

Nome proprietà	Descrizione	Valore proprietà
		username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Per ulteriori informazioni sulla configurazione di un'istanza di database accoppiata a un ambiente Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Download di un driver

Aggiungi il driver di database al file di requisiti del progetto.

Example requirements.txt: Django con MySQL

```
Django==2.2
mysqlclient==2.0.3
```

Pacchetti di driver comuni per Python

- MySQL – `mysqlclient`
- PostgreSQL – `psycopg2`
- Oracle: `cx_Oracle`
- SQL Server – `adodbapi`

Per ulteriori informazioni, vedere Python DatabaseInterfaces e Django 2.2 - database supportati.

Connessione a un database

Elastic Beanstalk fornisce informazioni di connessione per le istanze database collegate nelle proprietà dell'ambiente. Utilizza `os.environ['VARIABLE']` per leggere le proprietà e configurare una connessione di database.

Example File di impostazioni Django: dizionario dei DATABASE

```
import os
```

```
if 'RDS_HOSTNAME' in os.environ:
    DATABASES = {
        'default': {
            'ENGINE': 'django.db.backends.mysql',
            'NAME': os.environ['RDS_DB_NAME'],
            'USER': os.environ['RDS_USERNAME'],
            'PASSWORD': os.environ['RDS_PASSWORD'],
            'HOST': os.environ['RDS_HOSTNAME'],
            'PORT': os.environ['RDS_PORT'],
        }
    }
```

Risorse e strumenti Python

Vi sono diverse risorse in cui puoi trovare ulteriori informazioni durante lo sviluppo delle tue applicazioni Python:

Risorsa	Description
AWS SDK per Python (Boto3) su GitHub	Installa Boto3 proveniente da. GitHub
AWS SDK per Python (Boto3) home page	La AWS SDK per Python (Boto3) home page.
Python Developer Center	Punto di riferimento per il codice di esempio, la documentazione, gli strumenti e risorse aggiuntive.

Distribuzione di applicazioni Ruby con Elastic Beanstalk

Questo capitolo fornisce istruzioni per configurare e distribuire l'applicazione web Ruby su. AWS Elastic Beanstalk Elastic Beanstalk semplifica la distribuzione, la gestione e la scalabilità delle applicazioni web Ruby utilizzando Amazon Web Services.

Puoi distribuire la tua applicazione in pochi minuti utilizzando l'Elastic Beanstalk Command Line Interface (EB CLI) o utilizzando la console Elastic Beanstalk. Dopo aver distribuito l'applicazione Elastic Beanstalk, è possibile continuare a utilizzare l'EB CLI per gestire l'applicazione e l'ambiente, oppure è possibile utilizzare la console Elastic Beanstalk o il. AWS CLI APIs

Questo capitolo fornisce anche step-by-step istruzioni per distribuire un'applicazione di esempio su Elastic Beanstalk utilizzando l'EB CLI e quindi aggiornare l'applicazione per utilizzare i framework di applicazioni web Rails e Sinatra.

Gli argomenti di questo capitolo presuppongono una certa conoscenza degli ambienti Elastic Beanstalk. Se è la prima volta che utilizzi Elastic Beanstalk, segui il tutorial sulle nozioni di base per apprendere i concetti di base.

Argomenti

- Configurazione dell'ambiente di sviluppo Ruby per Elastic Beanstalk
- Utilizzo della piattaforma Ruby di Elastic Beanstalk
- Distribuzione di un'applicazione Rails su Elastic Beanstalk
- Distribuzione di un'applicazione Sinatra su Elastic Beanstalk
- Aggiungere un'istanza Amazon RDS DB all'ambiente Ruby Elastic Beanstalk

Configurazione dell'ambiente di sviluppo Ruby per Elastic Beanstalk

Questo capitolo fornisce istruzioni per configurare un ambiente di sviluppo Ruby per testare l'applicazione localmente prima di distribuirla. AWS Elastic Beanstalk Fa inoltre riferimento a siti Web che forniscono istruzioni di installazione per strumenti utili.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows è possibile installare il sottosistema Windows per Linux per ottenere una versione integrata per Windows di Ubuntu e Bash.

Sections

- Installazione di Ruby
- Installazione dell' AWS SDK per Ruby
- Installazione di un IDE o un editor di testo

Installazione di Ruby

Installa GCC se non hai un compilatore C. In Ubuntu usa `apt`.

```
~$ sudo apt install gcc
```

Su Amazon Linux, utilizza `yum`.

```
~$ sudo yum install gcc
```

Installa RVM per gestire le installazioni in linguaggio Ruby nel tuo computer. Usa i comandi indicati in rvm.io per ottenere le chiavi di progetto ed eseguire lo script di installazione.

```
~$ gpg2 --recv-keys key1 key2
~$ curl -sSL https://get.rvm.io | bash -s stable
```

Questo script installa RVM in una cartella denominata `.rvm` nella directory utente e modifica il profilo shell per caricare uno script di configurazione ogni volta che apri un nuovo terminale. Carica lo script manualmente per iniziare.

```
~$ source ~/.rvm/scripts/rvm
```

Usa `rvm get head` per ottenere la versione più recente.

```
~$ rvm get head
```

Visualizza le versioni disponibili di Ruby.

```
~$ rvm list known
```

Dai un'occhiata a Ruby nel documento relativo alle Piattaforme AWS Elastic Beanstalk per trovare la versione più recente di Ruby disponibile in Elastic Beanstalk. Installa la versione.

```
~$ rvm install 3.2
```

Testa l'installazione di Ruby.

```
~$ ruby --version
```

Installazione dell' AWS SDK per Ruby

Se devi gestire AWS le risorse dall'interno dell'applicazione, installa il. AWS SDK per Ruby Ad esempio, con l'SDK for Ruby, puoi utilizzare Amazon DynamoDB (DynamoDB) per archiviare le informazioni sulla sessione e sugli utenti senza creare un database relazionale.

Installa l'SDK for Ruby e le relative dipendenze con il comando `gem`.

```
$ gem install aws-sdk
```

Per ulteriori informazioni e le istruzioni di installazione, visita la homepage di AWS SDK per Ruby.

Installazione di un IDE o un editor di testo

Gli ambienti di sviluppo integrati (IDEs) offrono un'ampia gamma di funzionalità che facilitano lo sviluppo di applicazioni. Se non avete usato un IDE per lo sviluppo di Ruby, provate Aptana RubyMine e scoprite quale funziona meglio per voi.

- Installazione di Aptana
- RubyMine

> **Note**
>
> Un IDE potrebbe aggiungere file alla tua cartella di progetto di cui non intendi eseguire il commit nel controllo del codice sorgente. Per impedire il commit di questi file nel controllo

del codice sorgente, utilizza `.gitignore` o uno strumento di controllo del codice sorgente equivalente.

Se vuoi solo iniziare a scrivere codice e non hai bisogno di tutte le caratteristiche di un IDE, prendi in considerazione l'installazione di Sublime Text.

Utilizzo della piattaforma Ruby di Elastic Beanstalk

Questo argomento descrive come configurare, creare ed eseguire le applicazioni Ruby su Elastic Beanstalk.

AWS Elastic Beanstalk supporta una serie di rami di piattaforma per diverse versioni del linguaggio di programmazione Ruby. Vedi Ruby nel documento AWS Elastic Beanstalk Piattaforme per un elenco completo.

L'applicazione web Ruby può essere eseguita dietro un server proxy NGINX su un server di applicazioni Puma. Se lo utilizzi RubyGems, puoi includere un pacchetto `Gemfile`nel tuo pacchetto sorgente per installare i pacchetti durante la distribuzione.

Configurazione del server delle applicazioni

Elastic Beanstalk installa il server delle applicazioni Puma in base al ramo della piattaforma Ruby che scegli quando crei il tuo ambiente. Per ulteriori informazioni sui componenti forniti con le versioni della piattaforma Ruby, consulta la sezione Piattaforme supportate nella guida Piattaforme AWS Elastic Beanstalk .

È possibile configurare l'applicazione con il server Puma fornito. Ciò offre la possibilità di utilizzare una versione di Puma diversa da quella preinstallata con il ramo della piattaforma Ruby. È inoltre possibile configurare l'applicazione per utilizzare un server di applicazioni diverso, ad esempio Passenger. A tale scopo, devi includere e personalizzare un `Gemfile` nella tua implementazione. Dovresti inoltre configurare un `Procfile` per avviare il server delle applicazioni. Per ulteriori informazioni, consulta Configurazione del processo dell'applicazione con un Procfile.

Altre opzioni di configurazione

Elastic Beanstalk fornisce le opzioni di configurazione che puoi utilizzare per personalizzare il software in esecuzione sulle istanze Amazon Elastic Compute Cloud (Amazon EC2) nell'ambiente

Elastic Beanstalk. Puoi configurare le variabili di ambiente necessarie per la tua applicazione, accedere alla rotazione dei log su Amazon S3 e mappare le cartelle nel codice sorgente dell'applicazione che contengono i file statici ai percorsi serviti dal server proxy. La piattaforma, inoltre, predefinisce alcune variabili di ambiente comuni relative a Rails e Rack per facilitare il rilevamento e l'utilizzo.

Sono disponibili opzioni di configurazione specifiche della piattaforma nella console Elastic Beanstalk per la modifica della configurazione di un ambiente in esecuzione. Per conservare la configurazione dell'ambiente una volta terminata, puoi utilizzare configurazioni salvate per salvare le impostazioni e applicarle successivamente a un altro ambiente.

Per salvare le impostazioni nel codice sorgente, puoi includere i file di configurazione. Le impostazioni nei file di configurazione vengono applicate ogni volta che crei un ambiente o distribuisci l'applicazione. Puoi inoltre utilizzare i file di configurazione per installare pacchetti, eseguire script o altre operazioni di personalizzazione delle istanze durante le distribuzioni.

Le impostazioni applicate nella console Elastic Beanstalk sostituiscono le stesse impostazioni nei file di configurazione, se esistenti. Ciò ti consente di disporre di impostazioni predefinite nei file di configurazione, sostituibili con impostazioni specifiche per l'ambiente nella console. Per ulteriori informazioni sulla precedenza e altri metodi di modifica delle impostazioni, consulta Opzioni di configurazione.

Per informazioni dettagliate sui vari modi in cui è possibile estendere una piattaforma Elastic Beanstalk basata su Linux, consulta the section called “Estensione delle piattaforme Linux”.

Configurazione dell'ambiente Ruby

Puoi utilizzare la console Elastic Beanstalk per attivare la rotazione dei log in Amazon S3 e configurare variabili che l'applicazione è in grado di leggere dall'ambiente.

Per accedere alle impostazioni di configurazione software per l'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).

Opzioni di log

La sezione Log Options (Opzioni di log) ha due impostazioni:

- Instance profile (Profilo dell'istanza): specifica il profilo dell'istanza che dispone dell'autorizzazione ad accedere al bucket Amazon S3 associato all'applicazione.
- Enable log file rotation to Amazon S3 (Abilita rotazione file di log in Amazon S3): specifica se i file di log per le istanze Amazon EC2 dell'applicazione debbano essere copiati nel bucket Amazon S3 associato all'applicazione.

File statici

Per migliorare le prestazioni, la sezione Static files (File statici) consente di configurare il server proxy affinché fornisca file statici (ad esempio, HTML o immagini) da un set di directory all'interno dell'applicazione Web. Per ogni directory, è possibile impostare il percorso virtuale per la mappatura delle directory. Quando il server proxy riceve una richiesta da un file nel percorso specificato, questo serve il file direttamente anziché instradare la richiesta all'applicazione.

Per informazioni dettagliate sulla configurazione di file statici utilizzando file di configurazione o la console Elastic Beanstalk, consulta the section called “File statici”.

Per impostazione predefinita, il server proxy in un ambiente Ruby è configurato per servire file statici nel modo seguente:

- I file nelle cartelle `public` sono serviti dal percorso `/public` e dalla radice del dominio (percorso `/`).
- I file nelle sottocartelle `public/assets` sono serviti dal percorso `/assets`.

Negli esempi seguenti viene illustrato il funzionamento della configurazione predefinita:

- Se il codice sorgente dell'applicazione contiene un file con nome `logo.png` in una cartella chiamata `public`, il server proxy lo servirà agli utenti da *`subdomain`*`.elasticbeanstalk.com/public/logo.png` e *`subdomain`*`.elasticbeanstalk.com/logo.png`.
- Se il codice sorgente dell'applicazione contiene un file con nome `logo.png` in una cartella chiamata `assets` nella cartella `public`, il server proxy lo servirà agli utenti da *`subdomain`*`.elasticbeanstalk.com/assets/logo.png`.

È possibile configurare mappature aggiuntive per file statici. Per ulteriori informazioni, consulta Spazi dei nomi della configurazione Ruby più avanti in questo argomento.

> **Note**
>
> Per le versioni della piattaforma precedenti alla AL2 versione 3.3.7 di Ruby 2.7, la configurazione predefinita del server proxy Elastic Beanstalk nginx non supporta l'invio di file statici dal dominio root (). `subdomain.elasticbeanstalk.com/` Questa versione della piattaforma è stata rilasciata il 21 ottobre 2021. Per ulteriori informazioni, consulta Nuove versioni della piattaforma - Ruby nelle Note di rilascio di AWS Elastic Beanstalk .

Proprietà dell'ambiente

La sezione Environment Properties (Proprietà ambiente) consente di specificare le impostazioni di configurazione dell'ambiente nelle istanze Amazon EC2 che eseguono l'applicazione. Le proprietà di ambiente sono passate come coppie chiave-valore all'applicazione.

La piattaforma Ruby definisce le seguenti proprietà per la configurazione dell'ambiente:

- BUNDLE_WITHOUT: elenco di gruppi separati da due punti da ignorare quando si installano dipendenze da un Gemfile.
- BUNDLER_DEPLOYMENT_MODE: impostato su `true` (predefinito) per installare le dipendenze in deployment mode mediante Bundler. Imposta su `false` per eseguire `bundle install` in modalità di sviluppo.

  > **Note**
  >
  > Questa proprietà di ambiente non è definita sui rami della piattaforma Ruby AMI Amazon Linux (in precedenza Amazon Linux 2).

- RAILS_SKIP_ASSET_COMPILATION: impostalo su `true` per omettere l'esecuzione di `rake assets:precompile` durante la distribuzione.
- RAILS_SKIP_MIGRATIONS: impostalo su `true` per omettere l'esecuzione di `rake db:migrate` durante la distribuzione.
- RACK_ENV: specifica la fase di ambiente per Rack. Ad esempio, `development`, `production` o `test`.

All'interno dell'ambiente Ruby in esecuzione in Elastic Beanstalk, le variabili di ambiente sono accessibili tramite l'oggetto `ENV`. Ad esempio, è possibile leggere una proprietà denominata `API_ENDPOINT` su una variabile con il codice seguente:

```
endpoint = ENV['API_ENDPOINT']
```

Per ulteriori informazioni, consulta Variabili di ambiente e altre impostazioni software.

Spazi dei nomi della configurazione Ruby

È possibile utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

È possibile utilizzare lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles` dei nomi per configurare il proxy di ambiente per servire file statici. È possibile definire i mapping dei percorsi virtuali alle directory dell'applicazione.

La piattaforma Ruby non definisce spazi dei nomi specifici della piattaforma. ma proprietà di ambiente per opzioni comuni di Rails e Rack.

Il seguente file di configurazione specifica un'opzione file statici che esegue il mapping di una directory denominata `staticimages` al percorso `/images`, imposta ciascuna delle proprietà di ambiente definite dalla piattaforma e imposta una proprietà di ambiente aggiuntiva denominata `LOGGING`.

Example.ebextensions/ruby-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /images: staticimages
  aws:elasticbeanstalk:application:environment:
    BUNDLE_WITHOUT: test
    BUNDLER_DEPLOYMENT_MODE: true
    RACK_ENV: development
    RAILS_SKIP_ASSET_COMPILATION: true
    RAILS_SKIP_MIGRATIONS: true
```

```
    LOGGING: debug
```

> **Note**
>
> La proprietà dell'ambiente `BUNDLER_DEPLOYMENT_MODE` e lo spazio dei nomi `aws:elasticbeanstalk:environment:proxy:staticfiles` non sono definiti sui rami della piattaforma Ruby AMI Amazon Linux (in precedenza Amazon Linux 2).

In Elastic Beanstalk sono disponibili varie opzioni di configurazione per la personalizzazione dell'ambiente. Oltre ai file di configurazione, puoi impostare le opzioni di configurazione tramite la console, le configurazioni salvate, la CLI EB o AWS CLI. Per ulteriori informazioni, consulta Opzioni di configurazione.

Installazione di pacchetti con un Gemfile su Elastic Beanstalk

Da utilizzare RubyGems per installare i pacchetti richiesti dall'applicazione, includi un `Gemfile` file nella radice del codice sorgente del progetto.

Example Gemfile

```
source "https://rubygems.org"
gem 'sinatra'
gem 'json'
gem 'rack-parser'
```

Quando è presente un file `Gemfile`, Elastic Beanstalk esegue `bundle install` per installare le dipendenze. Per ulteriori informazioni, consulta le pagine Gemfile e Bundle pagine sul sito Web Bundler.io.

> **Note**
>
> Puoi usare una versione diversa di Puma oltre a quella predefinita preinstallata con la piattaforma Ruby. A tale scopo, includi una voce in un `Gemfile` che specifichi la versione. È inoltre possibile specificare un server di applicazioni diverso, ad esempio Passenger, utilizzando un `Gemfile` personalizzato.
> In entrambi i casi è necessario configurare un `Procfile` per avviare il server delle applicazioni.

Per ulteriori informazioni, consulta Configurazione del processo dell'applicazione con un Procfile.

Configurazione del processo di applicazione con un Procfile su Elastic Beanstalk.

Per specificare il comando che avvia l'applicazione Ruby, includere un file chiamato `Procfile` alla radice del bundle sorgente.

> **Note**
>
> Elastic Beanstalk non supporta questa caratteristica sui rami della piattaforma Ruby AMI Amazon Linux (in precedenza Amazon Linux 2). I rami delle piattaforme con nomi contenenti Puma o Passenger, indipendentemente dalle loro versioni di Ruby, precedono Amazon Linux 2 e non supportano la caratteristica `Procfile`.

Per informazioni dettagliate sulla scrittura e l'uso di a, consulta. `Procfile` Buildfile e Procfile

Quando non fornisci un`Procfile`, Elastic Beanstalk genera un valore predefinito. `Procfile` Se `Gemfile` includi Puma, Elastic Beanstalk presuppone che tu voglia usare la versione di Puma che hai fornito e genera la seguente impostazione predefinita. `Procfile`

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Se il tuo `Gemfile` non include Puma, Elastic Beanstalk presuppone che tu stia utilizzando il server delle applicazioni Puma preinstallato e genera il seguente valore predefinito. `Procfile` Sulle filiali della piattaforma Amazon Linux 2 Ruby, Elastic Beanstalk genera sempre il `Procfile` seguente valore predefinito se non fornisci un. `Procfile`

```
web: puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

> **Note**
>
> Il 10 ottobre 2024, le ultime filiali della piattaforma Ruby Amazon Linux 2 sono state ritirate. Tutti i rami della piattaforma Ruby attualmente supportati sono basati su Amazon Linux 2023.

Per informazioni sulla migrazione, consultaMigrazione da Amazon Linux 2 ad Amazon Linux 2023.

Se si desidera utilizzare il server delle applicazioni Passenger, utilizzare i seguenti file di esempio per configurare l'ambiente Ruby per installare e utilizzare Passenger.

1. Utilizzare questo file di esempio per installare Passenger.

   Example Gemfile

   ```
   source 'https://rubygems.org'
   gem 'passenger'
   ```

2. Utilizzare questo file di esempio per indicare a Elastic Beanstalk di avviare Passenger.

   Example Procfile

   ```
   web: bundle exec passenger start /var/app/current --socket /var/run/puma/my_app.sock
   ```

> **Note**
>
> Non è necessario modificare nulla nella configurazione del server proxy nginx per utilizzare Passenger. Per utilizzare altri server delle applicazioni, potrebbe essere necessario personalizzare la configurazione di nginx per inoltrare correttamente le richieste all'applicazione.

Distribuzione di un'applicazione Rails su Elastic Beanstalk

Rails è un framework open source model-view-controller (MVC) per Ruby. Questo tutorial illustra il processo di generazione di un'applicazione Rails e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

Sections

- Prerequisiti
- Conoscenza di base di Elastic Beanstalk

- Avvio di un ambiente Elastic Beanstalk
- Installazione di Rails e generazione di un sito Web
- Configurazione delle impostazioni di Rails
- Distribuzione dell'applicazione
- Pulizia
- Fasi successive

Prerequisiti

Conoscenza di base di Elastic Beanstalk

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Riga di comando

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata con Windows di Ubuntu e Bash.

Dipendenze Rails

Il framework Rails 6.1.4.1 ha le seguenti dipendenze. Assicurati di averli tutti installati.

- Ruby 2.5.0 o versioni successive: per istruzioni di installazione, consulta Configurazione dell'ambiente di sviluppo Ruby per Elastic Beanstalk.

  In questo tutorial utilizziamo Ruby 3.0.2 e la versione della piattaforma Elastic Beanstalk corrispondente.

- Node.js: per istruzioni di installazione, consulta Installazione di Node.js tramite il programma di gestione dei pacchetti.
- Yarn: per istruzioni di installazione, consulta Installazione sul sito Web Yarn.

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma Ruby e accetta le impostazioni predefinite e il codice di esempio.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- EC2 istanza: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per eseguire app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza dell'istanza: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa consente al traffico HTTP proveniente dal sistema di bilanciamento del carico di raggiungere l' EC2 istanza su cui è in esecuzione la tua app web. Per impostazione predefinita, il traffico non è consentito su altre porte.

- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.
- Gruppo di sicurezza Load Balancer: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon S3.

Installazione di Rails e generazione di un sito Web

Installa Rails e le relative dipendenze con il comando `gem`.

```
~$ gem install rails
Fetching: concurrent-ruby-1.1.9.gem
Successfully installed concurrent-ruby-1.1.9
Fetching: rack-2.2.3.gem
Successfully installed rack-2.2.3
...
```

Testa l'installazione di Rails.

```
~$ rails --version
Rails 6.1.4.1
```

Utilizza `rails new` con il nome dell'applicazione per creare un nuovo progetto Rails.

```
~$ rails new ~/eb-rails
```

Rails crea una directory con il nome specificato, genera tutti i file necessari per eseguire un progetto di esempio in locale, quindi esegue il bundler per installare tutte le dipendenze (Gems) definite nel Gemfile del progetto.

> **Note**
>
> Questo processo installa l'ultima versione di Puma per il progetto. Questa versione potrebbe essere diversa dalla versione fornita da Elastic Beanstalk sulla versione della piattaforma Ruby del tuo ambiente. Per visualizzare le versioni Puma fornite da Elastic Beanstalk, consulta Cronologia della piattaforma Ruby nella Guida per AWS Elastic Beanstalk Platforms. Per ulteriori informazioni sulla versione più recente di Puma, consulta il sito Web PUMA.io. Se c'è una mancata corrispondenza tra le due versioni di Puma, usa una delle seguenti opzioni:

- Utilizza la versione Puma installata dal comando `rails new` precedente. In questo caso, devi aggiungere un `Procfile` per consentire alla piattaforma di utilizzare la versione del server Puma fornita. Per ulteriori informazioni, consulta Configurazione del processo di applicazione con un Procfile su Elastic Beanstalk..
- Aggiorna la versione di Puma in modo che sia coerente con la versione preinstallata sulla versione della piattaforma Ruby del tuo ambiente. Per farlo, modifica la versione di Puma nel Gemfile situato nella root della directory di origine del progetto. Quindi esegui `bundle update`. Per ulteriori informazioni, consulta la pagina dell'aggiornamento del bundle sul sito Web di Bundler.io.

Testa la tua installazione Rails eseguendo il progetto predefinito in locale.

```
~$ cd eb-rails
~/eb-rails$ rails server
=> Booting Puma
=> Rails 6.1.4.1 application starting in development
=> Run `bin/rails server --help` for more startup options
Puma starting in single mode...
* Puma version: 5.5.2 (ruby 3.0.2-p107) ("Zawgyi")
*  Min threads: 5
*  Max threads: 5
*  Environment: development
*          PID: 77857
* Listening on http://127.0.0.1:3000
* Listening on http://[::1]:3000
Use Ctrl-C to stop
...
```

Apri `http://localhost:3000` in un browser Web per visualizzare il progetto predefinito in azione.

Questa pagina può essere visualizzata solo in modalità di sviluppo. Aggiungi alcuni contenuti alla pagina iniziale dell'applicazione per supportare una distribuzione di produzione in Elastic Beanstalk. Usa `rails generate` per creare un controller, una route e una visualizzazione per la pagina di benvenuto.

```
~/eb-rails$ rails generate controller WelcomePage welcome
      create  app/controllers/welcome_page_controller.rb
       route  get 'welcome_page/welcome'
      invoke  erb
      create    app/views/welcome_page
      create    app/views/welcome_page/welcome.html.erb
      invoke  test_unit
      create    test/controllers/welcome_page_controller_test.rb
      invoke  helper
      create    app/helpers/welcome_page_helper.rb
      invoke    test_unit
      invoke  assets
      invoke    coffee
```

```
      create      app/assets/javascripts/welcome_page.coffee
      invoke    scss
      create      app/assets/stylesheets/welcome_page.scss.
```

In questo modo, avrai tutto quello che ti serve per accedere alla pagina in `/welcome_page/welcome`. Prima di pubblicare le modifiche, tuttavia, cambia i contenuti della visualizzazione e aggiungi un instradamento per far apparire questa pagina nella parte superiore del sito.

Usa un editor di testo per modificare i contenuti in `app/views/welcome_page/welcome.html.erb`. Per questo esempio, potrai utilizzare `cat` per sovrascrivere semplicemente il contenuto del file esistente.

Example app/views/welcome_page/welcome.html.erb

```
<h1>Welcome!</h1>
<p>This is the front page of my first Rails application on Elastic Beanstalk.</p>
```

Infine, aggiungi il seguente instradamento a `config/routes.rb`:

Example config/routes.rb

```
Rails.application.routes.draw do
  get 'welcome_page/welcome'
  root 'welcome_page#welcome'
```

In questo modo, Rails instrada le richieste alla directory principale del sito Web al metodo di benvenuto del controller della pagina di benvenuto che trasforma il contenuto nella visualizzazione di benvenuto (`welcome.html.erb`).

Affinché Elastic Beanstalk distribuisca correttamente l'applicazione sulla piattaforma Ruby, è necessario aggiornare `Gemfile.lock`. Alcune dipendenze di `Gemfile.lock` potrebbero essere specifiche per la piattaforma. Pertanto, dobbiamo aggiungere **`platform ruby`** a `Gemfile.lock` in modo che tutte le dipendenze richieste vengano installate con la distribuzione.

Example

```
~/eb-rails$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/............
Resolving dependencies...
```

```
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-rails-doc-app/Gemfile.lock
```

Configurazione delle impostazioni di Rails

Usa la console Elastic Beanstalk per configurare Rails con proprietà dell'ambiente. Imposta la proprietà dell'ambiente `SECRET_KEY_BASE` su una stringa di massimo 256 caratteri alfanumerici.

Rails utilizza questa proprietà per creare le chiavi. Pertanto è necessario tenerla segreta e non memorizzarla nel controllo del codice sorgente in testo normale. La fornisci al codice Rails nel tuo ambiente attraverso una proprietà di ambiente.

Per configurare le variabili di ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri verso il basso fino alle variabili di ambiente Runtime.
6. Seleziona Aggiungi variabile di ambiente.
7. Per Sorgente seleziona Testo normale.

> **Note**
>
> I valori Secrets Manager e SSM Parameter Store nel menu a discesa servono a configurare le variabili di ambiente come segreti per archiviare dati sensibili, come credenziali e chiavi API. Per ulteriori informazioni, consulta Utilizzo di Elastic Gestione dei segreti AWS AWS Systems Manager Beanstalk con e Parameter Store.

8. Immettere le coppie del nome della variabile di ambiente e del valore della variabile di ambiente.
9. Se è necessario aggiungere altre variabili, ripetere i passaggi da 6 a 8.
10. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

A questo punto puoi distribuire il sito nell'ambiente.

Distribuzione dell'applicazione

Crea un bundle di origine che contiene i file creati da Rails. Il comando seguente crea un bundle di origine denominato `rails-default.zip`.

```
~/eb-rails$ zip ../rails-default.zip -r * .[^.]*
```

Carica il bundle di origine in Elastic Beanstalk per distribuire Rails nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Pulizia

Dopo aver finito di utilizzare il codice demo, puoi chiudere l'ambiente. Elastic Beanstalk elimina tutte le risorse AWS correlate, come istanze EC2 Amazon, istanze di database, sistemi di bilanciamento del carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Fasi successive

Per ulteriori informazioni su Rails, visita il sito rubyonrails.org.

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Distribuzione di un'applicazione Sinatra su Elastic Beanstalk

Questo procedura guidata illustra come implementare una semplice applicazione Web Sinatra in AWS Elastic Beanstalk.

Prerequisiti

Questo tutorial presuppone determinate conoscenze sulle operazioni di base di Elastic Beanstalk e della console Elastic Beanstalk. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk.

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una versione integrata per Windows di Ubuntu e Bash.

Sinatra 2.1.0 richiede Ruby 2.3.0 o versioni successive. In questo tutorial utilizziamo Ruby 3.0.2 e la versione della piattaforma Elastic Beanstalk corrispondente. Installa Ruby seguendo la procedura descritta in Configurazione dell'ambiente di sviluppo Ruby per Elastic Beanstalk.

Avvio di un ambiente Elastic Beanstalk

Utilizza la console Elastic Beanstalk per creare un ambiente Elastic Beanstalk. Scegli la piattaforma Ruby e accetta le impostazioni predefinite e il codice di esempio.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk utilizzando questo link preconfigurato: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare il ramo della piattaforma e la piattaforma che corrispondono al linguaggio utilizzato dall'applicazione.
3. In Application code (Codice applicazione), scegli Sample application (Applicazione di esempio).
4. Selezionare Review and launch (Controlla e avvia).
5. Verificare le opzioni disponibili. Scegliere l'opzione disponibile che si desidera utilizzare e, quando si è pronti, scegliere Create app (Crea app).

Occorrono circa cinque minuti per creare l'ambiente e le seguenti risorse:

- EC2 istanza: una macchina virtuale Amazon Elastic Compute Cloud (Amazon EC2) configurata per eseguire app Web sulla piattaforma scelta.

  Ogni piattaforma esegue un insieme specifico di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di essi. La maggior parte delle piattaforme utilizza Apache o NGINX come proxy inverso su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza dell'istanza: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa consente al traffico HTTP proveniente dal sistema di bilanciamento del carico di raggiungere l' EC2 istanza su cui è in esecuzione la tua app web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Sistema di bilanciamento del carico: un sistema di bilanciamento del carico Elastic Load Balancing configurato per distribuire richieste alle istanze in esecuzione sull'applicazione. Un sistema di bilanciamento del carico inoltre elimina la necessità di esporre le proprie istanze direttamente in Internet.

- Gruppo di sicurezza Load Balancer: un gruppo EC2 di sicurezza Amazon configurato per consentire il traffico in entrata sulla porta 80. Questa risorsa HTTP consente al traffico proveniente da Internet di raggiungere il sistema di bilanciamento del carico. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Gruppo Auto Scaling: un gruppo Auto Scaling configurato per sostituire un'istanza se viene terminata o diventa non disponibile.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e che vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

> **Sicurezza del dominio**
>
> Per aumentare la sicurezza delle tue applicazioni Elastic Beanstalk, il dominio elasticbeanstalk.com è registrato nella Public Suffix List (PSL).
> Se hai bisogno di impostare cookie sensibili nel nome di dominio predefinito per le tue applicazioni Elastic Beanstalk, ti consigliamo di utilizzare i cookie `__Host-` con un prefisso per una maggiore sicurezza. Questa pratica difende il tuo dominio dai tentativi di falsificazione delle richieste tra siti (CSRF). Per ulteriori informazioni, consulta la pagina Impostazione cookie nella pagina Mozilla Developer Network.

Tutte queste risorse sono gestite da Elastic Beanstalk. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

> **Note**
>
> Il bucket Amazon S3 creato da Elastic Beanstalk è condiviso tra gli ambienti e non viene eliminato quando l'ambiente viene terminato. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon S3.

Scrittura di un sito Web Sinatra Basic

Per creare e distribuire un'applicazione Sinatra

1. Crea un file di configurazione denominato config.ru con i seguenti contenuti.

   Example config.ru

   ```
   require './helloworld'
   run Sinatra::Application
   ```

2. Crea un file di codice Ruby denominato helloworld.rb con i seguenti contenuti.

   Example helloworld.rb

   ```
   require 'sinatra'
   get '/' do
     "Hello World!"
   end
   ```

3. Crea un Gemfile con i seguenti contenuti.

   Example Gemfile

   ```
   source 'https://rubygems.org'
   gem 'sinatra'
   gem 'puma'
   ```

4. Eseguire l'installazione del bundle per generare il `Gemfile.lock`

   Example

   ```
   ~/eb-sinatra$ bundle install
   Fetching gem metadata from https://rubygems.org/....
   Resolving dependencies...
   Using bundler 2.2.22
   Using rack 2.2.3
   ...
   ```

5. Affinché Elastic Beanstalk distribuisca correttamente l'applicazione sulla piattaforma Ruby, è necessario aggiornare `Gemfile.lock`. Alcune dipendenze di `Gemfile.lock` potrebbero essere specifiche per la piattaforma. Pertanto, dobbiamo aggiungere **`platform ruby`**

a `Gemfile.lock` in modo che tutte le dipendenze richieste vengano installate con la distribuzione.

Example

```
~/eb-sinatra$ bundle lock --add-platform ruby
Fetching gem metadata from https://rubygems.org/....
Resolving dependencies...
Writing lockfile to /Users/janedoe/EBDPT/RubyApps/eb-sinatra/Gemfile.lock
```

6. Crea un Procfile con i seguenti contenuti.

   Example Procfile

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Distribuzione dell'applicazione

Crea un bundle di origine che contiene i tuoi file di origine. Il comando seguente crea un bundle di origine denominato `sinatra-default.zip`.

```
~/eb-sinatra$ zip ../sinatra-default.zip -r * .[^.]*
```

Carica il bundle di origine in Elastic Beanstalk per distribuire Sinatra nel tuo ambiente.

Per distribuire un bundle di origine

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella pagina della panoramica dell'ambiente scegliere Upload and deploy (Carica e distribuisci).
4. Utilizzare la finestra di dialogo su schermo per caricare il bundle di origine.
5. Selezionare Deploy (Distribuisci).
6. Al termine della distribuzione, è possibile scegliere l'URL del sito per aprire il sito Web in una nuova scheda.

Pulizia

Dopo aver finito di utilizzare il codice demo, puoi chiudere l'ambiente. Elastic Beanstalk elimina tutte le risorse AWS correlate, come istanze EC2 Amazon, istanze di database, sistemi di bilanciamento del carico, gruppi di sicurezza e allarmi.

La rimozione delle risorse non elimina l'applicazione Elastic Beanstalk, quindi puoi creare nuovi ambienti per la tua applicazione in qualsiasi momento.

Per terminare l'ambiente Elastic Beanstalk dalla console

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Fasi successive

Per ulteriori informazioni su Sinatra, visita sinatrarb.com.

Mentre continui a sviluppare l'applicazione, potrebbe essere necessario un metodo per gestire gli ambienti e distribuire l'applicazione senza creare manualmente un file .zip e caricarlo sulla console Elastic Beanstalk. L'Elastic Beanstalk Command Line Interface (EB CLI easy-to-use) fornisce comandi per creare, configurare e distribuire applicazioni in ambienti Elastic Beanstalk dalla riga di comando.

Infine, se prevedi di usare l'applicazione in un ambiente di produzione, devi configurare un nome di dominio personalizzato per l'ambiente e abilitare HTTPS per le connessioni sicure.

Aggiungere un'istanza Amazon RDS DB all'ambiente Ruby Elastic Beanstalk

Questo argomento fornisce istruzioni per creare un Amazon RDS utilizzando la console Elastic Beanstalk. Puoi utilizzare un'istanza database Amazon Relational Database Service (Amazon RDS) per archiviare i dati raccolti e modificati dall'applicazione. Il database può essere accoppiato

all'ambiente e gestito da Elastic Beanstalk oppure può essere creato e gestito esternamente da un altro servizio. In queste istruzioni il database è accoppiato all'ambiente e gestito da Elastic Beanstalk. Per ulteriori informazioni sull'integrazione di un Amazon RDS con Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Sections

- Aggiunta di un'istanza database all'ambiente
- Download di un adattatore
- Connessione a un database

Aggiunta di un'istanza database all'ambiente

Per aggiungere un'istanza database al tuo ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, selezionare Configuration (Configurazione).
4. Nella categoria di configurazione del Database, scegliere Edit (Modifica).
5. Scegliere un motore di database e immettere un nome utente e una password.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'aggiunta di un'istanza database richiede circa 10 minuti. Quando l'aggiornamento dell'ambiente è completo, il nome host dell'istanza database e altre informazioni di connessione sono disponibili per la tua applicazione tramite le seguenti proprietà dell'ambiente:

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessio	Nella scheda Connectivity & security (Connettività e

Nome proprietà	Descrizione	Valore proprietà
	ni. Il valore predefinito varia tra i motori di database.	sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configura tion (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.

Per ulteriori informazioni sulla configurazione di un'istanza di database accoppiata a un ambiente Elastic Beanstalk, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Download di un adattatore

Aggiungi l'adattatore del database al Gemfile del tuo progetto.

Example Gemfile: Rails con MySQL

```
source 'https://rubygems.org'
gem 'puma'
gem 'rails', '~> 6.1.4', '>= 6.1.4.1'
gem 'mysql2'
```

Gemfile adattatore comune per Ruby

- MySQL – `mysql2`
- PostgreSQL – `pg`
- Oracle: `activerecord-oracle_enhanced-adapter`

- SQL Server – `activerecord-sqlserver-adapter`

Connessione a un database

Elastic Beanstalk fornisce informazioni di connessione per le istanze database collegate nelle proprietà dell'ambiente. Utilizza `ENV['VARIABLE']` per leggere le proprietà e configurare una connessione di database.

Example config/database.yml: configurazione database Ruby on Rails (MySQL)

```
production:
  adapter: mysql2
  encoding: utf8
  database: <%= ENV['RDS_DB_NAME'] %>
  username: <%= ENV['RDS_USERNAME'] %>
  password: <%= ENV['RDS_PASSWORD'] %>
  host: <%= ENV['RDS_HOSTNAME'] %>
  port: <%= ENV['RDS_PORT'] %>
```

Implementazione con contenitori Docker su Elastic Beanstalk

Questo capitolo spiega come utilizzare Elastic Beanstalk per distribuire applicazioni Web da contenitori Docker. I container Docker sono autonomi e includono tutte le informazioni di configurazione e il software di cui l'applicazione Web ha bisogno per l'esecuzione. Con i contenitori Docker puoi definire il tuo ambiente di runtime. Puoi anche scegliere il linguaggio di programmazione e le dipendenze delle applicazioni, come gestori di pacchetti o strumenti, che in genere non sono supportati da altre piattaforme Elastic Beanstalk.

Segui i passaggi QuickStart per Docker per creare un'applicazione Docker «Hello World» e distribuirla in un ambiente Elastic Beanstalk utilizzando l'EB CLI.

Argomenti

- Filiali della piattaforma Elastic Beanstalk Docker
- Utilizzo del ramo della piattaforma Elastic Beanstalk Docker
- Utilizzo del ramo della piattaforma Docker gestito da ECS in Elastic Beanstalk
- Autenticazione con archivi di immagini
- Configurazione degli ambienti Elastic Beanstalk Docker
- Piattaforme legacy

Filiali della piattaforma Elastic Beanstalk Docker

La piattaforma Docker Elastic Beanstalk include le seguenti ramificazioni della piattaforma:

Docker con Amazon Linux 2and Docker con AL2023

Elastic Beanstalk distribuisce i container Docker e il codice sorgente sulle istanze EC2 e le gestisce. Queste ramificazioni della piattaforma offrono supporto multi-container. È anche possibile sfruttare lo strumento Docker Compose per semplificare la configurazione, il test e l'implementazione delle applicazioni. Per ulteriori informazioni sulla versione della piattaforma, consulta the section called "Ramificazione della piattaforma Docker".

ECS in esecuzione su Amazon Linux 2 and ECS in esecuzione su AL2023

Forniamo questa filiale per i clienti che necessitano di un percorso di migrazione AL2023/AL2 dal ramo della piattaforma dismesso su cui è in esecuzione Multi-container Docker (Amazon Linux AMI). L'ultima versione delle ramificazioni della piattaforma supporta tutte le funzionalità della ramificazione ritirata. Non sono richieste modifiche al codice sorgente. Per ulteriori informazioni, consulta Migrazione dell'applicazione Elastic Beanstalk da Docker multi-container gestito da ECS a ECS su Amazon Linux 2023 AL1. Se non è disponibile un ambiente Elastic Beanstalk in esecuzione su una ramificazione della piattaforma basata su ECS, è consigliabile utilizzare l'altra ramificazione della piattaforma, Docker in esecuzione su AL2023 a 64 bit. Questo offre un approccio più semplice e richiede meno risorse.

Per un elenco delle versioni dei componenti software associate a ciascuno di questi rami della piattaforma, consulta Docker nel documento Piattaforme.AWS Elastic Beanstalk

Ramificazioni della piattaforma ritirate in esecuzione sull'AMI Amazon Linux (AL1)

Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate sull'AMI Amazon Linux (AL1) come ritirato. Espandi ogni sezione che segue per saperne di più su ogni ramificazione della piattaforma ritirata e sul relativo percorso di migrazione verso Amazon Linux 2 o Amazon Linux 2023 (scelta consigliata).

Docker (AMI Amazon Linux)

Questa ramificazione della piattaforma può essere utilizzata per implementare un'immagine Docker, descritta in un Dockerfile o una definizione `Dockerrun.aws.json` v1. Questa ramificazione della piattaforma esegue un solo container per ogni istanza. La ramificazione della piattaforma successiva, Docker in esecuzione su AL2023 a 64 bit e Docker in esecuzione su Amazon Linux 2 a 64 bit , supporta più container Docker per istanza.

Ti consigliamo di creare gli ambienti con la ramificazione della piattaforma più recente e supportata Docker in esecuzione su AL2023 a 64 bit. Potrai quindi migrare l'applicazione nell'ambiente appena creato. Per ulteriori informazioni sulla creazione di questi ambienti, consulta the section called “Ramificazione della piattaforma Docker”. Per ulteriori informazioni sulla migrazione, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Multi-container Docker (AMI Amazon Linux)

Questa versione della piattaforma utilizza Amazon ECS per coordinare un'implementazione di più container Docker in un cluster Amazon ECS di un ambiente Elastic Beanstalk. In caso di utilizzo

di questa ramificazione della piattaforma ritirata, consigliamo di eseguire la migrazione all'ultima versione della ramificazione della piattaforma ECS in esecuzione su Amazon Linux 2023. L'ultima versione della piattaforma supporta tutte le funzionalità di questa versione della piattaforma fuori produzione. Non sono richieste modifiche al codice sorgente. Per ulteriori informazioni, consulta Migrazione dell'applicazione Elastic Beanstalk da Docker multi-container gestito da ECS a ECS su Amazon Linux 2023 AL1.

Container Docker preconfigurati

Oltre alle piattaforme Docker menzionate in precedenza, esiste anche il ramo della GlassFish piattaforma Docker preconfigurato che funziona sul sistema operativo AMI Amazon Linux (AL1).

Questa ramificazione della piattaforma è stata sostituita da quelle in esecuzione su AL2023 a 64 bit e il Docker in esecuzione su Amazon Linux 2 a 64 bit. Per ulteriori informazioni, consulta Distribuzione di un' GlassFish applicazione sulla piattaforma Docker.

Utilizzo del ramo della piattaforma Elastic Beanstalk Docker

Questa sezione descrive come preparare l'immagine Docker per il lancio con uno dei rami della piattaforma Elastic Beanstalk Docker che esegue AL2 o AL2023.

Segui i passaggi QuickStart per Docker per creare un'applicazione Docker «Hello World» e distribuirla in un ambiente Elastic Beanstalk utilizzando l'EB CLI.

Argomenti

- QuickStart: distribuisci un'applicazione Docker su Elastic Beanstalk
- QuickStart: distribuisci un'applicazione Docker Compose su Elastic Beanstalk
- Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk

QuickStart: distribuisci un'applicazione Docker su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione Docker e la sua distribuzione in un ambiente. AWS Elastic Beanstalk

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti
- Passaggio 1: creare un'applicazione e un contenitore Docker
- Passaggio 2: Esegui l'applicazione localmente
- Fase 3: Implementa l'applicazione Docker con l'EB CLI
- Fase 4: Esegui l'applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti a un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Docker

Per seguire questo tutorial, avrai bisogno di un'installazione locale funzionante di Docker. Per ulteriori informazioni, consulta Ottieni Docker sul sito Web della documentazione di Docker.

Verifica che il demone Docker sia attivo e funzionante eseguendo il seguente comando.

```
~$ docker info
```

Passaggio 1: creare un'applicazione e un contenitore Docker

Per questo esempio, creiamo un'immagine Docker dell'applicazione Flask di esempio, anch'essa referenziata in. Implementazione di un'applicazione Flask in Elastic Beanstalk

L'applicazione è composta da due file:

- `app.py`— il file Python che contiene il codice che verrà eseguito nel contenitore.
- `Dockerfile`— il Dockerfile per creare la tua immagine.

Posiziona entrambi i file nella radice di una directory.

```
~/eb-docker-flask/
|-- Dockerfile
|-- app.py
```

Aggiungi i seguenti contenuti al tuo`Dockerfile`.

Example~/eb-docker- flask/Dockerfile

```
FROM public.ecr.aws/docker/library/python:3.12
COPY . /app
```

```
WORKDIR /app
RUN pip install Flask==3.1.1
EXPOSE 5000
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Aggiungi i seguenti contenuti al tuo file. `app.py`

Example~/eb-docker- .py flask/app

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello Elastic Beanstalk! This is a Docker application'
```

Usa il comando docker build per creare l'immagine del contenitore localmente, taggando l'immagine con. `eb-docker-flask` Il periodo (`.`) alla fine del comando specifica che il percorso è una directory locale.

```
~/eb-docker-flask$ docker build -t eb-docker-flask .
```

Passaggio 2: Esegui l'applicazione localmente

Esegui il tuo contenitore con il comando docker run. Il comando stamperà l'ID del contenitore in esecuzione. L'-dopzione esegue docker in modalità background. L'-popzione espone l'applicazione alla porta 5000. Elastic Beanstalk invia il traffico alla porta 5000 sulla piattaforma Docker per impostazione predefinita.

```
~/eb-docker-flask$ docker run -dp 127.0.0.1:5000:5000 eb-docker-flask
```

Accedi a nel tuo browser. `http://127.0.0.1:5000/` Dovresti vedere il testo «Hello Elastic Beanstalk! Questa è un'applicazione Docker».

Esegui il comando docker kill per terminare il contenitore.

```
~/eb-docker-flask$ docker kill container-id
```

Fase 3: Implementa l'applicazione Docker con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire l'applicazione Docker

1. Inizializza il repository della CLI EB con il comando eb init.

```
~/eb-docker-flask$ eb init -p docker docker-tutorial --region us-east-2
Application docker-tutorial has been created.
```

   Questo comando crea un'applicazione denominata `docker-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma Docker.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

```
~/eb-docker-flask$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

   Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk crea automaticamente un file zip per l'applicazione e lo avvia sulla porta 5000.

```
~/eb-docker-flask$ eb create docker-tutorial
```

   Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente.

Fase 4: Esegui l'applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-docker-flask$ eb open
```

Complimenti! Hai distribuito un'applicazione Docker con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione.

Fase 5: rimozione

Puoi chiudere il tuo ambiente quando finisci di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-docker-flask$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire applicazioni Docker localmente, consulta. Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare l'applicazione di esempio. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

QuickStart: distribuisci un'applicazione Docker Compose su Elastic Beanstalk

Questo QuickStart tutorial illustra il processo di creazione di un'applicazione Docker Compose multi-container e la sua distribuzione in un ambiente. AWS Elastic Beanstalk Creerai un'applicazione web Flask con un reverse proxy nginx per dimostrare come Docker Compose semplifica l'orchestrazione di più contenitori.

> **Non per uso in produzione**
>
> Gli esempi sono solo a scopo dimostrativo. Non utilizzare applicazioni di esempio in produzione.

Sections

- Vostro AWS account
- Prerequisiti

- Passaggio 1: creare un'applicazione Docker Compose
- Passaggio 2: Esegui l'applicazione localmente
- Passaggio 3: distribuisci l'applicazione Docker Compose con l'EB CLI
- Fase 4: Testa la tua applicazione su Elastic Beanstalk
- Fase 5: rimozione
- AWS risorse per la tua applicazione
- Fasi successive
- Esegui la distribuzione con la console Elastic Beanstalk

Vostro AWS account

Se non sei già AWS cliente, devi creare un AWS account. La registrazione ti consente di accedere a Elastic Beanstalk AWS e ad altri servizi di cui hai bisogno.

Se hai già un AWS account, puoi passare a. Prerequisiti

Crea un AWS account

Iscriviti a un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida introduttiva a un Account AWS nella Guida Gestione dell'account AWS di riferimento.

Prerequisiti

Per seguire le procedure in questa guida, devi usare un terminale a riga di comando o una shell per eseguire i comandi. I comandi vengono visualizzati in elenchi preceduti da un simbolo di prompt ($) e dal nome della directory corrente, se appropriato.

```
~/eb-project$ this is a command
this is output
```

Su Linux e macOS usa la tua shell e il tuo programma di gestione dei pacchetti preferiti. Su Windows puoi installare il sottosistema Windows per Linux per ottenere una Windows-integrated versione di Ubuntu e Bash.

CLI EB

Questo tutorial utilizza l'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per istruzioni dettagliate su come configurare e utilizzare la CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato) e Configurazione dell'interfaccia a riga di comando EB.

Docker e Docker Compose

Per seguire questo tutorial, avrai bisogno di un'installazione locale funzionante di Docker e Docker Compose. Per ulteriori informazioni, consulta Scarica Docker e installa Docker Compose sul sito Web della documentazione di Docker.

Verifica che Docker e Docker Compose siano installati e in esecuzione eseguendo i seguenti comandi.

```
~$ docker info
~$ docker compose version
```

Passaggio 1: creare un'applicazione Docker Compose

Per questo esempio, creiamo un'applicazione multi-contenitore utilizzando Docker Compose che consiste in un'applicazione web Flask e un reverse proxy nginx. Ciò dimostra come Docker Compose semplifica l'orchestrazione di più contenitori che funzionano insieme.

L'applicazione include una configurazione di monitoraggio dello stato che consente a Elastic Beanstalk di raccogliere metriche dettagliate dell'applicazione dal proxy nginx.

L'applicazione è costituita dalla seguente struttura:

```
~/eb-docker-compose-flask/
|-- docker-compose.yml
|-- web/
|   |-- Dockerfile
|   |-- app.py
|   `-- requirements.txt
|-- proxy/
|   |-- Dockerfile
|   `-- nginx.conf
`-- .platform/
    `-- hooks/
        `-- postdeploy/
            `-- 01_setup_healthd_permissions.sh
```

Create la struttura delle cartelle e aggiungete i seguenti file:

Innanzitutto, create il `docker-compose.yml` file principale che definisce i servizi e le relative relazioni.

Example~/eb-docker-compose- -compose.yml flask/docker

```
services:
  web:
    build: ./web
    expose:
      - "5000"

  nginx-proxy:
    build: ./proxy
    ports:
      - "80:80"
    volumes:
      - "/var/log/nginx:/var/log/nginx"
    depends_on:
      - web
```

Crea l'applicazione web Flask nella directory. `web` Aggiungi i seguenti contenuti al tuo `app.py` file.

Example~/eb-docker-compose- /app.py flask/web

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello Elastic Beanstalk! This is a Docker Compose application'
```

Aggiungi i seguenti contenuti al tuo servizio web. `Dockerfile`

Example~/eb-docker-compose- /Dockerfile flask/web

```
FROM public.ecr.aws/docker/library/python:3.12
COPY . /app
WORKDIR /app
RUN pip install Flask==3.1.1
EXPOSE 5000
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Crea il reverse proxy nginx nella directory. `proxy` Aggiungi i seguenti contenuti al tuo `nginx.conf` file.

Questa configurazione include la configurazione del monitoraggio dello stato che consente a Elastic Beanstalk di raccogliere metriche dettagliate delle applicazioni. Per ulteriori informazioni sulla personalizzazione dei formati dei log di monitoraggio dello stato di salute, consulta. Formato dei log dello stato migliorato

Example~/eb-docker-compose- /nginx.conf flask/proxy

```
events {
    worker_connections 1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    map $http_upgrade $connection_upgrade {
        default       "upgrade";
    }

    # Health monitoring log format for Elastic Beanstalk
    log_format healthd
 '$msec"$uri"$status"$request_time"$upstream_response_time"$http_x_forwarded_for';

    upstream flask_app {
        server web:5000;
    }

    server {
        listen 80 default_server;
        root /usr/share/nginx/html;

        # Standard access log
        access_log /var/log/nginx/access.log;

        # Health monitoring log for Elastic Beanstalk
        if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
            set $year $1;
            set $month $2;
            set $day $3;
            set $hour $4;
```

```
        }
        access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour
 healthd;

        location / {
            proxy_pass http://flask_app;
            proxy_http_version    1.1;

            proxy_set_header    Connection             $connection_upgrade;
            proxy_set_header    Upgrade                $http_upgrade;
            proxy_set_header    Host                   $host;
            proxy_set_header    X-Real-IP              $remote_addr;
            proxy_set_header    X-Forwarded-For        $proxy_add_x_forwarded_for;
        }
    }
}
```

Aggiungi i seguenti contenuti al tuo servizio proxy. `Dockerfile`

Example~/eb-docker-compose- /Dockerfile flask/proxy

```
FROM public.ecr.aws/nginx/nginx:alpine
COPY nginx.conf /etc/nginx/nginx.conf
EXPOSE 80
```

Infine, crea uno script hook per la piattaforma per configurare le directory di registro e le autorizzazioni necessarie per il monitoraggio dello stato di salute. Gli hook della piattaforma consentono di eseguire script personalizzati durante il processo di distribuzione. Per ulteriori informazioni sui platform hook, consulta. Hook della piattaforma

Example~/eb-docker-compose-flask/. platform/hooks/.sh postdeploy/01_setup_healthd_permissions

```
#!/bin/bash
set -ex

NGINX_CONTAINER=$(docker ps --filter "name=nginx-proxy" -q)

if [ -z "$NGINX_CONTAINER" ]; then
    echo "Error: No nginx-proxy container found running"
    exit 1
fi
```

```
NGINX_UID=$(docker exec ${NGINX_CONTAINER} id -u nginx)
NGINX_GID=$(docker exec ${NGINX_CONTAINER} id -g nginx)

mkdir -p /var/log/nginx/healthd
chown -R ${NGINX_UID}:${NGINX_GID} /var/log/nginx
```

Passaggio 2: Esegui l'applicazione localmente

Usa il comando docker compose up per creare ed eseguire la tua applicazione multi-contenitore localmente. Docker Compose creerà entrambe le immagini dei contenitori e avvierà i servizi definiti nel file. `docker-compose.yml`

```
~/eb-docker-compose-flask$ docker compose up --build
```

L'`--build`opzione garantisce che Docker Compose crei le immagini del contenitore prima di avviare i servizi. Dovresti vedere un output che mostra sia il servizio web che il servizio nginx-proxy in fase di avvio.

Vai a `http://localhost` nel tuo browser. Dovresti vedere il testo «Hello Elastic Beanstalk! Questa è un'applicazione Docker Compose». Il proxy nginx riceve la tua richiesta sulla porta 80 e la inoltra all'applicazione Flask in esecuzione sulla porta 5000.

Al termine del test, interrompi l'applicazione premendo Ctrl+C nel terminale o esegui il seguente comando in un terminale separato:

```
~/eb-docker-compose-flask$ docker compose down
```

Passaggio 3: distribuisci l'applicazione Docker Compose con l'EB CLI

Esegui i seguenti comandi per creare un ambiente Elastic Beanstalk per questa applicazione.

Per creare un ambiente e distribuire l'applicazione Docker Compose

1. Inizializza il repository della CLI EB con il comando eb init.

```
~/eb-docker-compose-flask$ eb init -p docker docker-compose-tutorial --region us-east-2
Application docker-compose-tutorial has been created.
```

Questo comando crea un'applicazione denominata `docker-compose-tutorial` e configura il repository locale per creare ambienti con l'ultima versione della piattaforma Docker.

2. (Facoltativo) Esegui nuovamente eb init per configurare una coppia di chiavi predefinite e connetterti all'istanza EC2 che esegue l'applicazione.

```
~/eb-docker-compose-flask$ eb init
Do you want to set up SSH for your instances?
(y/n): y
Select a keypair.
1) my-keypair
2) [ Create new KeyPair ]
```

Seleziona una coppia di chiavi se ne hai già una, oppure segui le istruzioni per creare una nuova. Se il prompt non è visualizzato o devi modificare le impostazioni in seguito, esegui eb init -i.

3. Crea un ambiente e distribuisci in esso l'applicazione con eb create: Elastic Beanstalk `docker-compose.yml` rileva automaticamente il file e distribuisce l'applicazione multi-contenitore.

```
~/eb-docker-compose-flask$ eb create docker-compose-tutorial
```

Elastic Beanstalk impiega circa cinque minuti per creare l'ambiente e distribuire l'applicazione multi-container.

Fase 4: Testa la tua applicazione su Elastic Beanstalk

Una volta completato il processo di creazione dell'ambiente, apri il sito Web con. eb open

```
~/eb-docker-compose-flask$ eb open
```

Ottimo! Hai distribuito un'applicazione Docker Compose multi-container con Elastic Beanstalk! Si aprirà una finestra del browser che utilizza il nome di dominio creato per l'applicazione. Dovresti vedere il messaggio dalla tua applicazione Flask, servito tramite il reverse proxy nginx.

Fase 5: rimozione

È possibile chiudere l'ambiente quando si finisce di lavorare con l'applicazione. Elastic Beanstalk AWS interrompe tutte le risorse associate all'ambiente.

Per terminare l'ambiente Elastic Beanstalk con l'EB CLI, esegui il comando seguente.

```
~/eb-docker-compose-flask$ eb terminate
```

AWS risorse per la tua applicazione

Hai appena creato un'applicazione a istanza singola che esegue più contenitori. Funge da semplice applicazione di esempio con una singola istanza EC2, quindi non richiede il bilanciamento del carico o la scalabilità automatica. Per le applicazioni a istanza singola Elastic Beanstalk crea le seguenti risorse: AWS

- Istanza EC2: una macchina virtuale di Amazon EC2 configurata per eseguire applicazioni Web sulla piattaforma selezionata.

  Ogni piattaforma esegue un insieme diverso di software, file di configurazione e script per supportare una versione della lingua, un framework, un container Web specifici o una determinata combinazione di tutti questi elementi. La maggior parte delle piattaforme utilizza Apache o nginx come proxy inverso che elabora il traffico web su cui viene eseguita l'app Web, inoltra le richieste all'app, fornisce asset statici e genera log degli accessi e di errore.
- Gruppo di sicurezza istanze: un gruppo di sicurezza di Amazon EC2 configurato per abilitare il traffico in entrata sulla porta 80. Questa risorsa consente al traffico TTP proveniente dal load balancer di raggiungere l'istanza EC2 in esecuzione sull'app Web. Per impostazione predefinita, il traffico non è consentito su altre porte.
- Bucket Amazon S3: posizione di storage per il codice sorgente, i log e altri artefatti creati quando si utilizza Elastic Beanstalk.
- CloudWatch Allarmi Amazon: due CloudWatch allarmi che monitorano il carico sulle istanze nel tuo ambiente e vengono attivati se il carico è troppo alto o troppo basso. Quando viene attivato un allarme, il gruppo Auto Scaling aumenta o diminuisce di conseguenza.
- CloudFormation stack: Elastic CloudFormation Beanstalk utilizza per avviare le risorse nell'ambiente e propagare le modifiche alla configurazione. Le risorse sono definite in un modello, visualizzabile nella console CloudFormation.
- Nome di dominio: un nome di dominio che indirizza alla tua app Web nel modulo. *subdomain region*.elasticbeanstalk.com.

Elastic Beanstalk gestisce tutte queste risorse. Quando arresti l'ambiente, Elastic Beanstalk termina tutte le risorse che contiene. L'applicazione Docker Compose esegue più contenitori sulla singola istanza EC2, con Elastic Beanstalk che gestisce automaticamente l'orchestrazione.

Fasi successive

Quando disponi di un'applicazione in esecuzione in un ambiente, puoi distribuire una nuova versione dell'applicazione o un'applicazione diversa in qualsiasi momento. La distribuzione di una nuova versione dell'applicazione è molto veloce perché non richiede il provisioning o il riavvio delle istanze EC2. Puoi anche esplorare il tuo nuovo ambiente utilizzando la console Elastic Beanstalk. Per i passaggi dettagliati, consulta Esplora il tuo ambiente nel capitolo Guida introduttiva di questa guida.

Dopo aver distribuito una o due applicazioni di esempio e essere pronti per iniziare a sviluppare ed eseguire le applicazioni Docker Compose localmente, vedi. Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk

Esegui la distribuzione con la console Elastic Beanstalk

Puoi anche utilizzare la console Elastic Beanstalk per avviare un'applicazione Docker Compose. Crea un file ZIP contenente il tuo `docker-compose.yml` file e tutte le directory e i file associati, quindi caricalo quando crei una nuova applicazione. Per i passaggi dettagliati, consulta Creare un'applicazione di esempio nel capitolo Guida introduttiva di questa guida.

Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk

Questa sezione descrive come preparare l'immagine Docker per la distribuzione su Elastic Beanstalk con uno dei rami della piattaforma Docker AL2 o AL2023. I file di configurazione necessari dipendono dal fatto che le immagini siano locali, remote e se utilizzi Docker Compose.

> **Note**
>
> Per un esempio di procedura che avvia un ambiente Docker, consulta l'argomento. QuickStart per Docker

Argomenti

- Gestione delle immagini con Docker Compose in Elastic Beanstalk
- Gestione delle immagini senza Docker Compose in Elastic Beanstalk
- Creazione di immagini personalizzate con un Dockerfile

Gestione delle immagini con Docker Compose in Elastic Beanstalk

Puoi scegliere di utilizzare Docker Compose per gestire vari servizi in un unico file YAML. Per saperne di più su Docker Compose, vedi Perché usare Compose? sul sito Web di Docker.

- Creare un `docker-compose.yml`. Questo file è necessario se utilizzi Docker Compose per gestire la tua applicazione con Elastic Beanstalk. Se tutte le distribuzioni provengono da immagini in archivi pubblici, non sono necessari altri file di configurazione. Se le immagini di origine della distribuzione si trovano in un repository privato, sarà necessario eseguire alcune configurazioni aggiuntive. Per ulteriori informazioni, consulta Utilizzo di immagini da un repository privato. Per ulteriori informazioni sul file`docker-compose.yml` , consulta le informazioni di riferimento sul file Compose sul sito Web Docker.
- Il `Dockerfile` è facoltativo. Creane una se hai bisogno di Elastic Beanstalk per creare ed eseguire un'immagine personalizzata locale. Per ulteriori informazioni sul file `Dockerfile`, consulta le informazioni di riferimento su Dockerfile sul sito Web Docker.
- Potrebbe essere necessario creare un file. `.zip` Se si utilizza solo un `Dockerfile` file per distribuire l'applicazione, non è necessario crearne uno. Se si utilizzano file di configurazione aggiuntivi, il file.zip deve includere il`Dockerfile`, il file, i `docker-compose.yml` file dell'applicazione e tutte le dipendenze tra i file dell'applicazione. L'`Dockerfile`e `docker-compose.yml` deve trovarsi alla radice, o al livello superiore, dell'archivio.zip. Se si utilizza l'EB CLI per distribuire l'applicazione, viene creato automaticamente un file.zip.

Per ulteriori informazioni su Docker Compose e su come installarlo, vedere la sezione Panoramica dei siti Docker Compose e Installazione di Docker Compose.

Gestione delle immagini senza Docker Compose in Elastic Beanstalk

Se non utilizzi Docker Compose per gestire le immagini Docker, dovrai configurare un file, uno o entrambi. `Dockerfile` `Dockerrun.aws.json`

- Crea un file `Dockerfile` per far creare a Elastic Beanstalk ed eseguire un'immagine personalizzata localmente.
- Crea un file `Dockerrun.aws.json v1` per distribuire un'immagine Docker su Elastic Beanstalk da un repository in hosting.
- Potrebbe essere necessario creare un file. `.zip` Se si utilizza solo uno dei due file, il `Dockerfile` o il`Dockerrun.aws.json`, non è necessario creare un file.zip. Se si utilizzano entrambi i file, è necessario un file.zip. Il file.zip deve includere sia il che`Dockerrun.aws.json`, oltre al file

contenente i file dell'applicazione `Dockerfile` e tutte le dipendenze dei file dell'applicazione. Se si utilizza la CLI EB per distribuire l'applicazione, crea un `.zip` file per te.

`Dockerrun.aws.json` file di configurazione v1

Un file `Dockerrun.aws.json` descrive come distribuire un'immagine Docker in remoto come applicazione Elastic Beanstalk. Questo file JSON è specifico per Elastic Beanstalk. Se la tua applicazione viene eseguita su un'immagine disponibile in un repository in hosting, puoi specificare l'immagine in un file `Dockerrun.aws.json v1` e omettere il `Dockerfile`.

> **`Dockerrun.aws.json`** versioni
>
> Il parametro `AWSEBDockerrunVersion` indica la versione del file `Dockerrun.aws.json`.
>
> - Le piattaforme Docker AL2 e AL2023 utilizzano le seguenti versioni del file.
> - `Dockerrun.aws.json v3`— ambienti che utilizzano Docker Compose.
> - `Dockerrun.aws.json v1`— ambienti che non utilizzano Docker Compose.
> - ECS in esecuzione su Amazon Linux 2 ed ECS in esecuzione su AL2023 utilizzano il file. `Dockerrun.aws.json v2` Anche la piattaforma ritirata ECS-The Multicontainer Docker Amazon Linux AMI (AL1) utilizzava questa stessa versione.

Dockerrun.aws.json v1

Chiavi e valori validi per il file `Dockerrun.aws.json v1` includono le seguenti operazioni:

AWSEBDockerrunVersion

(Obbligatorio) Specificate il numero di versione 1 se non utilizzate Docker Compose per gestire l'immagine.

Autenticazione

(Obbligatorio solo per i repository privati) Specifica l'oggetto Amazon S3 in cui è archiviato il file `.dockercfg`.

Vedi Autenticazione con archivi di immagini Usare immagini da un archivio privato più avanti in questo capitolo.

Immagine

Specifica l'immagine di base Docker su un repository Docker esistente da cui crei un container Docker. Specificare il valore della chiave Name nel formato *<organization>/<image name>* per le immagini su Docker Hub o *<site>/<organization name>/<image name>* per altri siti.

Quando nel file `Dockerrun.aws.json` si specifica un'immagine, ogni istanza nell'ambiente Elastic Beanstalk esegue `docker pull` per eseguire l'immagine. Opzionalmente, è possibile includere la chiave Update. Il valore predefinito è `true` e indica a Elastic Beanstalk di controllare il repository, estrarre gli aggiornamenti per l'immagine e sovrascrivere le immagini memorizzate nella cache.

Quando usi un file `Dockerfile`, non specificare la chiave Image nel file `Dockerrun.aws.json`. Elastic Beanstalk crea e utilizza sempre l'immagine descritta nel file `Dockerfile` quando presente.

Porte

(Obbligatorio quando specifichi la chiave Image) Elenca le porte da esporre sul container Docker. Elastic Beanstalk ContainerPortutilizza il valore per connettere il contenitore Docker al reverse proxy in esecuzione sull'host.

È possibile specificare più porte del container, ma Elastic Beanstalk utilizza solo la prima porta. Utilizza questa porta per connettere il container al proxy inverso dell'host e instradare le richieste da Internet pubblico. Se utilizzi un**`Dockerfile`**, il primo ContainerPortvalore deve corrispondere alla prima voce dell'elenco EXPOSE. **`Dockerfile`**

Facoltativamente, puoi specificare un elenco di porte in. HostPort HostPortle voci specificano le porte host a cui sono mappati ContainerPorti valori. Se non si specifica un HostPortvalore, il valore predefinito è il valore. ContainerPort

```
{
  "Image": {
    "Name": "image-name"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8000
```

```
    }
  ]
}
```

Volumi

Mappa volumi da un'istanza EC2 al tuo container Docker. Specifica uno o più array di volumi da mappare.

```
{
  "Volumes": [
    {
      "HostDirectory": "/path/inside/host",
      "ContainerDirectory": "/path/inside/container"
    }
  ]
...
```

Registrazione di log

Specificare la directory all'interno del container in cui l'applicazione scrive i log. Elastic Beanstalk carica gli eventuali log presenti in questa directory in Amazon S3 quando richiedi la parte finale del log o log in bundle. Se esegui l'accesso a una cartella denominata `rotated` all'interno di questa directory, puoi anche configurare Elastic Beanstalk per il caricamento di log ruotati in Amazon S3 per lo storage permanente. Per ulteriori informazioni, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Comando

Specificare un comando da eseguire nel container. Se specifichi un Entrypoint, Command viene aggiunto come argomento in Entrypoint. Per ulteriori informazioni, consulta CMD nella documentazione Docker.

Entrypoint

Specificare un comando predefinito da eseguire all'avvio del container. Per ulteriori informazioni, consulta la sezione relativa a ENTRYPOINT nella documentazione Docker.

Il seguente frammento è un esempio che illustra la sintassi del file `Dockerrun.aws.json` per un singolo container.

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "janedoe/image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
  "Volumes": [
    {
      "HostDirectory": "/var/app/mydb",
      "ContainerDirectory": "/etc/mysql"
    }
  ],
  "Logging": "/var/log/nginx",
  "Entrypoint": "/app/bin/myapp",
  "Command": "--argument"
}>
```

Puoi fornire a Elastic Beanstalk solo il file `Dockerrun.aws.json` o un archivio `.zip` contenente i file `Dockerrun.aws.json` e `Dockerfile`. Quando fornisci entrambi i file, il `Dockerfile` descrive l'immagine Docker e il file `Dockerrun.aws.json` fornisce informazioni aggiuntive per la distribuzione, come descritto di seguito in questa sezione.

> **Note**
>
> I due file devono essere al livello principale, o al livello superiore, dell'archivio `.zip`. Non creare l'archivio da una directory contenente i file. Passa invece a tale directory e crea lì l'archivio.
>
> Quando fornisci entrambi i file, non specificare un'immagine nel file `Dockerrun.aws.json`. Elastic Beanstalk crea e utilizza l'immagine descritta nel file `Dockerfile` e ignora l'immagine specificata nel file `Dockerrun.aws.json`.

Creazione di immagini personalizzate con un Dockerfile

Occorre creare un `Dockerfile` se non si dispone già di un'immagine ospitata in un repository.

Il seguente frammento è un esempio del `Dockerfile`. Quando segui le istruzioni in QuickStart per Docker, puoi caricare questo `Dockerfile` come indicato. Elastic Beanstalk esegue il gioco 2048 quando utilizzi questo `Dockerfile`.

Per ulteriori informazioni sulle istruzioni che puoi includere nel `Dockerfile`, consulta la documentazione di riferimento del Dockerfile sul sito Web Docker.

```
FROM ubuntu:12.04

RUN apt-get update
RUN apt-get install -y nginx zip curl

RUN echo "daemon off;" >> /etc/nginx/nginx.conf
RUN curl -o /usr/share/nginx/www/master.zip -L https://codeload.github.com/
gabrielecirulli/2048/zip/master
RUN cd /usr/share/nginx/www/ && unzip master.zip && mv 2048-master/* . && rm -rf 2048-
master master.zip

EXPOSE 80

CMD ["/usr/sbin/nginx", "-c", "/etc/nginx/nginx.conf"]
```

> **Note**
>
> È possibile eseguire build multi-fase da un singolo Dockerfile per produrre immagini di dimensioni più piccole con una significativa riduzione della complessità. Per ulteriori informazioni, consultare Utilizzo di build multi-fase sul sito Web della documentazione Docker.

Utilizzo del ramo della piattaforma Docker gestito da ECS in Elastic Beanstalk

Questo argomento fornisce una panoramica dei rami della piattaforma Docker gestita da Elastic Beanstalk ECS per Amazon Linux 2 e Amazon Linux 2023. Fornisce inoltre informazioni di configurazione specifiche per la piattaforma gestita Docker ECS.

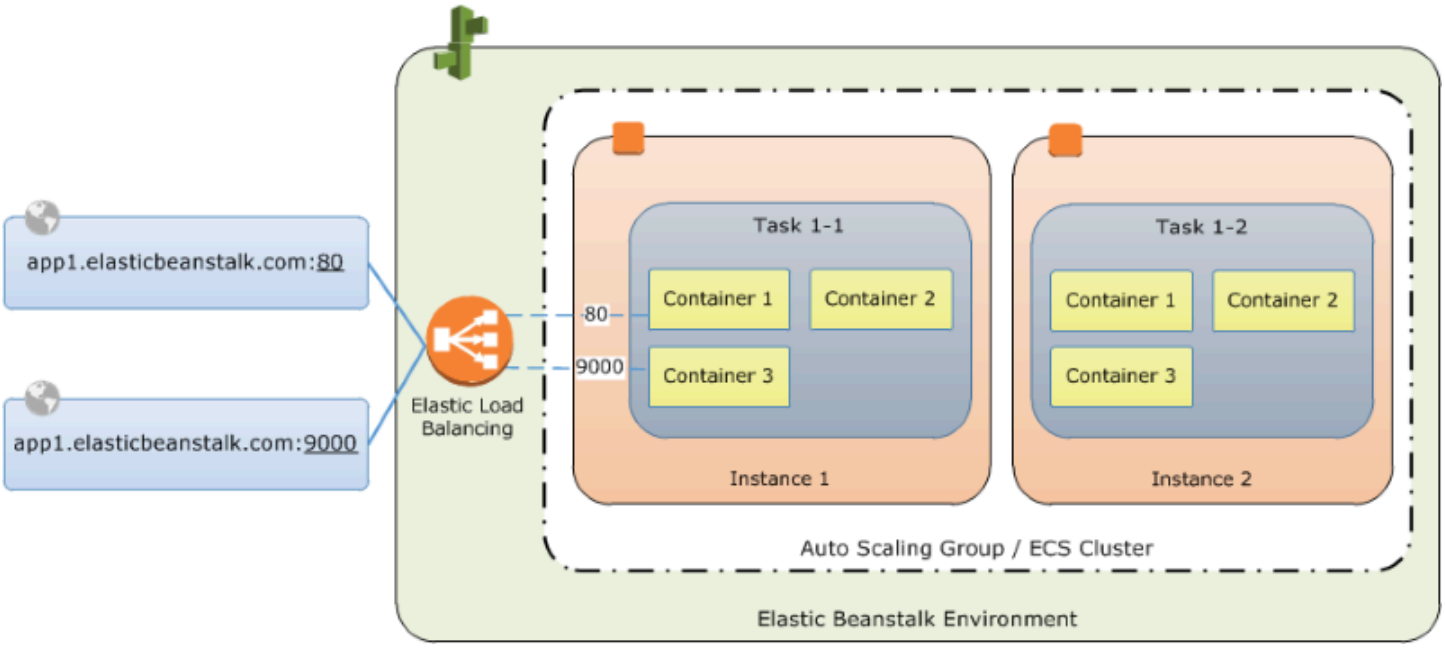
Migrazione da Multi-container Docker in AL1

Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Sebbene questo capitolo fornisca informazioni sulla configurazione per questa piattaforma ritirata, ti consigliamo vivamente di migrare all'ultima filiale della piattaforma supportata. Se al momento stai utilizzando il Multi-container Docker in esecuzione su AL1 platform branch, puoi migrare alla versione più recente di ECS Running on platform branch. AL2023 L'ultima versione della ramificazione della piattaforma supporta tutte le funzionalità della ramificazione fuori produzione. Non sono richieste modifiche al codice sorgente. Per ulteriori informazioni, consulta Migrazione dell'applicazione Elastic Beanstalk da Docker multi-container gestito da ECS a ECS su Amazon Linux 2023 AL1.

Panoramica della piattaforma Docker gestita da ECS

Elastic Beanstalk utilizza Amazon Elastic Container Service (Amazon ECS) per coordinare le distribuzioni dei container in ambienti Docker multicontainer. Amazon ECS fornisce strumenti per la gestione di un cluster di istanze che eseguono container Docker. Elastic Beanstalk si occupa delle attività Amazon ECS, tra cui la creazione di cluster, la definizione delle attività e l'esecuzione. Le istanze nell'ambiente eseguono lo stesso set di container, che sono definiti in un file `Dockerrun.aws.json` v2. Per ottenere il massimo da Docker, Elastic Beanstalk ti consente di creare un ambiente in cui le istanze Amazon EC2 eseguono più container Docker affiancati.

Il diagramma seguente mostra un esempio di ambiente Elastic Beanstalk configurato con tre container Docker in esecuzione su ogni istanza Amazon EC2 in un gruppo Auto Scaling:



Risorse Amazon ECS create da Elastic Beanstalk

Quando crei un ambiente utilizzando la piattaforma Docker gestita da ECS, Elastic Beanstalk crea e configura automaticamente diverse risorse Amazon Elastic Container Service durante la creazione dell'ambiente. In questo modo, crea i container necessari su ogni istanza di Amazon EC2.

- Cluster Amazon ECS: le istanze di container in Amazon ECS sono organizzate in cluster. Se usato con Elastic Beanstalk, viene sempre creato un cluster per ogni ambiente Docker gestito da ECS. Un cluster ECS contiene anche fornitori di capacità di gruppo Auto Scaling e altre risorse.
- Definizione attività Amazon ECS: Elastic Beanstalk usa il file `Dockerrun.aws.json` v2 nel progetto per generare la definizione delle attività Amazon ECS utilizzata per configurare le istanze di container nell'ambiente.
- Attività Amazon ECS: Elastic Beanstalk comunica con Amazon ECS per eseguire un'attività su ogni istanza nell'ambiente e coordinare la distribuzione dei container. In un ambiente scalabile, Elastic Beanstalk avvia una nuova attività ogni volta che un'istanza viene aggiunta al cluster.
- Agente container Amazon ECS: l'agente viene eseguito in un container Docker sulle istanze nell'ambiente. L'agente esegue il polling del servizio Amazon ECS e attende l'esecuzione di un'attività.
- Volumi di dati Amazon ECS: oltre ai volumi definiti nella versione `Dockerrun.aws.json` 2, Elastic Beanstalk inserisce le definizioni dei volumi nella definizione delle attività per facilitare la raccolta dei log.

  Elastic Beanstalk crea volumi di log sull'istanza di container, uno per ogni container Docker, in `/var/log/containers/`*`containername`*. Questi volumi sono denominati `awseb-logs-`*`containername`* e vengono forniti per il montaggio dei container. Vedi Formato della definizione del container per ulteriori informazioni su come montarli.

Per ulteriori informazioni sulle risorse di Amazon ECS, consulta la Amazon Elastic Container Service Developer Guide.

`Dockerrun.aws.json`file v2

Le istanze del contenitore richiedono un file di configurazione denominato. `Dockerrun.aws.json` Le istanze di container si riferiscono alle istanze Amazon EC2 che eseguono Docker gestito da ECS in un ambiente Elastic Beanstalk. Questo file è specifico per Elastic Beanstalk e può essere utilizzato da solo o abbinato a un codice sorgente e a contenuti in un bundle di origine per creare un ambiente su una piattaforma Docker.

> **Note**
>
> La versione 2 del formato `Dockerrun.aws.json` aggiunge il supporto per più container per l'istanza Amazon EC2 e può essere utilizzata solo con la piattaforma Docker gestita da ECS. Il formato differisce notevolmente dalle altre versioni dei file di configurazione che supportano i rami della piattaforma Docker che non sono gestiti da ECS.

`Dockerrun.aws.json v2`Per ulteriori dettagli sul formato aggiornato e su un file di esempio, consulta la

Immagini Docker

La piattaforma Docker gestita da ECS per Elastic Beanstalk richiede che le immagini siano precostruite e archiviate in un archivio di immagini online pubblico o privato prima di creare un ambiente Elastic Beanstalk.

> **Note**
>
> La creazione di immagini personalizzate durante la distribuzione con un `Dockerfile` non è supportata dalla piattaforma Docker gestita da ECS su Elastic Beanstalk. Crea le immagini e distribuiscile in un repository online prima di creare un ambiente Elastic Beanstalk.

Specifica le immagini per nome in `Dockerrun.aws.json` v2

Per configurare Elastic Beanstalk in modo che effettui l'autenticazione su un repository privato, includi il parametro `authentication` nel file `Dockerrun.aws.json` v2.

Distribuzioni di container non riuscite

Se un'attività Amazon ECS ha esito negativo, uno o più container nell'ambiente Elastic Beanstalk non verranno avviati. Elastic Beanstalk non esegue il rollback degli ambienti multicontainer a causa di un'attività Amazon ECS non riuscita. Se un container non viene avviato nell'ambiente, ridistribuisci la versione corrente o una versione funzionante precedente dalla console Elastic Beanstalk.

Per distribuire una versione esistente

1. Apri la console Elastic Beanstalk nella regione dell'ambiente in uso.

2. Fai clic su Actions (Operazioni) a destra del nome dell'applicazione, quindi su View application versions (Visualizza versioni dell'applicazione).
3. Seleziona una versione dell'applicazione e fai clic su Deploy (Distribuisci).

Estensione delle piattaforme Docker basate su ECS per Elastic Beanstalk

Elastic Beanstalk offre funzionalità di estensibilità che consentono di applicare comandi, script, software e configurazioni personalizzati alle distribuzioni delle applicazioni. Il flusso di lavoro di implementazione per ECS AL2 e i rami della AL2023 piattaforma varia leggermente rispetto alle altre piattaforme basate su Linux. Per ulteriori informazioni, consulta Flusso di lavoro di implementazione di istanze per ECS in esecuzione su Amazon Linux 2 e versioni successive.

Configurazione Docker gestita da ECS per Elastic Beanstalk

Questo capitolo spiega come configurare l'ambiente Docker gestito da ECS. L'elenco seguente riassume gli elementi di configurazione illustrati in questo capitolo.

- `Dockerrun.aws.jsonv2` — Questo file di configurazione specifica il tuo repository di immagini e il nome delle tue immagini Docker, tra gli altri componenti.
- Ruolo del profilo dell'istanza EC2: se disponi di un profilo di istanza personalizzato, spieghiamo come configurarlo in modo che le autorizzazioni richieste a ECS per la gestione dei contenitori rimangano aggiornate.
- Elastic Load Balancing listener: dovrai configurare più listener Elastic Load Balancing se hai bisogno che il tuo ambiente supporti il traffico in entrata per proxy o altri servizi che non funzionano sulla porta HTTP predefinita.

Argomenti

- Configurazione del file dockerRun.aws.json v2
- Politica gestita dai container e ruolo dell'istanza EC2
- Utilizzo di più listener Elastic Load Balancing

Configurazione del file dockerRun.aws.json v2

`Dockerrun.aws.json v2` è un file di configurazione Elastic Beanstalk che descrive come implementare un set di container Docker ospitati in un cluster ECS in un ambiente Elastic Beanstalk.

La piattaforma Elastic Beanstalk crea una definizione di attività ECS, che include una definizione di container ECS. Queste definizioni sono descritte nel file di configurazione `Dockerrun.aws.json`.

La definizione del container nel file `Dockerrun.aws.json` descrive i container da distribuire in ogni istanza Amazon EC2 nel cluster ECS. In questo caso, un'istanza Amazon EC2 viene anche considerata anche come istanza di container dell'host, perché ospita i container Docker. Il file di configurazione descrive anche i volumi di dati da creare sull'istanza di container host per il montaggio dei container Docker. Per ulteriori informazioni e un diagramma dei componenti in un ambiente Docker gestito da ECS su Elastic Beanstalk, consulta la sezione Panoramica della piattaforma Docker gestita da ECS precedente di questo capitolo.

Un file `Dockerrun.aws.json` può essere utilizzato da solo o compresso con un codice sorgente aggiuntivo in un singolo archivio. Il codice sorgente archiviato con un `Dockerrun.aws.json` viene distribuito in istanze di container di Amazon EC2 ed è accessibile dalla directory `/var/app/current/`.

Argomenti

- Dockerrun.aws.json v2
- Formato del volume
- Formato ARN del ruolo di esecuzione
- Formato della definizione del container
- Formato di autenticazione: utilizzo di immagini da un repository privato
- Esempio di Dockerrun.aws.json v2

`Dockerrun.aws.json` v2

Il file `Dockerrun.aws.json` include le seguenti sezioni:

AWSEBDockerrunVersione

Specifica il numero di versione come il valore 2 per gli ambienti ECS gestiti da Docker.

executionRoleArn

Speciifica i ruoli IAM per l'esecuzione delle attività per diversi scopi e servizi associati all'account. Affinché l'applicazione utilizzi le variabili di ambiente Elastic Beanstalk archiviate come segrete, devi specificare l'ARN di un ruolo di esecuzione dell'attività che concede le autorizzazioni richieste. Anche altri casi d'uso comuni possono richiedere questo parametro. Per ulteriori informazioni, consulta Formato ARN del ruolo di esecuzione.

volumi

Crea volumi dalle cartelle nell'istanza di container di Amazon EC2 o dal bundle di origine (distribuito in `/var/app/current`). Monta questi volumi nei percorsi all'interno dei container Docker utilizzando `mountPoints` nella sezione `containerDefinitions`.

containerDefinitions

Una serie di definizioni del container.

autenticazione (facoltativa)

Il percorso in Amazon S3 di un file `.dockercfg` che contiene i dati di autenticazione per un repository privato.

Le sezioni containerDefinitions e volumes di `Dockerrun.aws.json` utilizzano la stessa formattazione delle sezioni corrispondenti di un file di definizione delle attività di Amazon ECS. Per ulteriori informazioni sul formato delle definizioni di attività e un elenco completo dei parametri delle definizioni di attività, consulta Definizioni delle attività di Amazon ECS nella Guida per gli sviluppatori di Amazon Elastic Container Service.

Formato del volume

Il parametro volume crea volumi da altre cartelle nell'istanza di container Amazon EC2 o dal pacchetto sorgente (implementato su `/var/app/current`).

I volumi sono specificati nel seguente formato:

```
"volumes": [
    {
      "name": "volumename",
      "host": {
        "sourcePath": "/path/on/host/instance"
      }
    }
  ],
```

Monta questi volumi nei percorsi all'interno dei container Docker utilizzando `mountPoints` nella definizione del container.

Elastic Beanstalk configura volumi aggiuntivi per i log, uno per ogni container. Questi devono essere montati dai container Docker per scrivere log nell'istanza host.

Per maggiori dettagli, consulta il campo `mountPoints` nella sezione Formato di definizione del container di seguito.

Formato ARN del ruolo di esecuzione

Affinché l'applicazione utilizzi le variabili di ambiente Elastic Beanstalk archiviate come segreti, è necessario specificare un ruolo IAM per l'esecuzione delle attività. Il ruolo deve concedere al contenitore Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto utilizzando Gestione dei segreti AWS segreti o AWS Systems Manager parametri Parameter Store per fare riferimento a dati sensibili. Per istruzioni su come creare un ruolo IAM per l'esecuzione di attività con le autorizzazioni richieste per il tuo account, consulta il ruolo IAM di esecuzione delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

```
{
"AWSEBDockerrunVersion": 2,
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
```

Autorizzazioni aggiuntive richieste per la piattaforma Docker gestita da Amazon ECS

Concessione del profilo di istanza EC2 a ECS **`iam:PassRole`**

Affinché il profilo dell'istanza EC2 sia in grado di concedere questo ruolo al contenitore ECS, è necessario includere l'`iam:PassRole`autorizzazione dimostrata nell'esempio seguente. `iam:PassRole`Consente alle istanze EC2 di trasferire il ruolo di esecuzione dell'attività al contenitore ECS.

In questo esempio, limitiamo l'istanza EC2 a passare il ruolo solo al servizio ECS. Sebbene questa condizione non sia obbligatoria, la aggiungiamo per seguire le migliori pratiche volte a ridurre l'ambito dell'autorizzazione condivisa. Lo realizziamo con l'`Condition`elemento.

> **Note**
>
> Qualsiasi utilizzo del ruolo di esecuzione delle attività ECS IAM richiede l'`iam:PassRole`autorizzazione. Esistono altri casi d'uso comuni che richiedono il ruolo del servizio gestito di esecuzione delle attività ECS. Per ulteriori informazioni, consulta il ruolo IAM di esecuzione delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Example politica con autorizzazione **`iam:PassRole`**

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/ecs-task-execution-role"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        }
    ]
}
```

Concessione di segreti e parametri di accesso all'agente container Amazon ECS

Il ruolo IAM di esecuzione delle attività di Amazon ECS necessita anche delle autorizzazioni per accedere ai segreti e agli archivi dei parametri. Analogamente ai requisiti del ruolo del profilo dell'istanza EC2, l'agente contenitore ECS richiede l'autorizzazione per estrarre le risorse necessarie di Secrets Manager o Systems Manager. Per ulteriori informazioni, consulta le autorizzazioni di Secrets Manager o Systems Manager nella Amazon Elastic Container Service Developer Guide

Garantire l'accesso a segreti e parametri alle istanze Elastic Beanstalk EC2

Per supportare i segreti configurati come variabili di ambiente, dovrai anche aggiungere le autorizzazioni al profilo dell'istanza EC2. Per ulteriori informazioni, consultare Recupero di segreti e parametri nelle variabili di ambiente Elastic Beanstalk e Autorizzazioni IAM richieste per Secrets Manager.

Gli esempi seguenti combinano l'`iam:PassRole`esempio precedente con gli esempi forniti nella sezione di riferimento. Autorizzazioni IAM richieste per Secrets Manager Aggiungono le autorizzazioni

necessarie alle istanze EC2 per accedere Gestione dei segreti AWS e AWS Systems Manager archiviare per recuperare i segreti e i dati dei parametri per inizializzare le variabili di ambiente Elastic Beanstalk che sono state configurate per i segreti.

Example Politica di Secrets Manager combinata con **`iam:PassRole`** autorizzazione

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/ecs-task-execution-role"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:my-secret",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Example Politica di Systems Manager combinata con **`iam:PassRole`** autorizzazione

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/ecs-task-execution-role"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
               }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:parameter/my-parameter",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Formato della definizione del container

I seguenti esempi mostrano un sottoinsieme di parametri comunemente utilizzati nella sezione containerDefinitions. Sono disponibili ulteriori parametri opzionali.

La piattaforma Beanstalk crea una definizione di attività ECS, che include una definizione di container ECS. Beanstalk supporta un sottoinsieme di parametri per la definizione del container ECS. Per

ulteriori informazioni, consulta Definizioni di container nella Guida per gli sviluppatori di Amazon Elastic Container Service.

Un file `Dockerrun.aws.json` contiene un set di uno o più oggetti di definizione del container con i seguenti campi:

name

Il nome del container. Per informazioni sulla lunghezza massima e i caratteri consentiti, vedi Parametri della definizione del container standard.

image

Il nome di un'immagine Docker in un archivio Docker online da cui stai creando un container Docker. Nota queste convenzioni:

- Le immagini in repository ufficiali su Docker Hub utilizzano un singolo nome (ad esempio `ubuntu` o `mongo`).
- Le immagini in altri repository su Docker Hub vengono qualificate con un nome di organizzazione (ad esempi, `amazon/amazon-ecs-agent`.
- Le immagini in altri archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

environment

Un set di variabili di ambiente da passare al container.

Ad esempio, la seguente voce definisce una variabile di ambiente con il nome **`Container`** e il valore **`PHP`**:

```
"environment": [
  {
    "name": "Container",
    "value": "PHP"
  }
],
```

essential

True se l'attività deve interrompersi in caso di esito negativo del container. I container non essenziali possono arrestarsi o bloccarsi senza influire sul resto dei container sull'istanza.

memory

Quantità di memoria sull'istanza di container da riservare per il container. Specifica un numero intero diverso da zero per uno o entrambi i parametri `memory` e `memoryReservation` nelle definizioni del container.

memoryReservation

Il limite flessibile (in MiB) della memoria da prenotare per il container. Specifica un numero intero diverso da zero per uno o entrambi i parametri `memory` e `memoryReservation` nelle definizioni del container.

mountPoints

I volumi dell'istanza di container di Amazon EC2 da montare e la posizione sul file system del container Docker in cui montarli. Quando monti volumi che includono il contenuto dell'applicazione, il container è in grado di leggere i dati caricati nel bundle di origine. Quando monti volumi di log per la scrittura di dati di log, Elastic Beanstalk è in grado di raccogliere dati di log da questi volumi.

Elastic Beanstalk crea volumi di log sull'istanza di container, uno per ogni container Docker, in `/var/log/containers/`*`containername`*. Questi volumi sono denominati `awseb-logs-`*`containername`* e devono essere montati nel percorso all'interno della struttura del file container in cui vengono scritti i log.

Ad esempio, il seguente punto di montaggio mappa la posizione del log nginx nel container al volume generato da Elastic Beanstalk per il container `nginx-proxy`.

```
{
  "sourceVolume": "awseb-logs-nginx-proxy",
  "containerPath": "/var/log/nginx"
}
```

portMappings

Mappa porte di rete sul container alle porte sull'host.

links

Elenco dei container a cui effettuare il collegamento. I container collegati sono in grado di rilevarsi reciprocamente e comunicare in modo sicuro.

volumesFrom

Consente di montare tutti i volumi da un container diverso. Ad esempio, per montare i volumi da un container denominato `web`:

```
"volumesFrom": [
  {
    "sourceContainer": "web"
  }
],
```

Formato di autenticazione: utilizzo di immagini da un repository privato

La sezione `authentication` contiene i dati di autenticazione per un repository privato. Questa voce è facoltativa.

Aggiungi le informazioni sul bucket Amazon S3 che contiene il file di autenticazione al parametro `authentication` del file `Dockerrun.aws.json`. Assicurati che il parametro `authentication` contenga una chiave e un bucket Amazon S3 validi. Il bucket Amazon S3 deve essere ospitato nella stessa regione dell'ambiente che lo sta utilizzando. Elastic Beanstalk non scaricherà file dai bucket Amazon S3 ospitati in altre regioni.

Utilizza il seguente formato:

```
"authentication": {
    "bucket": "amzn-s3-demo-bucket",
    "key": "mydockercfg"
  },
```

Per informazioni sulla generazione e sul caricamento del file di autenticazione, vedi Autenticazione con archivi di immagini.

Esempio di Dockerrun.aws.json v2

Il seguente snippet è un esempio che illustra la sintassi del file `Dockerrun.aws.json` per un'istanza con due container.

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
```

```
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        }
      ]
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
```

```
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-proxy-conf",
          "containerPath": "/etc/nginx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-logs-nginx-proxy",
          "containerPath": "/var/log/nginx"
        }
      ]
    }
  ]
}
```

Politica gestita dai container e ruolo dell'istanza EC2

Quando crei un ambiente nella console Elastic Beanstalk, ti viene richiesto di creare un profilo di istanza predefinito che includa la policy gestita. `AWSElasticBeanstalkMulticontainerDocker` Quindi, inizialmente, il profilo di istanza EC2 predefinito dovrebbe includere questa policy gestita. Se il tuo ambiente utilizza un ruolo di profilo di istanza EC2 personalizzato anziché quello predefinito, assicurati che la policy gestita `AWSElasticBeanstalkMulticontainerDocker` sia allegata in modo da mantenere le autorizzazioni necessarie per la gestione dei contenitori. up-to-date

Elastic Beanstalk utilizza un'AMI ottimizzata per Amazon ECS con un agente del container Amazon ECS che viene eseguito in un container Docker. L'agente comunica con Amazon ECS per coordinare le distribuzioni dei container. Per comunicare con Amazon ECS, ogni istanza Amazon EC2 deve disporre delle autorizzazioni IAM corrispondenti, specificate in questa policy gestita. Consulta la AWSElasticBeanstalkMulticontainerDocker AWSManaged Policy Reference Guide per visualizzare queste autorizzazioni.

Se utilizzi variabili di ambiente Elastic Beanstalk configurate per accedere a segreti o parametri Gestione dei segreti AWS archiviati AWS Systems Manager nel nostro Parameter Store, devi

personalizzare il profilo dell'istanza EC2 con autorizzazioni aggiuntive. Per ulteriori informazioni, consulta Formato ARN del ruolo di esecuzione.

Utilizzo di più listener Elastic Load Balancing

È possibile configurare più ascoltatori Elastic Load Balancing su un ambiente Docker gestito da ECS per supportare il traffico in entrata per i proxy o altri servizi che non girano sulla porta HTTP predefinita.

Crea una cartella `.ebextensions` nel bundle di origine e aggiungi un file con un'estensione file `.config`. L'esempio seguente mostra un file di configurazione che crea un listener Elastic Load Balancing sulla porta 8080.

`.ebextensions/elb-listener.config`

```
option_settings:
  aws:elb:listener:8080:
    ListenerProtocol: HTTP
    InstanceProtocol: HTTP
    InstancePort: 8080
```

Se l'ambiente è in esecuzione in un Amazon Virtual Private Cloud (Amazon VPC) personalizzato che hai creato, Elastic Beanstalk si occupa del resto. In un VPC predefinito, è necessario configurare il gruppo di sicurezza dell'istanza per consentire l'ingresso dal load balancer. Aggiungi un secondo file di configurazione che aggiunge una regola in ingresso per il gruppo di sicurezza:

`.ebextensions/elb-ingress.config`

```
Resources:
  port8080SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 8080
      FromPort: 8080
      SourceSecurityGroupName: { "Fn::GetAtt": ["AWSEBLoadBalancer",
 "SourceSecurityGroup.GroupName"] }
```

Per ulteriori informazioni sul formato del file di configurazione consulta Aggiunta e personalizzazione delle risorse dell'ambiente Elastic Beanstalk e Impostazioni opzioni.

Oltre ad aggiungere un ascoltatore alla configurazione di Elastic Load Balancing e ad aprire una porta nel gruppo di sicurezza, è necessario mappare la porta sull'istanza host a una porta sul container Docker nella sezione `containerDefinitions` del file `Dockerrun.aws.json` v2. Di seguito viene riportato un estratto di esempio:

```
"portMappings": [
  {
    "hostPort": 8080,
    "containerPort": 8080
  }
]
```

Per ulteriori dettagli sul formato di file v2, consulta`Dockerrun.aws.json v2`. `Dockerrun.aws.json`

Creazione di un ambiente Docker gestito da ECS con la console Elastic Beanstalk

Questo tutorial descrive in dettaglio la configurazione del contenitore e la preparazione del codice sorgente per un ambiente Docker gestito da ECS che utilizza due contenitori.

I container, un'applicazione PHP e un proxy nginx, vengono eseguiti affiancati su ciascuna delle istanze Amazon Elastic Compute Cloud (Amazon EC2) in un ambiente Elastic Beanstalk. Dopo aver creato l'ambiente e aver verificato che le applicazioni sono in esecuzione, puoi connetterti a un'istanza di container per vedere come interagiscono.

Sections

- Definizione di contenitori Docker gestiti da ECS
- Aggiunta di contenuto
- Distribuzione in Elastic Beanstalk
- Connessione a un'istanza di container
- Esame dell'agente del container Amazon ECS

Definizione di contenitori Docker gestiti da ECS

La prima fase per la creazione di un nuovo ambiente Docker consiste nel creare una directory per i dati dell'applicazione. Questa cartella può essere posizionata ovunque nel computer locale e

avere qualsiasi nome a scelta. Oltre a un file di configurazione del container, la cartella includerà il contenuto che caricherai in Elastic Beanstalk e distribuirai nell'ambiente in uso.

> **Note**
>
> Tutto il codice di questo tutorial è disponibile nel repository awslabs all'indirizzo. GitHub https://github.com/awslabs/eb-docker-nginx-proxy

Il file utilizzato da Elastic Beanstalk per configurare i contenitori su un'istanza Amazon EC2 è un file di testo in formato JSON denominato v2. `Dockerrun.aws.json` Le versioni della piattaforma Docker gestita da ECS utilizzano il formato versione 2 di questo file. Questo formato può essere utilizzato solo con la piattaforma Docker gestita ECS, in quanto differisce notevolmente dalle altre versioni dei file di configurazione che supportano i rami della piattaforma Docker che non sono gestiti da ECS.

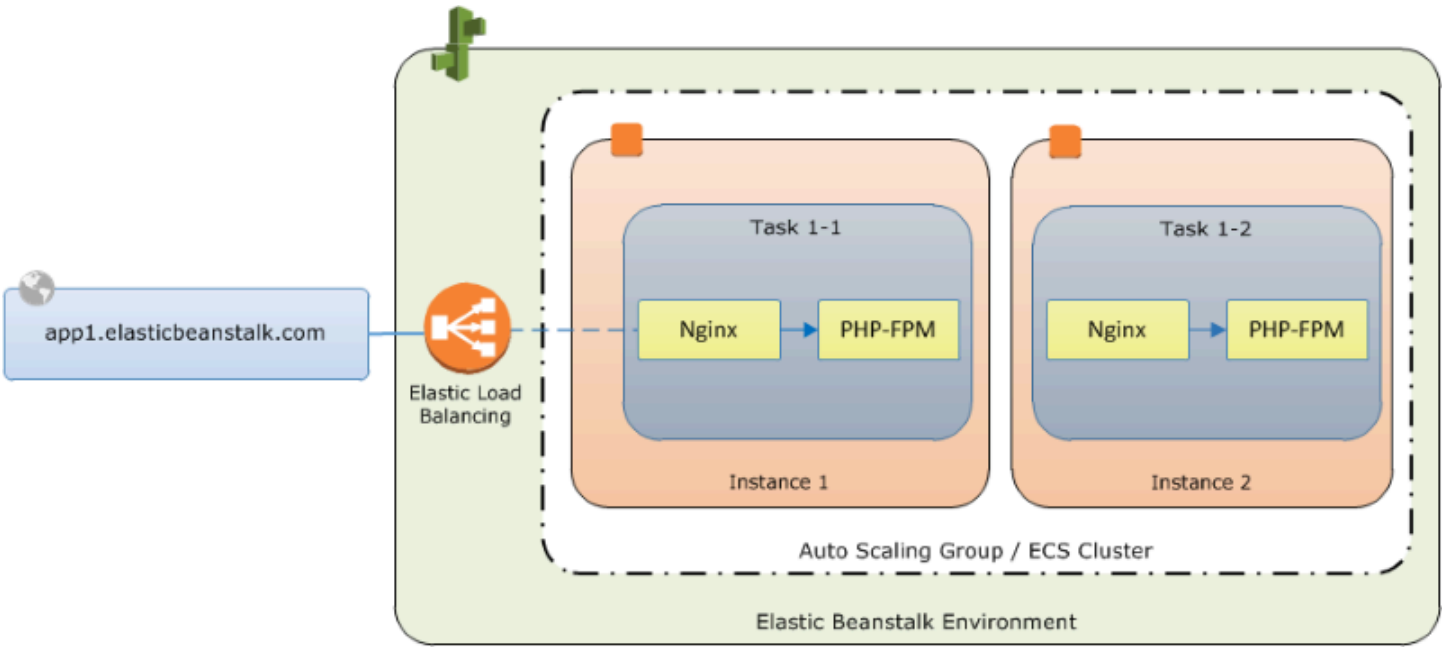
Crea un file di testo `Dockerrun.aws.json` v2 con questo nome nella radice dell'applicazione e aggiungi il testo seguente:

```
{
  "AWSEBDockerrunVersion": 2,
  "volumes": [
    {
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
```

```
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        }
      ]
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app",
          "containerPath": "/var/www/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-proxy-conf",
          "containerPath": "/etc/nginx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-logs-nginx-proxy",
          "containerPath": "/var/log/nginx"
        }
      ]
    }
  ]
}
```

In questa configurazione di esempio sono definiti due container, un sito Web PHP con un proxy nginx. Questi due container verranno eseguiti affiancati nei container Docker su ogni istanza nell'ambiente Elastic Beanstalk, con accesso al contenuto condiviso (il contenuto del sito Web) dai

volumi sull'istanza host, anch'essi definiti in questo file. I container stessi vengono creati da immagini ospitate in repository ufficiali su Docker Hub. L'ambiente risultante avrà il seguente aspetto:



I volumi definiti nella configurazione corrispondono al contenuto che creerai successivamente e caricherai come parte del tuo bundle di origine dell'applicazione. I container accedono al contenuto sull'host montando i volumi nella sezione `mountPoints` delle definizioni del container.

Per ulteriori informazioni sul formato della v2 e sui relativi parametri, consulta. `Dockerrun.aws.json` Formato della definizione del container

Aggiunta di contenuto

A questo punto puoi aggiungere al sito PHP contenuto da visualizzare ai visitatori e un file di configurazione per il proxy nginx.

php-app/index.php

```
<h1>Hello World!!!</h1>
<h3>PHP Version <pre><?= phpversion()?></pre></h3>
```

php-app/static.html

```
<h1>Hello World!</h1>
<h3>This is a static HTML page.</h3>
```

proxy/conf.d/default.conf

```
server {
  listen 80;
  server_name localhost;
  root /var/www/html;

  index index.php;

  location ~ [^/]\.php(/|$) {
    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
    if (!-f $document_root$fastcgi_script_name) {
      return 404;
    }

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;
    fastcgi_param PATH_TRANSLATED $document_root$fastcgi_path_info;

    fastcgi_pass php-app:9000;
    fastcgi_index index.php;
  }
}
```

Distribuzione in Elastic Beanstalk

La cartella dell'applicazione ora contiene i file seguenti:

```
### Dockerrun.aws.json
### php-app
#   ### index.php
#   ### static.html
### proxy
    ### conf.d
        ### default.conf
```

Questo è tutto ciò di cui hai bisogno per creare l'ambiente Elastic Beanstalk. Crea un archivio `.zip` dei file e delle cartelle sopra indicati (senza includere la cartella di progetto di primo livello). Per creare l'archivio in Esplora risorse, seleziona il contenuto della cartella di progetto, fai clic con il pulsante destro del mouse, scegli Send to (Invia a), quindi fai clic su Compressed (zipped) Folder (Cartella compressa (zip).

> **Note**
>
> Per ulteriori informazioni sulla struttura dei file richiesta e istruzioni sulla creazione di archivi in altri ambienti, vedi Crea un bundle di sorgenti dell'applicazione Elastic Beanstalk

Carica quindi il bundle di origine in Elastic Beanstalk e crea un ambiente. Per Platform (Piattaforma), selezionare Docker. Per Platform branch, seleziona ECS in esecuzione su Amazon Linux 2023 a 64 bit.

Per avviare un ambiente (console)

1. Apri la console Elastic Beanstalk con questo link preconfigurato: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
2. Per Platform (Piattaforma), selezionare la piattaforma e il ramo della piattaforma corrispondenti al linguaggio utilizzato dall'applicazione o la piattaforma Docker per le applicazioni basate su container.
3. Per Application code (Codice applicazione), scegliere Carica il tuo codice.
4. Selezionare Local file (File locale), quindi Browse (Sfoglia) e aprire il bundle di origine.
5. Selezionare Review and launch (Controlla e avvia).
6. Controlla le impostazioni disponibili, quindi seleziona Create app (Crea applicazione).

La console Elastic Beanstalk ti reindirizza al pannello di controllo di gestione per il nuovo ambiente. In questa schermata viene mostrato lo stato dell'ambiente e gli eventi generati dal servizio Elastic Beanstalk. Quando lo stato è Green, fai clic sull'URL accanto al nome dell'ambiente per visualizzare il tuo nuovo sito Web.

Connessione a un'istanza di container

È necessario connettersi a un'istanza Amazon EC2 nell'ambiente Elastic Beanstalk per vedere alcune delle parti mobili in azione.

Il modo più semplice per connettersi a un'istanza nel proprio ambiente è utilizzando l'interfaccia a riga di comando EB. Per utilizzarla, installare la CLI EB, se necessario. È inoltre necessario configurare l'ambiente con una coppia di chiavi SSH Amazon EC2. Usa la pagina di configurazione di sicurezza

della console o la CLI EB eb init. Per connettersi a un'istanza di ambiente, utilizzare l'interfaccia a riga EB eb ssh.

Ora che sei connesso all'istanza Amazon EC2 per l'hosting dei container Docker, puoi vedere se tutto funziona. Esegui `ls` su `/var/app/current`:

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/app/current
Dockerrun.aws.json  php-app  proxy
```

Questa directory contiene i file dal bundle di origine caricato in Elastic Beanstalk durante la creazione dell'ambiente.

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/log/containers
nginx-proxy    nginx-proxy-4ba868dbb7f3-stdouterr.log
php-app        php-app-dcc3b3c8522c-stdouterr.log       rotated
```

Questo è il punto in cui vengono creati i log sull'istanza di container e raccolte da Elastic Beanstalk. Elastic Beanstalk crea un volume in questa directory per ogni container, che monti nella posizione del container in cui vengono scritti i log.

È anche possibile esaminare il Docker per vedere i container in esecuzione con `docker ps`.

```
[ec2-user@ip-10-0-0-117 ~]$ sudo docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED
  STATUS                  PORTS                                NAMES

4ba868dbb7f3   nginx                            "/docker-entrypoint.…"   4 minutes ago
   Up 4 minutes            0.0.0.0:80->80/tcp, :::80->80/tcp   ecs-awseb-Tutorials-env-
dc2aywfjwg-1-nginx-proxy-acca84ef87c4aca15400
dcc3b3c8522c   php:fpm                          "docker-php-entrypoi…"   4 minutes ago
   Up 4 minutes            9000/tcp                             ecs-awseb-Tutorials-env-
dc2aywfjwg-1-php-app-b8d38ae288b7b09e8101
d9367c0baad6   amazon/amazon-ecs-agent:latest   "/agent"                 5 minutes ago
  Up 5 minutes (healthy)                                        ecs-agent
```

Questo mostra i due container distribuiti in esecuzione e l'agente del container Amazon ECS che ha coordinato la distribuzione.

Esame dell'agente del container Amazon ECS

Le istanze Amazon EC2 in un ambiente Docker gestito da ECS su Elastic Beanstalk eseguono un processo agente in un container Docker. Questo agente si connette al servizio Amazon ECS per coordinare le distribuzioni del container. Tali distribuzioni vengono eseguite come attività in Amazon ECS, configurate nei file di definizione delle attività. Elastic Beanstalk crea questi file in base al file `Dockerrun.aws.json` caricato in un bundle di origine.

Verifica lo stato dell'agente del container con una richiesta get HTTP a `http://localhost:51678/v1/metadata`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/metadata
{
  "Cluster":"awseb-Tutorials-env-dc2aywfjwg",
  "ContainerInstanceArn":"arn:aws:ecs:us-west-2:123456789012:container-instance/awseb-
Tutorials-env-dc2aywfjwg/db7be5215cd74658aacfcb292a6b944f",
  "Version":"Amazon ECS Agent - v1.57.1 (089b7b64)"
}
```

Questa struttura mostra il nome del cluster Amazon ECS e l'ARN (Amazon Resource Name) dell'istanza di cluster (l'istanza Amazon EC2 a cui sei connesso).

Per ulteriori informazioni, effettua una richiesta HTTP a `http://localhost:51678/v1/tasks`:

```
[ec2-user@ip-10-0-0-117 ~]$ curl http://localhost:51678/v1/tasks
{
   "Tasks":[
      {
         "Arn":"arn:aws:ecs:us-west-2:123456789012:task/awseb-Tutorials-env-dc2aywfjwg/
bbde7ebe1d4e4537ab1336340150a6d6",
         "DesiredStatus":"RUNNING",
         "KnownStatus":"RUNNING",
         "Family":"awseb-Tutorials-env-dc2aywfjwg",
         "Version":"1",
         "Containers":[
            {

 "DockerId":"dcc3b3c8522cb9510b7359689163814c0f1453b36b237204a3fd7a0b445d2ea6",
               "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-php-app-
b8d38ae288b7b09e8101",
               "Name":"php-app",
               "Volumes":[
```

```
                  {
                     "Source":"/var/app/current/php-app",
                     "Destination":"/var/www/html"
                  }
               ]
            },
            {

 "DockerId":"4ba868dbb7f3fb3328b8afeb2cb6cf03e3cb1cdd5b109e470f767d50b2c3e303",
               "DockerName":"ecs-awseb-Tutorials-env-dc2aywfjwg-1-nginx-proxy-
acca84ef87c4aca15400",
               "Name":"nginx-proxy",
               "Ports":[
                  {
                     "ContainerPort":80,
                     "Protocol":"tcp",
                     "HostPort":80
                  },
                  {
                     "ContainerPort":80,
                     "Protocol":"tcp",
                     "HostPort":80
                  }
               ],
               "Volumes":[
                  {
                     "Source":"/var/app/current/php-app",
                     "Destination":"/var/www/html"
                  },
                  {
                     "Source":"/var/log/containers/nginx-proxy",
                     "Destination":"/var/log/nginx"
                  },
                  {
                     "Source":"/var/app/current/proxy/conf.d",
                     "Destination":"/etc/nginx/conf.d"
                  }
               ]
            }
         ]
      }
   ]
}
```

Questa struttura descrive l'attività che viene eseguita per distribuire i due container Docker dal progetto di esempio di questo tutorial. Sono visualizzate le seguenti informazioni:

- KnownStatus— Lo stato indica che i contenitori sono ancora attivi. `RUNNING`
- Famiglia: Il nome della definizione dell'attività creata da Elastic Beanstalk in base a `Dockerrun.aws.json`.
- Versione: versione della definizione dell'attività. Viene incrementata ogni volta che viene aggiornato il file di definizione dell'attività.
- Container: informazioni sui container in esecuzione sull'istanza.

Sono disponibili ancora altre informazioni dal servizio Amazon ECS stesso, che è possibile chiamare utilizzando AWS Command Line Interface. Per istruzioni sull' AWS CLI uso di Amazon ECS e informazioni generali su Amazon ECS, consulta la Amazon ECS User Guide.

Migrazione dell'applicazione Elastic Beanstalk da Docker multi-container gestito da ECS a ECS su Amazon Linux 2023 AL1

> **Note**
>
> Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. .

Questo argomento ti guida nella migrazione delle tue applicazioni dal ramo di piattaforma dismesso Multi-container Docker in esecuzione su Amazon Linux a 64 bit a ECS Running su 64bit 023. AL2 Questo ramo della piattaforma di destinazione è attuale e supportato. Come il precedente ramo Multi-container Docker, il nuovo AL1 ramo della AL2piattaforma ECS 023 utilizza Amazon ECS per coordinare la distribuzione di più contenitori Docker in un cluster Amazon ECS in un ambiente Elastic Beanstalk. Il nuovo ramo della piattaforma ECS AL2 023 supporta tutte le funzionalità del precedente ramo della piattaforma Multi-container Docker. AL1 Inoltre, è supportato lo stesso file v2 `Dockerrun.aws.json`.

Sections

- Migrazione con la console Elastic Beanstalk
- Esegui la migrazione con AWS CLI

Migrazione con la console Elastic Beanstalk

Per migrare utilizzando la console Elastic Beanstalk, distribuisci lo stesso codice sorgente in un nuovo ambiente basato sul ramo della piattaforma ECS Running on 023. AL2 Non sono richieste modifiche al codice sorgente.

Per migrare alla ramificazione della piattaforma ECS in esecuzione su Amazon Linux 2023

1. Utilizzando l'origine dell'applicazione già distribuita nel vecchio ambiente, creare un bundle di origine dell'applicazione. È possibile utilizzare lo stesso bundle di origine dell'applicazione e lo stesso file v2 `Dockerrun.aws.json`.
2. Creare un nuovo ambiente utilizzando la ramificazione della piattaforma ECS in esecuzione su Amazon Linux 2023. Utilizzare il bundle di origine della fase precedente per Codice dell'applicazione. Per ulteriori passaggi dettagliati, consulta Distribuzione in Elastic Beanstalk nel Tutorial Docker gestito da ECS all'inizio di questo capitolo.

Esegui la migrazione con AWS CLI

Hai anche la possibilità di utilizzare AWS Command Line Interface (AWS CLI) per migrare il tuo ambiente Docker multi-container Amazon Linux Docker esistente al nuovo AL2 ramo della piattaforma ECS 023. In questo caso non è necessario che crei un nuovo ambiente o ridistribuisci il codice sorgente. Devi solo eseguire il comando update-environment. AWS CLI Eseguirà un aggiornamento della piattaforma per migrare il tuo ambiente esistente alla ramificazione della piattaforma Amazon Linux 2023 ECS.

Utilizza la seguente sintassi per migrare il tuo ambiente al nuovo ramo della piattaforma.

```
aws elasticbeanstalk update-environment \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 version running ECS" \
--region my-region
```

Di seguito è riportato un esempio del comando per migrare l'ambiente beta-101 alla versione 3.0.0 della ramificazione della piattaforma Amazon Linux 2023 ECS nella regione us-east-1.

```
aws elasticbeanstalk update-environment \
--environment-name beta-101 \
--solution-stack-name "64bit Amazon Linux 2023 v4.0.0 running ECS" \
--region us-east-1
```

Il parametro `solution-stack-name` fornisce il ramo della piattaforma e la sua versione. Utilizza la versione più recente del ramo della piattaforma specificando il corretto nome dello stack della soluzione. La versione di ogni ramo della piattaforma è inclusa nel nome dello stack della soluzione come mostrato nell'esempio precedente. Per un elenco degli stack di soluzioni più recenti per la piattaforma Docker, consulta Piattaforme supportate nella guida Piattaforme AWS Elastic Beanstalk .

> **Note**
>
> Il list-available-solution-stackscomando fornisce un elenco delle versioni della piattaforma disponibili per il tuo account in una AWS regione.
>
> ```
> aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --query
> SolutionStacks
> ```

Per ulteriori informazioni su AWS CLI, consulta la Guida AWS Command Line Interface per l'utente. Per ulteriori informazioni sui AWS CLI comandi per Elastic Beanstalk, AWS CLI consulta il Command Reference for Elastic Beanstalk.

Autenticazione con archivi di immagini

Questo argomento descrive come autenticarsi negli archivi di immagini online con Elastic Beanstalk. Per gli archivi privati, Elastic Beanstalk deve autenticarsi prima di poter estrarre e distribuire le immagini. Per Amazon ECR Public, l'autenticazione è facoltativa ma offre limiti di velocità più elevati e una maggiore affidabilità.

Utilizzo di immagini da un repository Amazon ECR

Puoi archiviare le tue immagini Docker personalizzate AWS con Amazon Elastic Container Registry (Amazon ECR).

Quando memorizzi le immagini Docker in Amazon ECR, Elastic Beanstalk si autentica automaticamente nel registro Amazon ECR con il profilo dell'istanza del tuo ambiente. Pertanto, dovrai fornire alle tue istanze l'autorizzazione ad accedere alle immagini nel tuo repository Amazon ECR. A tale scopo, aggiungi le autorizzazioni al profilo dell'istanza del tuo ambiente allegando la policy EC2 ContainerRegistryReadOnly gestita da Amazon al profilo dell'istanza. Ciò fornisce l'accesso in sola lettura a tutti i repository Amazon ECR del tuo account. Hai anche la possibilità

di accedere solo a un singolo repository utilizzando il seguente modello per creare una politica personalizzata:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEbAuth",
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "AllowPull",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:ecr:us-east-2:111122223333:repository/repository-name"
            ],
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:ListImages",
                "ecr:BatchGetImage"
            ]
        }
    ]
}
```

Sostituisci l'Amazon Resource Name (ARN) nella policy sopra indicata con l'ARN del repository.

Dovrai specificare le informazioni sull'immagine nel tuo `Dockerrun.aws.json` file. La configurazione sarà diversa a seconda della piattaforma utilizzata.

Per la piattaforma Docker gestita da ECS, usa la `image` chiave in un oggetto di definizione del contenitore:

```
"containerDefinitions": [
        {
        "name": "my-image",
        "image": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
```

Per la piattaforma Docker, fai riferimento all'immagine tramite URL. L'URL rientra nella `Image` definizione del `Dockerrun.aws.json` file:

```
  "Image": {
      "Name": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
      "Update": "true"
    },
```

Usando Gestione dei segreti AWS

Configura Elastic Beanstalk per l'autenticazione con il tuo repository privato prima della distribuzione per consentire l'accesso alle immagini del contenitore.

Questo approccio utilizza la fase di precompilazione del processo di distribuzione di Elastic Beanstalk con due componenti:

- ebextensions per definire le variabili di ambiente che memorizzano le credenziali del repository
- script platform hook da eseguire prima di estrarre le immagini docker login

Gli script hook recuperano un nome utente e una password dalle variabili di ambiente che vengono popolate da un singolo Gestione dei segreti AWS segreto in formato JSON. Questa funzionalità richiede le piattaforme Elastic Beanstalk Docker e ECS managed Docker rilasciate a partire dal 13 gennaio 2026. Per ulteriori dettagli, consulta i segreti dell'ambiente.

Per configurare Elastic Beanstalk per l'autenticazione nel tuo repository privato con Gestione dei segreti AWS

> **Note**
>
> Prima di procedere, assicurati di aver impostato le credenziali e configurato le autorizzazioni IAM necessarie. Gestione dei segreti AWS Per i dettagli, consulta Prerequisiti per configurare i segreti come variabili di ambiente.

1. Crea la seguente struttura di cartelle per il tuo progetto:

```
### .ebextensions
#   ### env.config
### .platform
#   ### confighooks
#   #   ### prebuild
#   #       ### 01login.sh
#   ### hooks
#       ### prebuild
#           ### 01login.sh
### Dockerfile
```

2. Utilizzatela Gestione dei segreti AWSper salvare le credenziali del vostro repository privato come segreto in formato JSON.

```
aws secretsmanager create-secret --name repo-credentials \
    --secret-string '{"username":"myuser","password":"mypassword"}'
```

3. Creare il file `env.config` seguente e posizionarlo nella directory `.ebextensions`, come mostrato nella struttura di directory precedente. Questa configurazione utilizza lo spazio dei aws: elasticbeanstalk:application:environmentsecrets nomi con estrazione di chiavi JSON per inizializzare le variabili di ambiente Elastic `USER` Beanstalk dai `PASSWD` singoli campi del segreto.

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    USER: arn:aws:secretsmanager:us-east-1:111122223333:secret:repo-credentials-
AbCd12:username
```

```
    PASSWD: arn:aws:secretsmanager:us-east-1:111122223333:secret:repo-credentials-
AbCd12:password
```

4. Crea il seguente file di `01login.sh` script e posizionalo nelle seguenti posizioni (mostrate anche nella struttura di directory precedente):

   - `.platform/confighooks/prebuild/01login.sh`
   - `.platform/hooks/prebuild/01login.sh`

```
#!/bin/bash
echo $PASSWD | docker login -u $USER --password-stdin
```

   Lo `01login.sh` script utilizza le variabili di ambiente configurate nel passaggio 3 e passa la password a docker login via`stdin`. Per ulteriori informazioni sull'autenticazione Docker, consulta docker login nella documentazione Docker.

   > **Note**
   >
   > - La piattaforma Docker gestita da ECS utilizza la sintassi ECS nativa per fare riferimento ai segreti. Per ulteriori informazioni, consulta Passare i segreti di Secrets Manager tramite le variabili di ambiente Amazon ECS nella Amazon Elastic Container Service Developer Guide.
   > - Per ulteriori informazioni sugli hook della piattaforma, consulta Extending the section called “Hook della piattaforma” Elastic Beanstalk Linux platforms.

Una volta configurata l'autenticazione, Elastic Beanstalk può estrarre e distribuire immagini dal tuo repository privato.

Utilizzo del file **`Dockerrun.aws.json`**

In questa sezione viene descritto un altro approccio per l'autenticazione di Elastic Beanstalk in un repository privato. Con questo approccio, si genera un file di autenticazione con il comando Docker e quindi si carica il file di autenticazione in un bucket Amazon S3. È inoltre necessario includere le informazioni relative al bucket nel file `Dockerrun.aws.json`.

Per generare e fornire un file di autenticazione per Elastic Beanstalk

1. Genera un file di autenticazione con il comando docker login. Per repository su Docker Hub, esegui docker login:

```
$ docker login
```

Per gli altri registri, includi l'URL del server di registro:

```
$ docker login registry-server-url
```

> **Note**
>
> Se l'ambiente Elastic Beanstalk utilizza la versione della piattaforma Docker su AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni pertinenti in the section called "Configurazione Docker su AMI Amazon Linux (precedente ad Amazon Linux 2)".

Per ulteriori informazioni sul file di autenticazione, vedi Memorizzare le immagini in Docker Hub e login docker sul sito Web Docker.

2. Carica una copia del file di autenticazione denominato `.dockercfg` in un bucket Amazon S3 protetto.

   - Il bucket Amazon S3 deve essere ospitato nello Regione AWS stesso ambiente che lo utilizza. Elastic Beanstalk non può scaricare file da un bucket Amazon S3 ospitato in altre regioni.
   - Concedi le autorizzazioni per l'operazione `s3:GetObject` al ruolo IAM nel profilo dell'istanza. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.

3. Includi le informazioni sul bucket Amazon S3 nel parametro `Authentication` nel file `Dockerrun.aws.json`.

   Il seguente esempio mostra l'uso di un file di autenticazione denominato `mydockercfg` in un bucket denominato `amzn-s3-demo-bucket` per l'utilizzo di un'immagine privata in un registro di terze parti. Per il numero di versione corretto per`AWSEBDockerrunVersion`, consulta la nota che segue l'esempio.

```
{
```

```
  "AWSEBDockerrunVersion": "version-no",
  "Authentication": {
    "Bucket": "amzn-s3-demo-bucket",
    "Key": "mydockercfg"
  },
  "Image": {
    "Name": "quay.io/johndoe/private-image",
    "Update": "true"
  },
  "Ports": [
    {
      "ContainerPort": "1234"
    }
  ],
  "Volumes": [
    {
      "HostDirectory": "/var/app/mydb",
      "ContainerDirectory": "/etc/mysql"
    }
  ],
  "Logging": "/var/log/nginx"
}
```

Versioni **`Dockerrun.aws.json`**

Il parametro `AWSEBDockerrunVersion` indica la versione del file `Dockerrun.aws.json`.

- Il Docker AL2 e AL2023 le piattaforme utilizzano le seguenti versioni del file.
  - `Dockerrun.aws.json v3`— ambienti che utilizzano Docker Compose.
  - `Dockerrun.aws.json v1`— ambienti che non utilizzano Docker Compose.
- ECS in esecuzione su Amazon Linux 2 e ECS in esecuzione su AL2023 utilizzano il `Dockerrun.aws.json v2` file. Anche la piattaforma ritirata ECS-The Multicontainer Docker Amazon Linux AMI AL1 () utilizzava questa stessa versione.

Quando Elastic Beanstalk sarà in grado di eseguire l'autenticazione con il registro online che ospita il repository privato, le immagini potranno essere distribuite ed estratte.

Utilizzo di immagini da Amazon ECR Public

Amazon ECR Public è un registro di container pubblico che ospita immagini Docker. Sebbene gli archivi pubblici di Amazon ECR siano accessibili al pubblico, l'autenticazione offre limiti di velocità più elevati e una migliore affidabilità per le distribuzioni.

> **Note**
>
> L'autenticazione pubblica di Amazon ECR non è supportata nelle regioni (`cn-*`) e nelle AWS GovCloud regioni (`us-gov-*`) della Cina. In queste regioni, Elastic Beanstalk utilizzerà pull non autenticati.

Per abilitare l'autenticazione pubblica di Amazon ECR, aggiungi le seguenti autorizzazioni al profilo dell'istanza del tuo ambiente. Per ulteriori informazioni sull'autenticazione pubblica di Amazon ECR, consulta l'autenticazione del registro in Amazon ECR public nella Amazon Elastic Container Registry Public User Guide:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
       {
          "Effect": "Allow",
          "Action": [
             "ecr-public:GetAuthorizationToken",
             "sts:GetServiceBearerToken"
          ],
          "Resource": "*"
       }
    ]
}
```

Una volta associate queste autorizzazioni al profilo dell'istanza, Elastic Beanstalk si autenticherà automaticamente con i registri pubblici di Amazon ECR. Puoi fare riferimento alle immagini pubbliche di Amazon ECR utilizzando il `public.ecr.aws/`*`registry-alias`*`/`*`repository-name:tag` formato standard nel tuo `Dockerrun.aws.json` file o Dockerfile.

Configurazione degli ambienti Elastic Beanstalk Docker

Questo capitolo spiega informazioni di configurazione aggiuntive per tutti i rami della piattaforma Docker supportati, incluso il ramo della piattaforma Docker gestito da ECS. A meno che non venga identificato un ramo o un componente specifico della piattaforma in una sezione, si applicano a tutti gli ambienti che eseguono piattaforme Docker supportate e Docker gestite da ECS.

> **Note**
>
> Se l'ambiente Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni aggiuntive presenti in the section called "Configurazione Docker su AMI Amazon Linux (precedente ad Amazon Linux 2)".

Sezioni

- Configurazione del software in ambienti Docker
- Riferimento alle variabili di ambiente nei container
- Utilizzo della funzione di interpolazione per le variabili di ambiente con Docker Compose
- Generazione di log per report sanitari avanzati con Docker Compose
- Registrazione personalizzata del contenitore Docker con Docker Compose
- Immagini Docker
- Configurazione degli aggiornamenti gestiti per gli ambienti Docker
- Spazio dei nomi di configurazione di Docker
- Configurazione Docker su AMI Amazon Linux (precedente ad Amazon Linux 2)

Configurazione del software in ambienti Docker

È possibile utilizzare la console Elastic Beanstalk per configurare il software in esecuzione sulle istanze dell'ambiente.

Per configurare l'ambiente Docker nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Apportare le modifiche necessarie alla configurazione.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Per informazioni sulla configurazione delle impostazioni software in qualsiasi ambiente, consulta the section called “Variabili di ambiente e impostazioni del software”. Le sezioni seguenti sono relative alle informazioni specifiche di Docker.

Opzioni del container

La sezione Container options (Opzioni container) contiene opzioni specifiche della piattaforma. Per gli ambienti Docker, consente di scegliere se l'ambiente include o meno il server proxy Nginx.

Ambienti con Docker Compose

Se si gestisce l'ambiente Docker con Docker Compose, Elastic Beanstalk presuppone che si esegua un server proxy come container. Pertanto, il valore predefinito è None (Nessuno) per l'impostazione del Proxy server (Server Proxy) e Elastic Beanstalk non fornisce una configurazione NGINX.

> **Note**
>
> Anche se si seleziona NGINX come server proxy, questa impostazione viene ignorata in un ambiente con Docker Compose. L'impostazione predefinita del Proxy server (Server Proxy) continua a essere None (Nessuno).

Poiché il proxy del server Web NGINX è disabilitato per Docker sulla piattaforma Amazon Linux 2 con Docker Compose, è necessario seguire le istruzioni per la generazione dei log per la creazione di report sullo stato avanzato. Per ulteriori informazioni, consulta Generazione di log per report sanitari avanzati con Docker Compose.

Proprietà dell'ambiente (variabili di ambiente)

È possibile utilizzare le proprietà di ambiente, note anche come variabili di ambiente, per passare valori, ad esempio endpoint, impostazioni di debug e altre informazioni all'applicazione. La sezione delle variabili di ambiente della console consente di specificare le variabili di ambiente sulle istanze

EC2 che eseguono l'applicazione. Le variabili di ambiente vengono passate all'applicazione come coppie chiave-valore.

Il codice dell'applicazione in esecuzione in un container può fare riferimento a una variabile di ambiente in base al nome e leggerne il valore. Il codice sorgente che legge queste variabili di ambiente varia in base al linguaggio di programmazione. È possibile trovare istruzioni per la lettura dei valori delle variabili di ambiente nei linguaggi di programmazione supportati dalle piattaforme gestite da Elastic Beanstalk nell'argomento della piattaforma corrispondente. Per un elenco di collegamenti a questi argomenti, consulta the section called “Variabili di ambiente e impostazioni del software”.

Segreti e parametri nelle variabili di ambiente Elastic Beanstalk

Elastic Beanstalk offre la possibilità di Gestione dei segreti AWS fare AWS Systems Manager riferimento e memorizzare i dati in variabili di ambiente. Si tratta di un'opzione sicura per consentire all'applicazione di accedere in modo nativo ai segreti e ai parametri archiviati da questi servizi senza dover gestire le chiamate API verso di essi. Le piattaforme Elastic Beanstalk Docker e Docker gestite da ECS devono essere una versione rilasciata a partire dal 26 marzo 2025 per supportare questa funzionalità. Per ulteriori informazioni sull'utilizzo delle variabili di ambiente per fare riferimento ai segreti, consulta. Recupero di segreti e parametri nelle variabili di ambiente Elastic Beanstalk

Ambienti con Docker Compose

Se si gestisce l'ambiente Docker con Docker Compose, è necessario creare alcune configurazioni aggiuntive per recuperare le variabili di ambiente nei container. Affinché gli eseguibili in esecuzione nel container accedano a queste variabili di ambiente, è necessario fare riferimento a tali variabili in `docker-compose.yml`. Per ulteriori informazioni, consulta Riferimento alle variabili di ambiente nei container.

Riferimento alle variabili di ambiente nei container

Se si utilizza lo strumento Docker sulla piattaforma Docker per Amazon Linux 2, Elastic Beanstalk genera un file di ambiente Docker Compose denominato `.env` nella directory principale del progetto dell'applicazione. Questo file memorizza le variabili di ambiente configurate per Elastic Beanstalk.

> **Note**
>
> Se si include un file `.env` nel bundle di applicazioni, Elastic Beanstalk non genererà un file `.env`.

Affinché un container possa fare riferimento alle variabili di ambiente definite in Elastic Beanstalk, è necessario seguire uno o entrambi questi approcci di configurazione.

- Aggiungi il file `.env` generato da Elastic Beanstalk all'opzione di configurazione `env_file` nel file `docker-compose.yml`.
- Definisci direttamente le variabili di ambiente nel file `docker-compose.yml`.

I seguenti file forniscono un esempio. Il file di esempio `docker-compose.yml` illustra entrambi gli approcci.

- Se definisci le proprietà di ambiente `DEBUG_LEVEL=1` e `LOG_LEVEL=error`, Elastic Beanstalk genera il seguente file `.env`:

```
DEBUG_LEVEL=1
LOG_LEVEL=error
```

- In questo file `docker-compose.yml`, l'opzione di configurazione `env_file` punta al file `.env` e definisce anche la variabile di ambiente `DEBUG=1` direttamente nel file `docker-compose.yml`.

```
services:
  web:
    build: .
    environment:
      - DEBUG=1
    env_file:
      - .env
```

> **Note**
>
> - Se si imposta la stessa variabile di ambiente in entrambi i file, la variabile definita nel file `docker-compose.yml` ha la priorità sulla variabile definita nel file `.env`.
> - Fare attenzione a non lasciare spazi tra il segno di uguale (=) e il valore assegnato alla variabile per evitare che gli spazi vengano aggiunti alla stringa.

Per ulteriori informazioni sulle variabili di ambiente in Docker Compose, vedere Variabili di ambiente in Compose

Utilizzo della funzione di interpolazione per le variabili di ambiente con Docker Compose

A partire dal rilascio della piattaforma del 28 luglio 2023, la ramificazione della piattaforma Docker Amazon Linux 2 offre la funzionalità di interpolazione Docker Compose. Con questa funzionalità, i valori in un file Compose possono essere impostati da variabili e interpolati in fase di runtime. Per ulteriori informazioni su questa funzionalità, consulta la pagina Interpolation sul sito Web della documentazione di Docker.

> **Important**
>
> Se desideri utilizzare questa funzionalità con le tue applicazioni, tieni presente che dovrai implementare un approccio che utilizzi gli hook della piattaforma.
> Ciò è necessario a causa di una mitigazione che abbiamo implementato nel motore della piattaforma. Questa mitigazione garantisce la compatibilità con le versioni precedenti per i clienti che non sono a conoscenza della nuova funzionalità di interpolazione e dispongono di applicazioni esistenti che utilizzano variabili di ambiente con il carattere $. Il motore della piattaforma aggiornato evita l'interpolazione per impostazione predefinita sostituendo il carattere $ con i caratteri $$.

Di seguito è riportato un esempio di script hook della piattaforma che è possibile configurare per abilitare l'uso della funzionalità di interpolazione.

```
#!/bin/bash

: '
example data format in .env file
key1=value1
key2=value2
'
envfile="/var/app/staging/.env"
tempfile=$(mktemp)

while IFS= read -r line; do
  # split each env var string at '='
  split_str=(${line//=/ })
  if [ ${#split_str[@]} -eq 2 ]; then
    # replace '$$' with '$'
    replaced_str=${split_str[1]//\$\$/\$}
```

```
    # update the value of env var using ${replaced_str}
    line="${split_str[0]}=${replaced_str}"
  fi
  # append the updated env var to the tempfile
  echo "${line}" #"${tempfile}"
done < "${envfile}"
# replace the original .env file with the tempfile
mv "${tempfile}" "${envfile}"
```

Posiziona gli hook della piattaforma in entrambe queste directory:

- `.platform/confighooks/predeploy/`
- `.platform/hooks/predeploy/`

Per ulteriori informazioni, consulta Hook della piattaforma nell'argomento Estensione delle piattaforme Linux di questa guida.

Generazione di log per report sanitari avanzati con Docker Compose

L'agente di integrità di Elastic Beanstalk fornisce parametri di integrità del sistema operativo e delle applicazioni per gli ambienti Elastic Beanstalk. Si basa sui formati di log del server Web che inoltrano le informazioni in un formato specifico.

Elastic Beanstalk presuppone l'esecuzione di un proxy del server Web come container. Di conseguenza, il proxy del server Web NGINX è disabilitato per gli ambienti Docker che eseguono Docker Compose. È necessario configurare il server per scrivere i log nel percorso e nel formato utilizzati dall'agente di integrità di Elastic Beanstalk. In questo modo è possibile utilizzare appieno i report sullo stato avanzato, anche se il proxy del server Web è disabilitato.

Per istruzioni su come eseguire questa operazione, consultare Configurazione dei log del server Web

Registrazione personalizzata del contenitore Docker con Docker Compose

Per risolvere in modo efficiente i problemi e monitorare i servizi containerizzati, è possibile richiedere i log delle istanze da Elastic Beanstalk tramite la console di gestione dell'ambiente o la CLI EB.
I registri delle istanze sono composti da log di bundle e log di coda, combinati e confezionati per consentire di visualizzare i registri e gli eventi recenti in modo efficiente e diretto.

Elastic Beanstalk crea directory di log sull'istanza di container, una per ogni servizio definito nel file `docker-compose.yml`, in `/var/log/eb-docker/containers/<service name>`. Se si utilizza

la funzionalità Docker Compose sulla piattaforma Docker su Amazon Linux 2, è possibile montare queste directory nella posizione all'interno della struttura del file container in cui vengono scritti i log. Quando si montano le directory di log per la scrittura dei dati di log, Elastic Beanstalk può raccogliere i dati di log da queste directory.

Per configurare i file di registro del servizio in modo che siano file di coda recuperabili e registri raggruppati

1. Modificare il file `docker-compose.yml`.
2. Sotto la chiave `volumes` per il servizio aggiungere un mount bind come il seguente:

   `"${EB_LOG_BASE_DIR}/<service name>:<log directory inside container>`

   Nel file di esempio `docker-compose.yml` seguente:

   - `nginx-proxy` è *<service name>*
   - `/var/log/nginx` è *<log directory inside container>*

```
services:
  nginx-proxy:
    image: "nginx"
    volumes:
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
```

- La directory `var/log/nginx` contiene i log per il servizio nginx-proxy nel container e verrà mappata alla directory `/var/log/eb-docker/containers/nginx-proxy` sull'host.
- Tutti i log in questa directory sono ora recuperabili come log di bundle e log di coda attraverso la funzionalità dei log di istanza della richiesta di Elastic Beanstalk.

> **Note**
>
> - ${EB_LOG_BASE_DIR} è una variabile di ambiente impostata da Elastic Beanstalk con il valore `/var/log/eb-docker/containers`.

- Elastic Beanstalk crea automaticamente la directory `/var/log/eb-docker/containers/<service name>` per ogni servizio nel file `docker-compose.yml`.

Immagini Docker

Le versioni della piattaforma Docker e ECS gestita da Docker per Elastic Beanstalk supportano l'uso di immagini Docker memorizzate in un repository di immagini online pubblico o privato.

Specifica le immagini per nome in `Dockerrun.aws.json`. Nota queste convenzioni:

- Le immagini in repository ufficiali su Docker Hub utilizzano un singolo nome (ad esempio `ubuntu` o `mongo`).
- Le immagini in altri repository su Docker Hub vengono qualificate con un nome di organizzazione (ad esempi, `amazon/amazon-ecs-agent`).
- Le immagini in altri repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu` o `account-id.dkr.ecr.us-east-2.amazonaws.com/ubuntu:trusty`).

Per gli ambienti che utilizzano solo la piattaforma Docker, è anche possibile creare una propria immagine durante la creazione dell'ambiente con un Dockerfile. Per informazioni dettagliate, consulta Creazione di immagini personalizzate con un Dockerfile. La piattaforma Docker gestita da ECS non supporta questa funzionalità.

Configurazione degli aggiornamenti gestiti per gli ambienti Docker

Grazie agli aggiornamenti gestiti della piattaforma, puoi configurare il tuo ambiente per eseguire automaticamente l'aggiornamento alla versione più recente in base a una pianificazione.

Per gli ambienti Docker, puoi decidere se eseguire un aggiornamento automatico della piattaforma su più versioni Docker, nel caso in cui la nuova versione della piattaforma includa una nuova versione Docker. Elastic Beanstalk supporta gli aggiornamenti della piattaforma gestita tra versioni Docker durante l'aggiornamento da un ambiente che esegue una versione della piattaforma Docker più recente della 2.9.0. Se una nuova versione della piattaforma include una nuova versione di Docker, Elastic Beanstalk incrementa il numero di versione dell'aggiornamento secondario. Di conseguenza, per consentire l'esecuzione di questa funzionalità su più versioni Docker, abilita gli aggiornamenti gestiti della piattaforma sia per la versione secondaria, sia per le patch. Per impedirla invece, abilita

gli aggiornamenti gestiti della piattaforma per la sola applicazione degli aggiornamenti di versione delle patch.

Ad esempio, il seguente file di configurazione abilita gli aggiornamenti gestiti della piattaforma alle 9:00 UTC di ogni martedì, sia per la versione secondaria sia per le patch; di conseguenza, consente gli aggiornamenti gestiti su più versioni Docker:

Example.ebextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: minor
```

Se in un ambiente è in esecuzione la versione della piattaforma Docker 2.9.0 o precedente, Elastic Beanstalk non esegue mai gli aggiornamenti gestiti se la nuova versione di configurazione della piattaforma include una nuova versione Docker.

Spazio dei nomi di configurazione di Docker

Puoi utilizzare un file di configurazione per impostare le opzioni di configurazione ed eseguire alte attività di configurazione delle istanze durante le distribuzioni. Le opzioni di configurazione possono essere specifiche della piattaforma o essere applicate a tutte le piattaforme del servizio Elastic Beanstalk nel suo complesso. Le opzioni di configurazione sono organizzate in namespace.

> **Note**
>
> Queste informazioni si applicano solo all'ambiente Docker in cui non è in esecuzione Docker Compose. Questa opzione ha un comportamento diverso con gli ambienti Docker che eseguono Docker Compose. Per ulteriori informazioni sui servizi proxy con Docker Compose, consulta Opzioni del container.

Oltre alle opzioni supportate per tutti gli ambienti Elastic Beanstalk, la piattaforma Docker supporta quelle incluse negli spazi dei nomi seguenti:

- `aws:elasticbeanstalk:environment:proxy`: scegliere il server proxy per l'ambiente in uso. Docker supporta l'esecuzione di Nginx oppure nessun server proxy.

Il seguente file di configurazione di esempio consente di configurare un ambiente Docker in modo che non venga eseguito alcun server proxy.

Example.ebextensions/docker-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

Configurazione Docker su AMI Amazon Linux (precedente ad Amazon Linux 2)

Se l'ambiente Docker di Elastic Beanstalk utilizza una versione della piattaforma AMI Amazon Linux (precedente ad Amazon Linux 2), leggi le informazioni aggiuntive presenti in questa sezione.

Utilizzo di un file di autenticazione per un repository privato

Queste informazioni sono rilevanti per l'utente se si utilizzano immagini provenienti da un repository privato. A partire dalla versione Docker 1.7, il comando docker login ha modificato il nome del file di autenticazione e il formato del file. Le versioni della piattaforma Docker su AMI Amazon Linux (precedenti ad Amazon Linux 2) richiedono la versione precedente del file di configurazione del formato `~/.dockercfg`.

Con Docker versione 1.7 e successive, il comando docker login crea il file di autenticazione in `~/.docker/config.json` nel formato seguente.

```
{
    "auths":{
      "server":{
        "auth":"key"
      }
    }
  }
```

Con Docker versione 1.6.2 e versioni precedenti, il comando docker login crea il file di autenticazione in `~/.dockercfg` nel formato seguente.

```
{
    "server" :
    {
```

```
      "auth" : "auth_token",
      "email" : "email"
    }
  }
```

Per convertire un file `config.json`, rimuovi la chiave `auths` esterna, aggiungi una chiave `email` e appiattisci il documento in formato JSON per farlo corrispondere al formato precedente.

Sulle versioni della piattaforma Docker per Amazon Linux 2, Elastic Beanstalk utilizza il nome e il formato del file di autenticazione più recenti. Se si utilizza una versione della piattaforma Docker per Amazon Linux 2, è possibile utilizzare il file di autenticazione creato dal comando docker login senza alcuna conversione.

Configurazione di volumi di storage aggiuntivi

Per migliorare le prestazioni su AMI Amazon Linux, Elastic Beanstalk configura due volumi di storage Amazon EBS per le istanze Amazon EC2 dell'ambiente Docker. Oltre al volume root assegnato per tutti gli ambienti Elastic Beanstalk, un secondo volume da 12 GB denominato `xvdcz` viene assegnato per lo storage delle immagini negli ambienti Docker.

Se hai bisogno di più spazio di archiviazione o i maggior IOPS per le immagini Docker, puoi personalizzare il volume dell'immagine dello storage utilizzando l'opzione di configurazione `BlockDeviceMapping` nel namespace aws:autoscaling:launchconfiguration.

Ad esempio, il seguente file di configurazione aumenta le dimensioni del volume di storage a 100 GB con 500 provisioned IOPS:

Example.ebextensions/blockdevice-xvdcz.config

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:100::io1:500
```

Se utilizzi l'opzione `BlockDeviceMappings` per configurare volumi aggiuntivi per la tua applicazione, devi includere una mappatura per `xvdcz` per essere certo che venga creato. L'esempio seguente consente di configurare due volumi, il volume di storage dell'immagine `xvdcz` con le impostazioni predefinite e ulteriori 24 GB di volume dell'applicazione con nome `sdh`:

Example.ebextensions/blockdevice-sdh.config

```
option_settings:
```

```
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:12:true:gp2,/dev/sdh=:24
```

> **Note**
>
> Quando modifichi le impostazioni in questo spazio dei nomi, Elastic Beanstalk sostituisce tutte le istanze dell'ambiente con le istanze che eseguono la nuova configurazione. Per informazioni dettagliate, consulta Modifiche di configurazione.

Piattaforme legacy

Questo capitolo elenca i contenuti relativi alle precedenti piattaforme Docker che non sono più supportate da. AWS Elastic Beanstalk Gli argomenti qui elencati rimangono in questo documento come riferimento per tutti i clienti che hanno utilizzato queste funzionalità o componenti prima del pensionamento.

Argomenti

- Migrazione a Elastic Beanstalk Docker in esecuzione su Amazon Linux 2 da Docker multi-container in esecuzione su Amazon Linux
- GlassFish Contenitori Docker preconfigurati su Elastic Beanstalk

Migrazione a Elastic Beanstalk Docker in esecuzione su Amazon Linux 2 da Docker multi-container in esecuzione su Amazon Linux

Prima del rilascio del ramo di piattaforma ECS in esecuzione su Amazon Linux 2 (a 64 bit), Elastic Beanstalk ha offerto un percorso di migrazione alternativo ad Amazon Linux 2 per i clienti con ambienti basati sul ramo della piattaforma Docker multi-container in esecuzione su Amazon Linux (a 64 bit). In questo argomento viene descritto il percorso di migrazione e rimane in questo documento come riferimento per tutti i clienti che hanno completato tale percorso di migrazione.

Raccomandiamo ai clienti con ambienti basati sul ramo della piattaforma Docker multi-container in esecuzione su Amazon Linux (a 64 bit)di migrare al ramo della piattaforma ECS in esecuzione su Amazon Linux 2 (a 64 bit). A differenza del percorso di migrazione alternativo, questo approccio continua a utilizzare Amazon ECS per coordinare le distribuzioni di container negli ambienti Docker gestiti da ECS. Questo aspetto consente un approccio più semplice. Non sono richieste modifiche al codice sorgente e lo stesso `Dockerrun.aws.json` v2 è supportato. Per ulteriori informazioni,

consulta Migrazione dell'applicazione Elastic Beanstalk da Docker multi-container gestito da ECS a ECS su Amazon Linux 2023 AL1.

(Legacy) Eseguire la migrazione a Docker in esecuzione sul ramo della piattaforma Amazon Linux 2 da Multi-container Docker in esecuzione su Amazon Linux

Puoi migrare le applicazioni in esecuzione sulla piattaforma Docker multicontainer su AMI Amazon Linux alla piattaforma Docker su Amazon Linux 2. La piattaforma Docker multicontainer su AMI Amazon Linux richiede di specificare immagini dell'applicazione precompilate da eseguire come container. Dopo la migrazione, non si avrà più questa limitazione, perché la piattaforma Docker su Amazon Linux 2 consente a Elastic Beanstalk di creare le immagini del container anche durante la distribuzione. Le applicazioni continueranno a essere eseguite in ambienti Multicontainer con i vantaggi aggiuntivi dello strumento Docker Compose.

Docker Compose è uno strumento per la definizione e l'esecuzione di applicazioni Docker multi-container. Per ulteriori informazioni su Docker Compose e su come installarlo, vedere la sezione Panoramica dei siti Docker Compose e Installazione di Docker Compose.

Il file **`docker-compose.yml`**

Lo strumento Docker Compose utilizza il file `docker-compose.yml` per la configurazione dei servizi applicativi. Questo file sostituisce il file `Dockerrun.aws.json v2` nella directory del progetto dell'applicazione e nel bundle di origine dell'applicazione. Il file `docker-compose.yml` viene creato manualmente e sarà utile fare riferimento al file `Dockerrun.aws.json v2` per la maggior parte dei valori dei parametri.

Di seguito è riportato un esempio di un file `docker-compose.yml` e il file `Dockerrun.aws.json v2` corrispondente per la stessa applicazione. Per ulteriori informazioni sul file `docker-compose.yml`, consulta le informazioni di rifermento sul file Compose. Per ulteriori informazioni sul file `Dockerrun.aws.json v2`, consulta `Dockerrun.aws.json v2`.

`docker-compose.yml`	**`Dockerrun.aws.json v2`**

`version: '2.4'`
`services:`
`  php-app:`
`  image: "php:fpm"`
	`{`
	`  "AWSEBDockerrunVersion": 2,`
	`  "volumes": [`
	`  {`

docker-compose.yml

```
    volumes:
      - "./php-app:/var/www/html:ro
"
      - "${EB_LOG_BASE_DIR}/php-app
:/var/log/sample-app"
    mem_limit: 128m
    environment:
      Container: PHP
  nginx-proxy:
    image: "nginx"
    ports:
      - "80:80"
    volumes:
      - "./php-app:/var/www/html:ro
"
      - "./proxy/conf.d:/etc/nginx/
conf.d:ro"
      - "${EB_LOG_BASE_DIR}/nginx-p
roxy:/var/log/nginx"
    mem_limit: 128m
    links:
      - php-app
```

Dockerrun.aws.json v2

```
      "name": "php-app",
      "host": {
        "sourcePath": "/var/app/
current/php-app"
      }
    },
    {
      "name": "nginx-proxy-conf",
      "host": {
        "sourcePath": "/var/app/
current/proxy/conf.d"
      }
    }
  ],
  "containerDefinitions": [
    {
      "name": "php-app",
      "image": "php:fpm",
      "environment": [
        {
          "name": "Container",
          "value": "PHP"
        }
      ],
      "essential": true,
      "memory": 128,
      "mountPoints": [
        {
          "sourceVolume": "php-app"
,
          "containerPath": "/var/www
/html",
          "readOnly": true
        }
      ]
    },
    {
      "name": "nginx-proxy",
      "image": "nginx",
      "essential": true,
      "memory": 128,
      "portMappings": [
```

docker-compose.yml	Dockerrun.aws.json v2

```
        {
          "hostPort": 80,
          "containerPort": 80
        }
      ],
      "links": [
        "php-app"
      ],
      "mountPoints": [
        {
          "sourceVolume": "php-app"
,
          "containerPath": "/var/www
/html",
          "readOnly": true
        },
        {
          "sourceVolume": "nginx-pr
oxy-conf",
          "containerPath": "/etc/ngi
nx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-lo
gs-nginx-proxy",
          "containerPath": "/var/log
/nginx"
        }
      ]
    }
  ]
}
```

Ulteriori considerazioni sulla migrazione

La piattaforma Docker Amazon Linux 2 e la piattaforma AMI Docker Amazon Linux multicontainer implementano le proprietà dell'ambiente in modo diverso. Queste due piattaforme hanno anche

directory di registro diverse che Elastic Beanstalk crea per ciascuno dei rispettivi container. Dopo la migrazione dalla piattaforma AMI Docker Amazon Linux multicontainer, sarà necessario essere consapevoli di queste diverse implementazioni per il nuovo ambiente della piattaforma Docker Amazon Linux 2.

Area	Piattaforma Docker su Amazon Linux 2 con Docker Compose	Piattaforma Docker multi-container su AMI Amazon Linux
Proprietà dell'ambiente	Affinché i container possano accedere alle proprietà dell'ambiente, è necessario aggiungere un riferimento al file `.env` nel file `docker-compose.yml` . Elastic Beanstalk genera il file `.env`, elencando ciascuna proprietà come variabili di ambiente. Per ulteriori informazioni, consulta Riferimento alle variabili di ambiente nei container.	Elastic Beanstalk può passare direttamente le proprietà di ambiente al container. Il codice in esecuzione nel container può accedere a queste proprietà come variabili di ambiente senza alcuna configurazione aggiuntiva.
Directory di log	Per ogni container, Elastic Beanstalk crea una directory di log chiamata `/var/log/eb-docker/containers/` *`<service name>`* (o `${EB_LOG_BASE_DIR}/<service name>`). Per ulteriori informazioni, consulta Registrazione personalizzata del contenitore Docker con Docker Compose.	Per ogni container, Elastic Beanstalk crea una directory di log chiamata `/var/log/containers/` *`<containername>`* . Per ulteriori informazioni, fai riferimento al campo `mountPoints` in Formato della definizione del container.

Fasi della migrazione

Per eseguire la migrazione alla piattaforma Docker su Amazon Linux 2

1. Creare il file `docker-compose.yml` per l'applicazione, in base al file `Dockerrun.aws.json` v2 esistente. Per ulteriori informazioni, consulta la sezione di cui sopr Il file `docker-compose.yml`.
2. Nella directory principale della cartella del progetto dell'applicazione, sostituire il file `Dockerrun.aws.json` v2 con il file `docker-compose.yml` appena creato.

La struttura della directory dovrebbe essere la seguente.

```
~/myApplication
|-- docker-compose.yml
|-- .ebextensions
|-- php-app
|-- proxy
```

3. Utilizzare il comando eb init per configurare la directory locale per la distribuzione in Elastic Beanstalk.

```
~/myApplication$ eb init -p docker application-name
```

4. Utilizzare il eb create comando per creare un ambiente e distribuire l'immagine Docker.

```
~/myApplication$ eb create environment-name
```

5. Se l'app è un'applicazione Web, dopo l'avvio dell'ambiente, utilizzare il comando eb open per visualizzarla in un browser Web.

```
~/myApplication$ eb open environment-name
```

6. È possibile visualizzare lo stato dell'ambiente appena creato utilizzando il comando eb status.

```
~/myApplication$ eb status environment-name
```

GlassFish Contenitori Docker preconfigurati su Elastic Beanstalk

> **Note**
>
> Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Per ulteriori informazioni sulla migrazione a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Il ramo della GlassFish piattaforma Docker preconfigurato che funziona sull'AMI Amazon Linux (AL1) non è più supportato. Per migrare la tua GlassFish applicazione su una piattaforma Amazon Linux 2023 supportata, distribuisci GlassFish e il codice dell'applicazione su un'immagine Docker Amazon Linux 2023. Per ulteriori informazioni, consulta l'argomento seguente, the section called “Tutorial - GlassFish su Docker: percorso verso Amazon Linux 2023”.

Guida introduttiva ai container Docker preconfigurati su AMI Amazon Linux (precedente ad Amazon Linux 2)

Questa sezione descrive come sviluppare un'applicazione di esempio in locale, quindi distribuire l'applicazione in Elastic Beanstalk con un container Docker preconfigurato.

Configurazione dell'ambiente di sviluppo locale

Per questa procedura dettagliata utilizziamo un'applicazione di esempio. GlassFish

Come configurare l'ambiente

1. Crea una nuova cartella per l'applicazione di esempio.

```
~$ mkdir eb-preconf-example
~$ cd eb-preconf-example
```

2. Scarica il codice dell'applicazione di esempio nella nuova cartella.

```
~$ wget https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-
glassfish-v1.zip
~$ unzip docker-glassfish-v1.zip
~$ rm docker-glassfish-v1.zip
```

Sviluppo e test in locale

Per sviluppare un'applicazione di esempio GlassFish

1. Aggiungi un `Dockerfile` alla cartella principale dell'applicazione. Nel file, specifica l'immagine di base AWS Elastic Beanstalk Docker da utilizzare per eseguire il contenitore Docker preconfigurato locale. Successivamente distribuirai l'applicazione su una versione della piattaforma Docker preconfigurata Elastic GlassFish Beanstalk. Scegli l'immagine di base del Docker che si avvale di questa versione. Per individuare l'immagine Docker corrente della

versione della piattaforma, consulta la sezione Docker preconfigurato nella pagina Piattaforme AWS Elastic Beanstalk supportate nella guida Piattaforme AWS Elastic Beanstalk .

Example b-preconf-example~/E /Dockerfile

```
# For Glassfish 5.0 Java 8
FROM amazon/aws-eb-glassfish:5.0-al-onbuild-2.11.1
```

Per ulteriori informazioni sull'utilizzo di un `Dockerfile`, consulta Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk.

2. Creazione dell'immagine Docker.

```
~/eb-preconf-example$ docker build -t my-app-image .
```

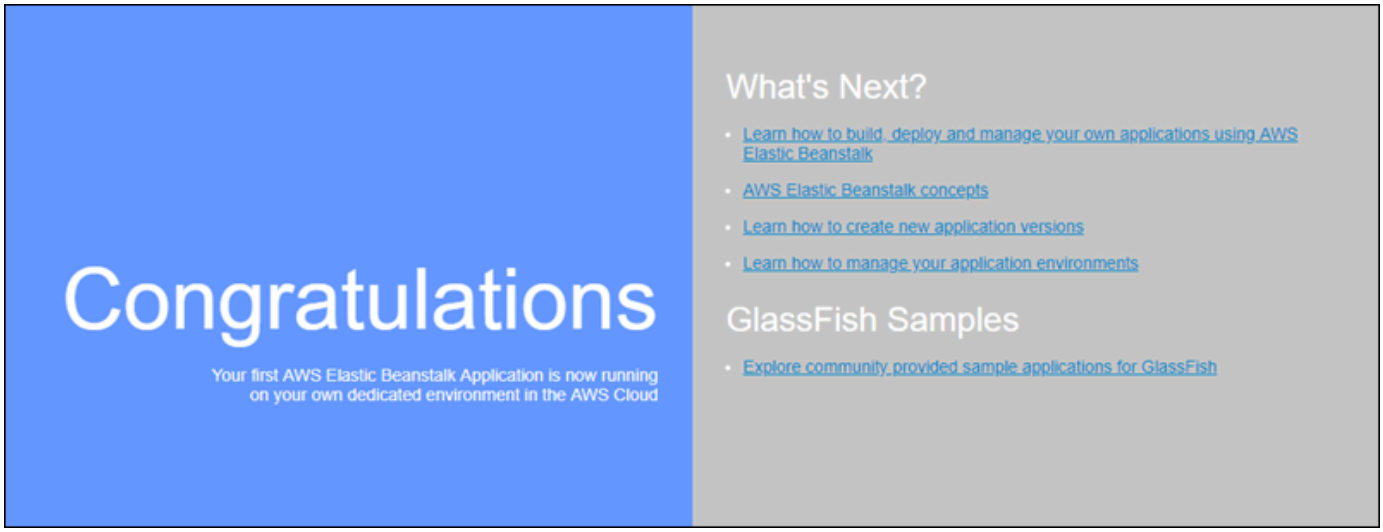
3. Esegui il container Docker dall'immagine.

> **Note**
>
> Devi includere il flag `-p` per mappare la porta 8080 sul container alla porta 3000 del localhost. I container Docker Elastic Beanstalk espongono sempre l'applicazione sulla porta 8080 del container. I flag `-it` eseguono l'immagine come processo interattivo. Il flag `--rm` svuota il file system del container quando il container esce. Opzionalmente puoi includere il flag `-d` per eseguire l'immagine come un daemon.

```
$ docker run -it --rm -p 3000:8080 my-app-image
```

4. Per visualizzare l'applicazione di esempio, digita l'URL seguente nel tuo browser Web.

```
http://localhost:3000
```



Distribuzione in Elastic Beanstalk

Dopo aver testato l'applicazione, puoi distribuirla su Elastic Beanstalk.

Per distribuire l'applicazione su Elastic Beanstalk

1. Nella cartella principale dell'applicazione rinomina `Dockerfile` in `Dockerfile.local`. Questo passaggio è necessario affinché Elastic Beanstalk utilizzi il file `Dockerfile` contenente le istruzioni corrette affinché Elastic Beanstalk crei un'immagine Docker personalizzata su ogni istanza Amazon EC2 nell'ambiente Elastic Beanstalk in uso.

   > **Note**
   >
   > Non è necessario eseguire questa operazione se `Dockerfile` include le istruzioni che modificano l'immagine Docker di base della versione della piattaforma. Non è necessario utilizzare un `Dockerfile` se tutto il `Dockerfile` include solo una riga `FROM` per specificare l'immagine di base da cui creare il container. In tal caso, il `Dockerfile` è ridondante.

2. Crea un bundle di origine dell'applicazione.

```
~/eb-preconf-example$ zip myapp.zip -r *
```

3. Apri la console Elastic Beanstalk con questo link preconfigurato: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? ApplicationName=Tutorials&EnvironmentType=LoadBalanced
4. In Platform (Piattaforma), in Preconfigured – Docker (Preconfigurata - Docker) scegli Glassfish.
5. Per Application code (Codice applicazione), selezionare Upload your code (Carica il tuo codice), quindi Upload (Carica).
6. Scegliere Local file (File locale), selezionare Browse (Sfoglia), quindi per aprir e quindi operare il bundle risorsa dell'applicazione appena creato.
7. Scegli Carica.
8. Selezionare Review and launch (Controlla e avvia).
9. Controlla le impostazioni disponibili, quindi seleziona Create app (Crea applicazione).
10. Quando l'ambiente viene creato, è possibile visualizzare l'applicazione distribuita. Scegli l'ambiente URL, visualizzato nella parte superiore del pannello di controllo della console.

Distribuzione di un' GlassFish applicazione sulla piattaforma Docker: un percorso di migrazione verso Amazon Linux 2023

L'obiettivo di questo tutorial è fornire ai clienti che utilizzano la GlassFish piattaforma Docker preconfigurata (basata sull'AMI Amazon Linux) un percorso di migrazione verso Amazon Linux 2023. Puoi migrare la tua GlassFish applicazione su Amazon Linux 2023 distribuendo GlassFish e il codice dell'applicazione su un'immagine Docker Amazon Linux 2023.

Il tutorial illustra come utilizzare la piattaforma AWS Elastic Beanstalk Docker per distribuire un'applicazione basata sull' GlassFish application server Java EE in un ambiente Elastic Beanstalk.

Dimostriamo due approcci alla creazione di un'immagine Docker:

- Semplice: fornisci il codice sorgente GlassFish dell'applicazione e consenti a Elastic Beanstalk di creare ed eseguire un'immagine Docker come parte del provisioning del tuo ambiente. È facile da configurare, con un costo maggiore del tempo di provisioning delle istanze.
- Advanced (Avanzato): crea un'immagine Docker personalizzata contenente il codice dell'applicazione e le dipendenze, quindi forniscila a Elastic Beanstalk affinché venga usata nell'ambiente. Questo approccio è leggermente più coinvolto e riduce il tempo di provisioning delle istanze nell'ambiente.

Prerequisiti

Questo tutorial presuppone una certa conoscenza delle operazioni di base di Elastic Beanstalk, dell'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) e di Docker. Se non lo hai già fatto, segui le istruzioni in Scopri come iniziare a usare Elastic Beanstalk per avviare il tuo primo ambiente Elastic Beanstalk. Questo tutorial utilizza la CLI EB, ma è anche possibile creare ambienti e caricare applicazioni utilizzando la console Elastic Beanstalk.

Per completare questo tutorial, sono necessari anche i seguenti componenti Docker:

- Un'installazione locale funzionante di Docker. Per ulteriori informazioni, consulta Ottieni Docker sul sito Web della documentazione di Docker.
- Accedi a Docker Hub. Sarà necessario creare un ID Docker per accedere a Docker Hub. Per ulteriori informazioni, consulta Condividere l'applicazione sul sito Web della documentazione di Docker.

Per ulteriori informazioni sulla configurazione di ambienti Docker sulle piattaforme Elastic Beanstalk, consulta Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk in questo stesso capitolo.

Esempio semplice: fornisci il codice dell'applicazione

Questo è un modo semplice per distribuire la tua applicazione. GlassFish Fornisci il codice sorgente dell'applicazione insieme al file `Dockerfile` incluso in questo tutorial. Elastic Beanstalk crea un'immagine Docker che include l'applicazione e lo stack software. GlassFish Elastic Beanstalk esegue quindi l'immagine sulle istanze dell'ambiente.

Un problema con questo approccio è che Elastic Beanstalk crea l'immagine Docker in locale ogni volta che crea un'istanza per l'ambiente. La generazione dell'immagine aumenta il tempo di provisioning dell'istanza. Questo impatto non si limita alla creazione iniziale dell'ambiente, ma si verifica anche durante le operazioni di scalabilità orizzontale.

Per avviare un ambiente con un'applicazione di esempio GlassFish

1. Scaricare l'esempio `docker-glassfish-al2-v1.zip`, quindi espandere il `.zip` file in una directory nell'ambiente di sviluppo.

```
~$ curl https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-
glassfish-al2-v1.zip --output docker-glassfish-al2-v1.zip
```

```
~$ mkdir glassfish-example
~$ cd glassfish-example
~/glassfish-example$ unzip ../docker-glassfish-al2-v1.zip
```

La struttura della directory dovrebbe essere la seguente.

```
~/glassfish-example
|-- Dockerfile
|-- Dockerrun.aws.json
|-- glassfish-start.sh
|-- index.jsp
|-- META-INF
|   |-- LICENSE.txt
|   |-- MANIFEST.MF
|   `-- NOTICE.txt
|-- robots.txt
`-- WEB-INF
    `-- web.xml
```

I seguenti file sono fondamentali per creare ed eseguire un container Docker nel proprio ambiente:

- `Dockerfile`: fornisce istruzioni utilizzate da Docker per creare un'immagine con l'applicazione e le dipendenze richieste.
- `glassfish-start.sh`: uno script di shell eseguito dall'immagine Docker per avviare l'applicazione.
- `Dockerrun.aws.json`— Fornisce una chiave di registrazione, per includere le richieste dei file di registro del server delle GlassFish applicazioni. Se non siete interessati ai GlassFish log, potete omettere questo file.

2. Configura la directory locale per la distribuzione su Elastic Beanstalk.

```
~/glassfish-example$ eb init -p docker glassfish-example
```

3. (Facoltativo) Utilizza il comando eb local run per creare ed eseguire il tuo container in locale.

```
~/glassfish-example$ eb local run --port 8080
```

> **Note**
>
> Per ulteriori informazioni sul comando eb local, consulta the section called “eb local”. Il comando non è supportato da Windows. In alternativa, è possibile creare ed eseguire il container con i comandi docker build e docker run. Per ulteriori informazioni, consulta la documentazione Docker.

4. (Facoltativo) Mentre il tuo container è in esecuzione, utilizza il comando eb local open per visualizzare l'applicazione in un browser Web. In alternativa, apri http://localhost:8080/ in un browser Web.

```
~/glassfish-example$ eb local open
```

5. Utilizza il comando eb create per creare un ambiente e distribuire l'applicazione.

```
~/glassfish-example$ eb create glassfish-example-env
```

6. Dopo l'avvio dell'ambiente, utilizzare il comando eb open per visualizzarlo in un browser Web.

```
~/glassfish-example$ eb open
```

Al termine dell'utilizzo dell'esempio, terminare l'ambiente ed eliminare le risorse correlate.

```
~/glassfish-example$ eb terminate --all
```

Esempio avanzato: fornire un'immagine Docker precompilata

Si tratta di un modo più avanzato per distribuire l'applicazione. GlassFish Basandoti sul primo esempio, crei un'immagine Docker contenente il codice dell'applicazione e lo stack GlassFish software e la invii a Docker Hub. Dopo aver eseguito questo passaggio una tantum, è possibile avviare ambienti Elastic Beanstalk basati sull'immagine personalizzata.

Quando si avvia un ambiente e si fornisce l'immagine Docker, le istanze dell'ambiente scaricano e utilizzano direttamente questa immagine e non è necessario creare un'immagine Docker. Di conseguenza, il tempo di provisioning dell'istanza viene ridotto.

> **Note**
>
> - I seguenti passaggi creano un'immagine Docker disponibile pubblicamente.
> - Dovrai utilizzare i comandi Docker dell'installazione Docker locale, insieme alle credenziali di Docker Hub. Per ulteriori informazioni, vedere la precedente sezione Prerequisiti in questo argomento.

Per avviare un ambiente con un'immagine Docker dell'applicazione predefinita GlassFish

1. Scaricare ed espandere l'esempio `docker-glassfish-al2-v1.zip` come nel precedente esempio semplice. Se hai completato l'esempio, puoi usare la directory che hai già.
2. Crea un'immagine Docker e spingila su Docker Hub. Inserisci il tuo ID Docker per accedere *`docker-id`* a Docker Hub.

```
~/glassfish-example$ docker build -t docker-id/beanstalk-glassfish-example:latest .
~/glassfish-example$ docker push docker-id/beanstalk-glassfish-example:latest
```

> **Note**
>
> Prima di eseguire il push dell'immagine, potrebbe essere necessario eseguire docker login. Se si esegue il comando senza parametri, verrà richiesto di specificare le credenziali di Docker Hub.

3. Creare una directory aggiuntiva.

```
~$ mkdir glassfish-prebuilt
~$ cd glassfish-prebuilt
```

4. Copiare l'esempio seguente in un file denominato `Dockerrun.aws.json`.

   Example~/glassfish-prebuilt/Dockerrun.aws.json

```
{
  "AWSEBDockerrunVersion": "1",
  "Image": {
    "Name": "docker-username/beanstalk-glassfish-example"
  },
```

```
  "Ports": [
    {
      "ContainerPort": 8080,
      "HostPort": 8080
    }
  ],
  "Logging": "/usr/local/glassfish5/glassfish/domains/domain1/logs"
}
```

5. Configura la directory locale per la distribuzione su Elastic Beanstalk.

```
~/glassfish-prebuilt$ eb init -p docker glassfish-prebuilt$
```

6. (Facoltativo) Utilizzare il eb local run comando per eseguire il container in locale.

```
~/glassfish-prebuilt$ eb local run --port 8080
```

7. (Facoltativo) Mentre il tuo container è in esecuzione, utilizza il comando eb local open per visualizzare l'applicazione in un browser Web. In alternativa, apri http://localhost:8080/ in un browser Web.

```
~/glassfish-prebuilt$ eb local open
```

8. Utilizzare il eb create comando per creare un ambiente e distribuire l'immagine Docker.

```
~/glassfish-prebuilt$ eb create glassfish-prebuilt-env
```

9. Dopo l'avvio dell'ambiente, utilizzare il comando eb open per visualizzarlo in un browser Web.

```
~/glassfish-prebuilt$ eb open
```

Al termine dell'utilizzo dell'esempio, terminare l'ambiente ed eliminare le risorse correlate.

```
~/glassfish-prebuilt$ eb terminate --all
```

Ambienti di monitoraggio in Elastic Beanstalk

Con il monitoraggio dello stato di Elastic Beanstalk, puoi verificare la disponibilità delle applicazioni e creare avvisi che si attivano quando le metriche superano le tue soglie. Puoi utilizzare il monitoraggio dello stato di Elastic Beanstalk sia nella console che nella riga di comando per tenere traccia dello stato del tuo ambiente.

Argomenti

- Monitoraggio dello stato dell'ambiente nella console AWS di gestione
- Utilizzo dell'interfaccia a riga di comando EB per monitorare lo stato dell'ambiente
- Report dello stato di base
- Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk
- Gestione degli allarmi
- Visualizzazione della cronologia delle modifiche di un ambiente Elastic Beanstalk
- Visualizzazione del flusso di eventi di un ambiente Elastic Beanstalk
- Creazione di elenchi e connessioni alle istanze dei server
- Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk
- Visualizzazione dei log di distribuzione per un ambiente Elastic Beanstalk

Monitoraggio dello stato dell'ambiente nella console AWS di gestione

Puoi accedere a informazioni operative sull'applicazione dalla console Elastic Beanstalk. La console offre una visione d'insieme dello stato dell'ambiente e dell'applicazione. Nella pagina Environments (Ambienti) della console e nella pagina di ogni applicazione, gli ambienti nell'elenco hanno una codifica a colori che ne indica lo stato.

Per monitorare un ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, scegli Monitoring (Monitoraggio).

La pagina di monitoraggio mostra le statistiche generali sull'ambiente, ad esempio l'utilizzo della CPU e la latenza media. Oltre alle statistiche globali, è possibile visualizzare grafici di monitoraggio che mostrano l'utilizzo delle risorse nel corso del tempo. È possibile fai clic su uno qualsiasi dei grafici per visualizzare informazioni più dettagliate.

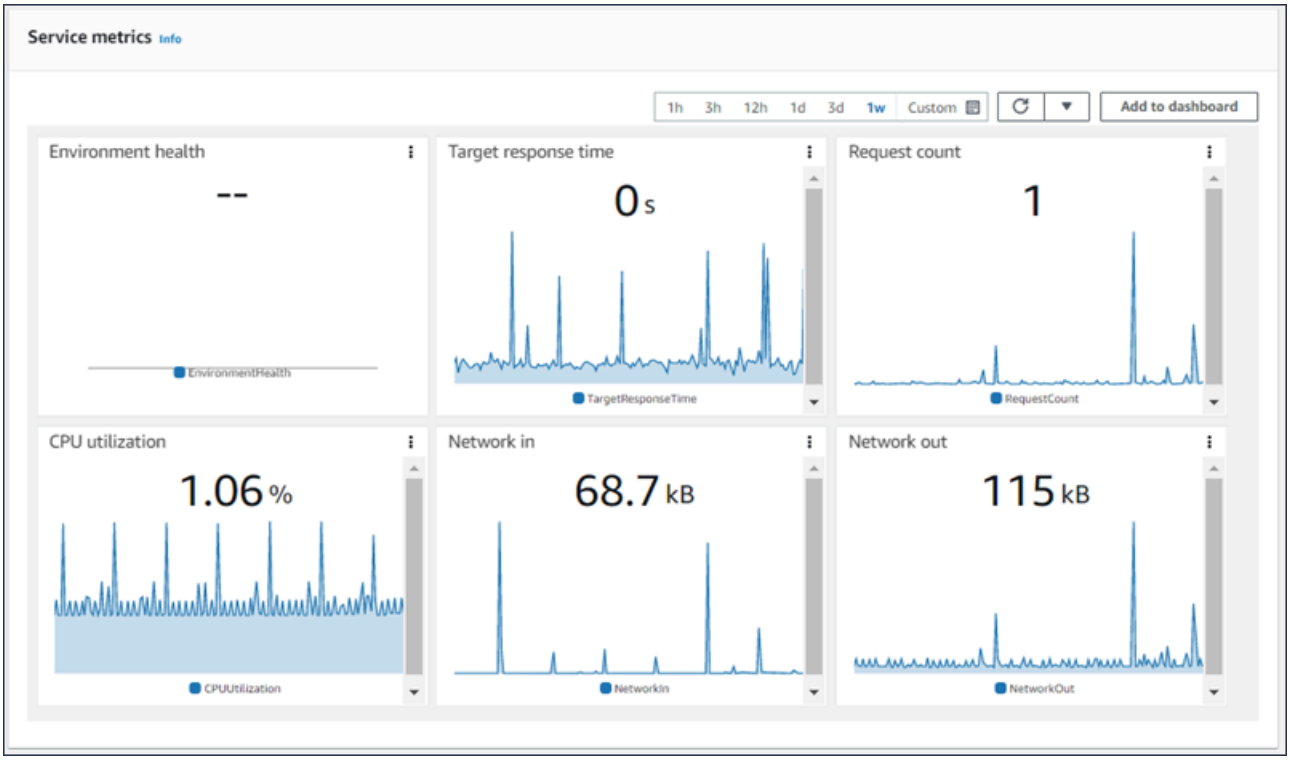
> **Note**
>
> Per impostazione predefinita, sono abilitate solo le CloudWatch metriche di base, che restituiscono i dati in periodi di cinque minuti. Puoi abilitare CloudWatch metriche più granulari di un minuto modificando le impostazioni di configurazione dell'ambiente.

Grafici di monitoraggio

La pagina Monitoraggio mostra una panoramica dei parametri relativi allo stati di integrità dell’ambiente. Ciò include il set di metriche predefinito fornito da Elastic Load Balancing e EC2 Amazon e grafici che mostrano come lo stato dell'ambiente è cambiato nel tempo.

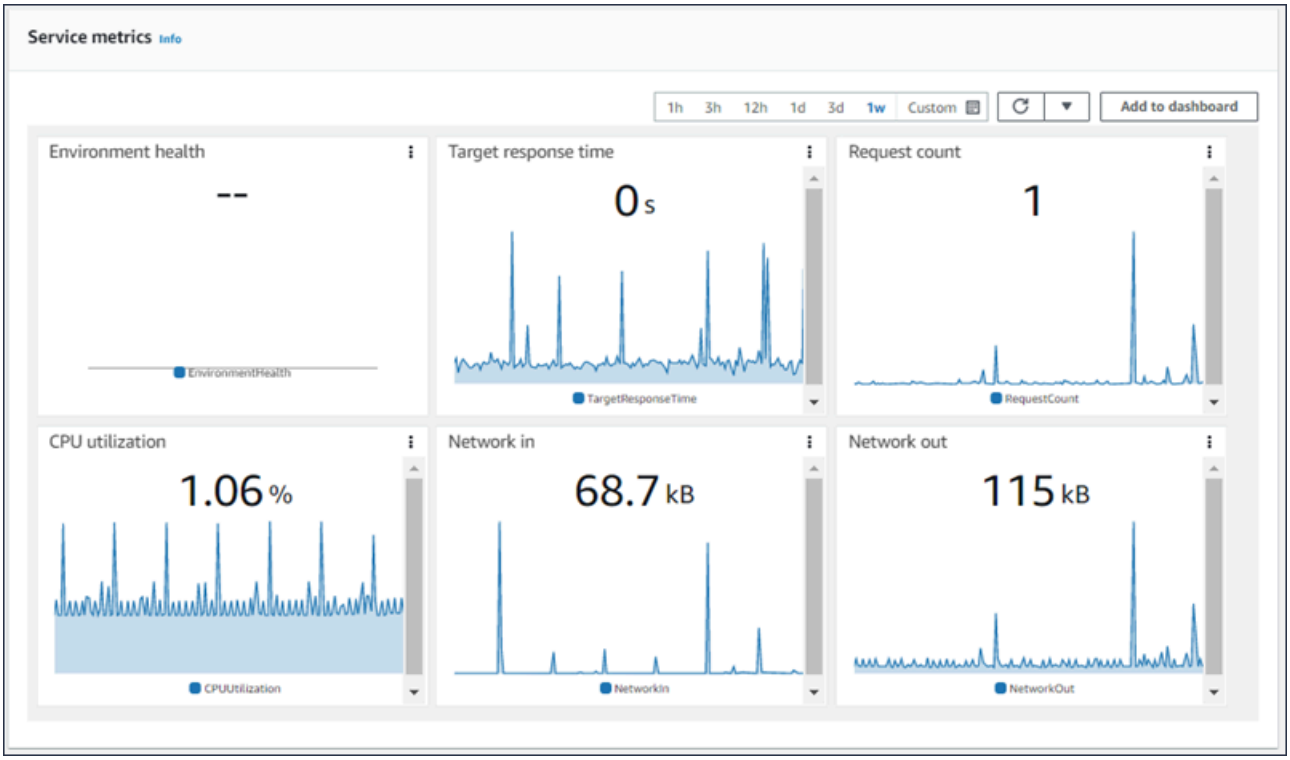
La barra sopra i grafici fornisce una varietà di intervalli di tempo da poter selezionare. Ad esempio, seleziona 1w per visualizzare le informazioni relative all'ultima settimana. Oppure seleziona 3 ore per visualizzare le informazioni relative alle ultime tre ore.

Per una maggiore varietà di selezioni di intervalli di tempo, scegli Personalizzato. Da qui sono disponibili due opzioni di intervallo: Assoluto o Relativo. L'opzione Assoluto consente di specificare un intervallo di date specifico, ad esempio dal 1° gennaio 2023 al 30 giugno 2023. L'opzione Relativo consente di selezionare un numero intero con un'unità di tempo specifica: minuti, ore, giorni, settimane o mesi. Gli esempi includono 10 ore, 10 giorni e 10 mesi.



Personalizzazione della console di monitoraggio

Per creare e visualizzare metriche personalizzate devi usare Amazon CloudWatch. Con CloudWatch puoi creare dashboard personalizzate per monitorare le tue risorse in un'unica visualizzazione. Seleziona Aggiungi alla dashboard per accedere alla CloudWatch console Amazon dalla pagina Monitoraggio. Amazon ti CloudWatch offre la possibilità di creare una nuova dashboard o selezionarne una esistente. Per ulteriori informazioni, consulta Using Amazon CloudWatch dashboard nella Amazon CloudWatch User Guide.



I EC2 parametri Elastic Load Balancing e Amazon sono abilitati per tutti gli ambienti.

Con un livello di integrità migliorato, la EnvironmentHealth metrica viene abilitata e un grafico viene aggiunto automaticamente alla console di monitoraggio. Lo stato migliorato aggiunge inoltre la pagina sullo stato alla console di gestione. Consulta Pubblicazione di parametri Amazon CloudWatch personalizzati per un ambiente per un elenco di parametri di stato migliorato disponibili.

Utilizzo dell'interfaccia a riga di comando EB per monitorare lo stato dell'ambiente

L'Elastic Beanstalk Command Line Interface (EB CLI) è uno strumento a riga di comando per la gestione degli ambienti. AWS Elastic Beanstalk È inoltre possibile utilizzare la CLI EB per monitorare lo stato dell'ambiente in tempo reale e con maggiore granularità rispetto al monitoraggio al momento disponibile nella console Elastic Beanstalk.

Dopo aver installato e configurato l'EB CLI, puoi avviare un nuovo ambiente e distribuirvi il codice con il comando. eb create Se disponi già di un ambiente creato nella console Elastic Beanstalk, puoi collegare la CLI EB eseguendo eb init in una cartella di progetto e seguendo le istruzioni (la cartella del progetto può essere vuota).

> **Important**
>
> Assicurati di utilizzare la versione più recente dell'interfaccia a riga di comando EB eseguendo `pip install` con l'opzione `--upgrade`:
>
> ```
> $ sudo pip install --upgrade awsebcli
> ```
>
> Per istruzioni complete sulla CLI EB, consulta Installa EB CLI con lo script di configurazione (consigliato).

Per utilizzare l'interfaccia a riga di comando EB per monitorare l'integrità dell'ambiente, è necessario configurare una cartella di progetto locale eseguendo eb init e seguendo le istruzioni. Per istruzioni complete, consulta Configurazione dell'interfaccia a riga di comando EB.

Se disponi già di un ambiente in esecuzione in Elastic Beanstalk e desideri utilizzare la CLI EB per monitorare il suo stato, completa questa procedura per collegarla all'ambiente esistente.

Per collegare l'interfaccia a riga di comando EB a un ambiente esistente

1. Apri un terminale a riga di comando e vai alla cartella utente.
2. Crea e apri una nuova cartella per l'ambiente.
3. Esegui il comando eb init, quindi scegli l'applicazione e l'ambiente di cui desideri monitorare lo stato. Se disponi di un solo ambiente che esegue l'applicazione scelta, l'interfaccia a riga di comando EB lo selezionerà automaticamente e non è necessario scegliere l'ambiente, come mostrato nell'esempio seguente.

   ```
   ~/project$ eb init
   Select an application to use
   1) elastic-beanstalk-example
   2) [ Create new Application ]
   (default is 2): 1
   Select the default environment.
   You can change this later by typing "eb use [environment_name]".
   1) elasticBeanstalkEx2-env
   2) elasticBeanstalkExa-env
   (default is 1): 1
   ```

Per monitorare lo stato dell'ambiente utilizzando l'interfaccia a riga di comando EB

1. Apri un terminale a riga di comando e vai alla cartella del progetto.
2. Esegui il comando eb health per visualizzare lo stato di integrità delle istanze nell'ambiente. In questo esempio ci sono cinque istanze in esecuzione in un ambiente Linux.

```
~/project $ eb health
 elasticBeanstalkExa-env                                Ok
 2015-07-08 23:13:20
WebServer
    Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0

  instance-id   status     cause
                                              health
    Overall     Ok
  i-d581497d    Ok
  i-d481497c    Ok
  i-136e00c0    Ok
  i-126e00c1    Ok
  i-8b2cf575    Ok

  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75
 p50     p10                                  requests
    Overall     671.8   100.0    0.0    0.0    0.0    0.003    0.002    0.001
 0.001   0.000
  i-d581497d    143.0    1430      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-d481497c    128.8    1288      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-136e00c0    125.4    1254      0      0      0    0.004    0.002    0.001
 0.001   0.000
  i-126e00c1    133.4    1334      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-8b2cf575    141.2    1412      0      0      0    0.003    0.002    0.001
 0.001   0.000

  instance-id   type       az   running     load 1  load 5      user%  nice%
 system%  idle%   iowait%                              cpu
  i-d581497d    t2.micro   1a   12 mins        0.0    0.04        6.2    0.0
 1.0   92.5       0.1
```

```
  i-d481497c    t2.micro   1a   12 mins       0.01    0.09        5.9    0.0
 1.6   92.4       0.1
  i-136e00c0    t2.micro   1b   12 mins       0.15    0.07        5.5    0.0
 0.9   93.2       0.0
  i-126e00c1    t2.micro   1b   12 mins       0.17    0.14        5.7    0.0
 1.4   92.7       0.1
  i-8b2cf575    t2.micro   1c   1 hour        0.19    0.08        6.5    0.0
 1.2   92.1       0.1

  instance-id   status     id   version              ago
                                       deployments
  i-d581497d    Deployed   1    Sample Application   12 mins
  i-d481497c    Deployed   1    Sample Application   12 mins
  i-136e00c0    Deployed   1    Sample Application   12 mins
  i-126e00c1    Deployed   1    Sample Application   12 mins
  i-8b2cf575    Deployed   1    Sample Application   1 hour
```

In questo esempio c'è una singola istanza in esecuzione in un ambiente Windows.

```
~/project $ eb health
 WindowsSampleApp-env                                  Ok
      2018-05-22 17:33:19
WebServer                                                 IIS 10.0 running on 64bit
 Windows Server 2016/2.2.0
  total      ok    warning  degraded  severe    info   pending  unknown
    1        1        0        0        0        0        0        0

  instance-id           status     cause
                                            health
    Overall             Ok
  i-065716fba0e08a351   Ok

  instance-id           r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90
 p75     p50     p10                        requests
    Overall              13.7   100.0    0.0    0.0    0.0    1.403    0.970
 0.710   0.413   0.079
  i-065716fba0e08a351     2.4   100.0    0.0    0.0    0.0    1.102*   0.865
 0.601   0.413   0.091

  instance-id           type       az   running     % user time    % privileged
 time  % idle time                                 cpu
  i-065716fba0e08a351   t2.large   1b   4 hours             0.2
 0.1         99.7
```

```
  instance-id           status     id   version              ago
                                        deployments
  i-065716fba0e08a351   Deployed   2    Sample Application   4 hours
```

Lettura dell'output

L'output visualizza il nome dell'ambiente, lo stato generale dell'ambiente e la data corrente nella parte superiore della schermata.

```
elasticBeanstalkExa-env                                  Ok
 2015-07-08 23:13:20
```

Le tre righe successive mostrano il tipo di ambiente (» WebServer "in questo caso), la configurazione (Ruby 2.1 con Puma) e un'analisi del numero di istanze presenti in ciascuno dei sette stati.

```
WebServer
 Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0
```

Il resto dell'output è diviso in quattro sezioni. La prima mostra lo stato e la relativa causa per l'ambiente in generale, quindi per ciascuna istanza. L'esempio seguente mostra due istanze nell'ambiente con lo stato `Info` e una causa che indica che è iniziata una distribuzione.

```
  instance-id    status     cause
                                       health
    Overall      Ok
  i-d581497d     Info       Performing application deployment (running for 3 seconds)
  i-d481497c     Info       Performing application deployment (running for 3 seconds)
  i-136e00c0     Ok
  i-126e00c1     Ok
  i-8b2cf575     Ok
```

Per ulteriori informazioni sugli stati di integrità e i relativi colori, consulta Colori e stati indicanti l'integrità.

La sezione requests (richieste) mostra le informazioni dai log del server Web in ciascuna istanza. In questo esempio, ogni istanza sta effettuando le richieste normalmente e non ci sono errori.

```
  instance-id    r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75     p50
    p10                                requests
    Overall      13.7    100.0    0.0    0.0    0.0    1.403    0.970    0.710   0.413
  0.079
  i-d581497d     2.4     100.0    0.0    0.0    0.0    1.102*   0.865    0.601   0.413
  0.091
  i-d481497c     2.7     100.0    0.0    0.0    0.0    0.842*   0.788    0.480   0.305
  0.062
  i-136e00c0     4.1     100.0    0.0    0.0    0.0    1.520*   1.088    0.883   0.524
  0.104
  i-126e00c1     2.2     100.0    0.0    0.0    0.0    1.334*   0.791    0.760   0.344
  0.197
  i-8b2cf575     2.3     100.0    0.0    0.0    0.0    1.162*   0.867    0.698   0.477
  0.076
```

La sezione cpu mostra i parametri del sistema operativo per ogni istanza. L'output cambia in base al sistema operativo. Di seguito è riportato l'output per gli ambienti Linux.

```
  instance-id   type       az   running     load 1  load 5      user%  nice%  system%
 idle%   iowait%                              cpu
  i-d581497d    t2.micro   1a   12 mins        0.0    0.03        0.2    0.0      0.0
 99.7       0.1
  i-d481497c    t2.micro   1a   12 mins        0.0    0.03        0.3    0.0      0.0
 99.7       0.0
  i-136e00c0    t2.micro   1b   12 mins        0.0    0.04        0.1    0.0      0.0
 99.9       0.0
  i-126e00c1    t2.micro   1b   12 mins       0.01    0.04        0.2    0.0      0.0
 99.7       0.1
  i-8b2cf575    t2.micro   1c   1 hour         0.0    0.01        0.2    0.0      0.1
 99.6       0.1
```

Di seguito è riportato l'output per gli ambienti Windows.

```
  instance-id           type       az   running     % user time    % privileged time  %
 idle time
  i-065716fba0e08a351   t2.large   1b   4 hours             0.2                  0.0
      99.8
```

Per ulteriori informazioni sui parametri del server e del sistema operativo, consulta Parametri dell'istanza.

La sezione finale, deployments (distribuzioni), mostra lo stato della distribuzione di ciascuna istanza. Se una distribuzione in sequenza ha esito negativo, è possibile utilizzare l'ID della distribuzione e l'etichetta dello stato e della versione per identificare le istanze dell'ambiente che eseguono la versione errata.

```
  instance-id   status     id   version              ago
                                   deployments
  i-d581497d    Deployed   1    Sample Application   12 mins
  i-d481497c    Deployed   1    Sample Application   12 mins
  i-136e00c0    Deployed   1    Sample Application   12 mins
  i-126e00c1    Deployed   1    Sample Application   12 mins
  i-8b2cf575    Deployed   1    Sample Application   1 hour
```

Visualizzazione dello stato Interattivo

Il comando eb health mostra uno snapshot dello stato dell'ambiente. Per aggiornare le informazioni visualizzate ogni 10 secondi, utilizza l'opzione `--refresh`.

```
$ eb health --refresh
 elasticBeanstalkExa-env                            Ok
 2015-07-09 22:10:04 (1 secs)
WebServer
          Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0

  instance-id   status     cause
                                       health
    Overall     Ok
  i-bb65c145    Ok         Application deployment completed 35 seconds ago and took 26
 seconds
  i-ba65c144    Ok         Application deployment completed 17 seconds ago and took 25
 seconds
  i-f6a2d525    Ok         Application deployment completed 53 seconds ago and took 26
 seconds
  i-e8a2d53b    Ok         Application deployment completed 32 seconds ago and took 31
 seconds
  i-e81cca40    Ok

  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75     p50
   p10                                  requests
```

```
   Overall     671.8   100.0    0.0    0.0    0.0    0.003    0.002    0.001   0.001
0.000
 i-bb65c145    143.0    1430      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-ba65c144    128.8    1288      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-f6a2d525    125.4    1254      0      0      0    0.004    0.002    0.001   0.001
0.000
 i-e8a2d53b    133.4    1334      0      0      0    0.003    0.002    0.001   0.001
0.000
 i-e81cca40    141.2    1412      0      0      0    0.003    0.002    0.001   0.001
0.000

 instance-id   type       az   running     load 1  load 5      user%  nice%  system%
idle%   iowait%                              cpu
 i-bb65c145    t2.micro   1a   12 mins        0.0    0.03        0.2    0.0      0.0
99.7       0.1
 i-ba65c144    t2.micro   1a   12 mins        0.0    0.03        0.3    0.0      0.0
99.7       0.0
 i-f6a2d525    t2.micro   1b   12 mins        0.0    0.04        0.1    0.0      0.0
99.9       0.0
 i-e8a2d53b    t2.micro   1b   12 mins       0.01    0.04        0.2    0.0      0.0
99.7       0.1
 i-e81cca40    t2.micro   1c   1 hour         0.0    0.01        0.2    0.0      0.1
99.6       0.1

 instance-id   status     id   version              ago
                                   deployments
 i-bb65c145    Deployed   1    Sample Application   12 mins
 i-ba65c144    Deployed   1    Sample Application   12 mins
 i-f6a2d525    Deployed   1    Sample Application   12 mins
 i-e8a2d53b    Deployed   1    Sample Application   12 mins
 i-e81cca40    Deployed   1    Sample Application   1 hour

(Commands: Help,Quit, # # # #)
```

Questo esempio illustra un ambiente che ha di recente avuto un ridimensionamento da una a cinque istanze. L'operazione di ridimensionamento è riuscita e tutte le istanze stanno superando i controlli dello stato e sono pronte per accettare le richieste. In modalità interattiva, lo stato si aggiorna ogni 10 secondi. Nell'angolo in alto a destra, un timer segna il tempo fino al prossimo aggiornamento.

Nell'angolo in basso a sinistra, il report visualizza un elenco di opzioni. Per uscire dalla modalità interattiva, premere Q. Per scorrere, premi i tasti freccia. Per consultare un elenco di comandi aggiuntivi, premi H.

Opzioni di visualizzazione dello stato Interattivo

Durante la visualizzazione interattiva dello stato dell'ambiente, è possibile utilizzare la tastiera per regolare la visualizzazione e chiedere a Elastic Beanstalk di sostituire o riavviare le singole istanze. Per visualizzare l'elenco dei comandi disponibili durante la visualizzazione del report di integrità in modalità interattiva, premi H.

```
  up,down,home,end   Scroll vertically
  left,right         Scroll horizontally
  F                  Freeze/unfreeze data
  X                  Replace instance
  B                  Reboot instance
  <,>                Move sort column left/right
  -,+                Sort order descending/ascending
  P                  Save health snapshot data file
  Z                  Toggle color/mono mode
  Q                  Quit this program

  Views
  1                  All tables/split view
  2                  Status Table
  3                  Request Summary Table
  4                  CPU%/Load Table
  H                  This help menu


(press Q or ESC to return)
```

Report dello stato di base

Questo argomento spiega le funzionalità offerte da Elastic Beanstalk Basic Health.

AWS Elastic Beanstalk utilizza informazioni provenienti da più fonti per determinare se l'ambiente è disponibile ed elaborare le richieste da Internet. L'integrità di un ambiente è rappresentata da uno dei quattro colori e viene visualizzata nella pagina di panoramica dell'ambiente della console Elastic Beanstalk. È disponibile anche dall'DescribeEnvironmentsAPI e chiamando eb status con l'EB CLI.

Il sistema di reportistica dello stato di base fornisce informazioni sullo stato delle istanze in un ambiente Elastic Beanstalk basate sui controlli dello stato eseguiti da Elastic Load Balancing per ambienti con carico bilanciato o Amazon Elastic Compute Cloud per ambienti di istanza singola.

Oltre a controllare lo stato delle EC2 istanze, Elastic Beanstalk monitora anche le altre risorse dell'ambiente e segnala le risorse mancanti o configurate in modo errato che possono rendere l'ambiente non disponibile per gli utenti.

Le metriche raccolte dalle risorse del tuo ambiente vengono pubblicate su Amazon CloudWatch a intervalli di cinque minuti. Ciò include le metriche del sistema operativo e le metriche di EC2 richiesta di Elastic Load Balancing. È possibile visualizzare i grafici basati su queste CloudWatch metriche nella pagina Monitoraggio della console di ambiente. Per lo stato di base, questi parametri non sono usati per stabilire lo stato di un ambiente.

Argomenti

- Colori dello stato
- Controlli dello stato di Elastic Load Balancing
- Controlli dello stato per ambienti a singola istanza e con piano lavoratore
- Controlli aggiuntivi
- CloudWatch Metriche Amazon

Colori dello stato

Elastic Beanstalk comunica lo stato di un ambiente di server Web, a seconda del modo in cui l'applicazione in esecuzione risponde al controllo dello stato. Elastic Beanstalk usa uno dei quattro colori per descrivere lo stato, come illustrato nella seguente tabella:

Colore	Description
Grigio	Il tuo ambiente è in fase di aggiornamento.
Verde	Il tuo ambiente ha superato il recente controllo dello stato. Almeno un'istanza nel tuo ambiente è disponibile e riceve le richieste.
Giallo	Il tuo ambiente non ha superato uno o più controlli dello stato. Alcune richieste al tuo ambiente non vanno a buon fine.

Colore	Description
Rosso	Il tuo ambiente ha fallito tre o più controlli dello stato o una risorsa dell'ambiente è diventata non disponibile. Le richieste falliscono costantemente.

Queste descrizioni si applicano solo agli ambienti che utilizzano reportistiche dello stato di base. Consulta Colori e stati indicanti l'integrità per dettagli relativi allo stato migliorato.

Controlli dello stato di Elastic Load Balancing

In un ambiente con carico bilanciato, Elastic Load Balancing invia una richiesta a ogni istanza in un ambiente ogni 10 secondi per confermare che le istanze funzionano correttamente. Per impostazione predefinita, il sistema di bilanciamento del carico è configurato per aprire una connessione TCP sulla porta 80. Se l'istanza riconosce la connessione, si ritiene che stia funzionando correttamente.

È possibile scegliere di ignorare questa impostazione specificando una risorsa esistente nella tua applicazione. Se specifichi un percorso, ad esempio `/health`, l'URL di controllo dello stato è impostato su `HTTP:80/health`. L'URL di controllo dello stato deve essere impostato su un percorso che è sempre servito dall'applicazione. Se è impostato su una pagina statica che viene servita o memorizzata nella cache dal server Web prima dell'applicazione, i controlli dello stato non riveleranno problemi con il server delle applicazioni o il container Web. Per istruzioni su come modificare l'URL del controllo dello stato, vedi Controllo dell'integrità.

Se è stato configurato un URL di controllo dello stato, Elastic Load Balancing prevede una richiesta GET che invia per restituire una risposta di `200 OK`. L'applicazione non supera il controllo dello stato nel caso in cui non risponde entro 5 secondi o se risponde con qualsiasi altro codice di stato HTTP. Dopo 5 errori di controllo dello stato consecutivi, Elastic Load Balancing ritiene che l'istanza sia fuori servizio.

Per ulteriori informazioni sui controlli dello stato di Elastic Load Balancing, consulta Controllo dello stato nella Guida per l'utente di Elastic Load Balancing.

> **Note**
>
> La configurazione di un URL per il controllo dello stato non modifica il comportamento del controllo dello stato di un gruppo Auto Scaling dell'ambiente. Un'istanza non integra viene rimossa dal sistema di bilanciamento del carico, ma non viene sostituita automaticamente da

Amazon Auto EC2 Scaling a meno che non configuri Amazon Auto EC2 Scaling per utilizzare il controllo dello stato di Elastic Load Balancing come base per la sostituzione delle istanze. Per configurare Amazon EC2 Auto Scaling per sostituire le istanze che non superano un controllo dello stato di Elastic Load Balancing, consulta. Impostazione del controllo dello stato di Auto Scaling per l'ambiente Elastic Beanstalk

Controlli dello stato per ambienti a singola istanza e con piano lavoratore

In un ambiente a singola istanza o a livello di lavoro, Elastic Beanstalk determina lo stato dell'istanza monitorandone lo stato dell'istanza Amazon. EC2 Le impostazioni di integrità di Elastic Load Balancing, incluso il controllo dello stato HTTP URLs, non possono essere utilizzate in questi tipi di ambiente.

Per ulteriori informazioni sui controlli dello stato delle EC2 istanze Amazon, consulta Monitoring Instances with Status Checks nella Amazon EC2 User Guide.

Controlli aggiuntivi

Oltre ai controlli dello stato di Elastic Load Balancing, Elastic Beanstalk monitora le risorse nell'ambiente e modifica lo stato di integrità in rosso se non riescono a distribuire, non sono configurate correttamente, oppure non sono disponibili. Questi controlli confermano che:

- Il gruppo Auto Scaling dell'ambiente è disponibile e dispone di almeno una istanza.
- Il gruppo di sicurezza dell'ambiente è disponibile ed è configurato per consentire il traffico in entrata sulla porta 80.
- L'ambiente CNAME esiste ed è rivolto al sistema di bilanciamento del carico di destra.
- In un ambiente worker, sulla coda Amazon Simple Queue Service (Amazon SQS) viene eseguito il polling almeno una volta ogni tre minuti.

CloudWatch Metriche Amazon

Con i report sanitari di base, il servizio Elastic Beanstalk non pubblica alcuna metrica su Amazon. CloudWatch Le CloudWatch metriche utilizzate per produrre grafici nella pagina Monitoraggio della console di ambiente vengono pubblicate dalle risorse presenti nell'ambiente.

Ad esempio, EC2 pubblica le seguenti metriche per le istanze nel gruppo Auto Scaling del tuo ambiente:

`CPUUtilization`

Percentuale di unità di elaborazione attualmente in uso.

`DiskReadBytes`, `DiskReadOps`, `DiskWriteBytes`, `DiskWriteOps`

Numero di byte in lettura e scrittura e numero di operazioni in lettura e scrittura.

`NetworkIn`, `NetworkOut`

Numero di byte inviati e ricevuti.

Elastic Load Balancing pubblica i seguenti parametri per il sistema di bilanciamento del carico dell'ambiente:

`BackendConnectionErrors`

Numero di guasti di connessione tra il sistema di bilanciamento del carico e le istanze dell'ambiente.

`HTTPCode_Backend_2XX`, `HTTPCode_Backend_4XX`

Numero di codici di risposta di successo (2XX) e di errore del client (4XX) generati dalle istanze nel tuo ambiente.

`Latency`

Numero di secondi tra quando il sistema di bilanciamento del carico inoltra una richiesta a un'istanza e quando viene ricevuta la risposta.

`RequestCount`

Numero di richieste completate.

Questi elenchi non sono completi. Per un elenco completo delle metriche che possono essere riportate per queste risorse, consulta i seguenti argomenti nell'Amazon CloudWatch Developer Guide:

Metriche

Namespace	Topic
AWS::ElasticLoadBalancing::LoadBalancer	Parametri e risorse di Elastic Load Balancing

Namespace	Topic
AWS::AutoScaling::AutoScalingGruppo	Parametri e risorse di Amazon Elastic Compute Cloud
AWS::SQS::Queue	Parametri e risorse di Amazon SQS
AWS: :RDS:: DBInstance	Parametri e dimensioni di Amazon RDS

Parametri di stato dell'ambiente lavoratore

Solo per gli ambienti di lavoro, il daemon SQS pubblica una metrica personalizzata per lo stato dell'ambiente in cui il valore 1 è CloudWatch Verde. Puoi esaminare i dati delle metriche CloudWatch sanitarie nel tuo account utilizzando il namespace. `ElasticBeanstalk/SQSD` La dimensione del parametro è `EnvironmentName` e il nome parametro è `Health`. Tutte le istanze pubblicano i parametri sullo stesso spazio dei nomi.

Per abilitare il daemon per la pubblicazione dei parametri, il profilo di istanza dell'ambiente deve avere l'autorizzazione per chiamare `cloudwatch:PutMetricData`. Questa autorizzazione è inclusa nel profilo dell'istanza predefinita. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.

Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk

Questa sezione spiega la funzionalità della funzione Elastic Beanstalk Enhanced Health.

La reportistica avanzata sullo stato di salute è una funzionalità che è possibile abilitare nell'ambiente AWS Elastic Beanstalk per consentire la raccolta di informazioni aggiuntive sulle risorse presenti nell'ambiente. Elastic Beanstalk analizza le informazioni raccolte per offrire un quadro più chiaro dello stato dell'intero ambiente e per facilitare l'identificazione di problemi che possono causare la non disponibilità della tua applicazione.

Oltre alle modifiche sul funzionamento dei colori di stato, il report migliorato aggiunge un descrittore dello stato che offre un'indicazione della gravità dei problemi osservati quando un ambiente è giallo o rosso. Quando sono disponibili ulteriori informazioni sullo stato attuale, puoi scegliere il pulsante Causes (Cause) per visualizzare informazioni dettagliate nella pagina di stato.

Per fornire informazioni dettagliate sullo stato delle istanze Amazon EC2 in esecuzione nel tuo ambiente, Elastic Beanstalk include un agente di stato in Amazon Machine Image (AMI) per ogni versione della piattaforma che supporta lo stato avanzato. L'agente di stato monitora i log del server Web e i parametri di sistema e li trasferisce al servizio Elastic Beanstalk. Elastic Beanstalk analizza questi parametri e i dati di Elastic Load Balancing e Amazon EC2 Auto Scaling per fornire un quadro generale dello stato di un ambiente.

Oltre a raccogliere e a presentare le informazioni relative alle risorse del tuo ambiente, Elastic Beanstalk monitora le risorse nell'ambiente per diverse condizioni di errore e fornisce notifiche per aiutarti a evitare guasti e a risolvere i problemi di configurazione. Tra i fattori che influenzano la stato del tuo ambiente vi sono anche i risultati di ogni richiesta servita dalla tua applicazione, i parametri del sistema operativo delle tue istanze e lo stato delle distribuzioni più recenti.

Puoi visualizzare lo stato in tempo reale utilizzando la pagina Panoramica ambiente nella console Elastic Beanstalk o il comando eb health nell'interfaccia a riga di comando di Elastic Beanstalk (CLI EB). Per registrare e monitorare lo stato dell'ambiente e dell'istanza nel tempo, puoi configurare il tuo ambiente in modo da pubblicare le informazioni raccolte da Elastic Beanstalk per una migliore reportistica sullo stato di salute su Amazon come metriche personalizzate. CloudWatch CloudWatch i costi per le metriche personalizzate si applicano a tutte le metriche diverse da`EnvironmentHealth`, che sono gratuite.

> **Note sulla piattaforma Windows**
>
> Quando si abilita il reporting dello stato avanzato in un ambiente Windows Server, non modificare la configurazione di registrazione IIS. Per il corretto funzionamento del monitoraggio dello stato, la registrazione IIS deve essere configurata con il formato W3C e le destinazioni evento di log ETW event only (Solo evento ETW) o Both log file and ETW event (File di log ed evento ETW).
> Inoltre, non disabilitare né arrestare il servizio Windows agente di stato Elastic Beanstalk su qualsiasi istanza dell'ambiente. Per raccogliere e segnalare informazioni di stato avanzate su un'istanza, questo servizio deve essere abilitato e in esecuzione.

La prima volta che crei un ambiente, Elastic Beanstalk ti chiede di creare i ruoli richiesti e abilita la reportistica sanitaria avanzata per impostazione predefinita. Continua a leggere i dettagli sul funzionamento del reporting dell'integrità avanzato oppure consulta Abilitazione del reporting dello stato avanzato Elastic Beanstalk per iniziare a usarlo.

Argomenti

- Agente di stato Elastic Beanstalk
- Fattori che determinano lo stato delle istanze e dell'ambiente
- Personalizzazione delle regole di controllo dello stato
- Ruoli dello stato migliorato
- Autorizzazione stato avanzata
- Eventi dello stato migliorato
- Comportamento del reporting dello stato migliorato durante aggiornamenti, distribuzioni e dimensionamento
- Abilitazione del reporting dello stato avanzato Elastic Beanstalk
- Monitoraggio dello stato migliorato con la console di gestione dell'ambiente
- Colori e stati indicanti l'integrità
- Parametri dell'istanza
- Configurazione delle regole dello stato migliorato per un ambiente
- Pubblicazione di parametri Amazon CloudWatch personalizzati per un ambiente
- Utilizzo di report dello stato avanzato con l'API Elastic Beanstalk
- Formato dei log dello stato migliorato
- Notifiche e risoluzione dei problemi
- AI-powered analisi dell'ambiente

Agente di stato Elastic Beanstalk

L'agente di stato Elastic Beanstalk è un processo daemon (o servizio, in ambienti Windows) che viene eseguito su ogni istanza Amazon EC2 nell'ambiente per monitorare i parametri di stato a livello di applicazione e di sistema operativo e comunicare eventuali problematiche a Elastic Beanstalk. L'agente di stato è incluso in tutte le versioni della piattaforma a partire dalla versione 2.0 di ciascuna piattaforma.

L'agente sanitario riporta parametri simili a quelli pubblicati CloudWatch da Amazon EC2 Auto Scaling ed Elastic Load Balancing come parte dei report di base sullo stato, tra cui carico della CPU, codici HTTP e latenza. L'agente di stato, tuttavia, comunica direttamente a Elastic Beanstalk, con maggiore granularità e frequenza rispetto al reporting dello stato di base.

Per lo stato di base, questi parametri vengono pubblicati ogni cinque minuti e possono essere monitorati con grafici nella console di gestione dell'ambiente. Con stato avanzato, l'agente di stato Elastic Beanstalk segnala i parametri in Elastic Beanstalk ogni 10 secondi. Elastic Beanstalk utilizza i parametri forniti dall'agente di stato per verificare lo stato di ogni istanza nell'ambiente e, insieme ad altri fattori, per stabilire lo stato di integrità generale dell'ambiente.

Lo stato generale dell'ambiente può essere visualizzato in tempo reale nella pagina di panoramica dell'ambiente della console Elastic Beanstalk e viene pubblicato CloudWatch da Elastic Beanstalk ogni 60 secondi. Puoi visualizzare parametri dettagliati riportati dall'agente di stato in tempo reale con il comando eb health nella interfaccia a riga di comando EB.

A un costo aggiuntivo, puoi scegliere di pubblicare metriche a livello di istanza e ambiente ogni 60 secondi. CloudWatch I parametri pubblicati su CloudWatch possono quindi essere utilizzati per creare grafici di monitoraggio nella console di gestione dell'ambiente.

Il reporting sullo stato migliorato comporta un addebito solo se scegli di pubblicare i parametri del reporting di stato migliorato su CloudWatch. Quando utilizzi lo stato di reporting migliorato,puoi comunque pubblicare i parametri dello stato di base gratuitamente, anche se non scegli di pubblicare i parametri avanzati.

Consulta Parametri dell'istanza per dettagli sui parametri riportati dall'agente dello stato. Per dettagli sulla pubblicazione di metriche sanitarie avanzate su, consulta. CloudWatch Pubblicazione di parametri Amazon CloudWatch personalizzati per un ambiente

Fattori che determinano lo stato delle istanze e dell'ambiente

Oltre ai controlli di sistema del reporting dell'integrità di base, incluso Controlli dello stato di Elastic Load Balancing e il monitoraggio delle risorse, il reporting dell'integrità avanzato di Elastic Beanstalk raccoglie dati aggiuntivi sullo stato delle istanze nel tuo ambiente, inclusi i parametri del sistema operativo, i log del server e lo stato delle operazioni dell'ambiente in corso, come distribuzioni e aggiornamenti. Il servizio di reporting dello stato di Elastic Beanstalk abbina informazioni provenienti da tutte le origini disponibili e le analizza per stabilire lo stato generale dell'ambiente.

Operazioni e comandi

Quando esegui un'operazione nel tuo ambiente, come la distribuzione di una nuova versione di un'applicazione, Elastic Beanstalk esegue diverse modifiche che influenzano lo stato di integrità dell'ambiente.

Ad esempio, quando distribuisci una nuova versione di un'applicazione in un ambiente in cui sono in esecuzione diverse istanze, potresti vedere messaggi simili ai seguenti se monitori lo stato dell'ambiente con EB CLI.

```
  id              status     cause
    Overall       Info       Command is executing on 3 out of 5 instances
  i-bb65c145      Pending    91 % of CPU is in use. 24 % in I/O wait
                             Performing application deployment (running for 31 seconds)
  i-ba65c144      Pending    Performing initialization (running for 12 seconds)
  i-f6a2d525      Ok         Application deployment completed 23 seconds ago and took 26
 seconds
  i-e8a2d53b      Pending    94 % of CPU is in use. 52 % in I/O wait
                             Performing application deployment (running for 33 seconds)
  i-e81cca40      Ok
```

In questo esempio, lo stato generale dell'ambiente è `Ok` e la causa di questo stato è che il comando è in esecuzione su 3 delle 5 istanze. Tre delle istanze nell'ambiente presentano lo stato In attesa, che indica che un'operazione è in corso.

Quando l'operazione viene completata, Elastic Beanstalk comunica ulteriori informazioni sull'operazione. Ad esempio, Elastic Beanstalk visualizza le seguenti informazioni su un'istanza che è già stata aggiornata con la nuova versione dell'applicazione:

```
i-f6a2d525     Ok         Application deployment completed 23 seconds ago and took 26
 seconds
```

Nelle informazioni sullo stato dell'istanza sono inclusi inoltre dettagli sulla distribuzione più recente a ogni istanza nel tuo ambiente. Ogni istanza presenta un ID della distribuzione e lo stato. L'ID della distribuzione è un numero intero che aumenta di uno ogni volta che distribuisci una nuova versione dell'applicazione o modifichi le impostazioni per le opzioni di configurazione dell'istanza, ad esempio le variabili di ambiente. È possibile utilizzare le informazioni di distribuzione per identificare le istanze che eseguono la versione errata dell'applicazione dopo una distribuzione in sequenza non riuscita.

Nella colonna relativa alle cause, Elastic Beanstalk include messaggi informativi sulle operazioni riuscite e su altri stati rilevati in molteplici controlli dello stato, ma che non persistono a tempo indeterminato. Le cause relative a stati di ambienti non integri persistono finché l'ambiente ritorna a uno stato di integrità.

Timeout del comando

Elastic Beanstalk applica un timeout di comando dal momento in cui viene avviata un'operazione per consentire a un'istanza di passare a uno stato di integrità. Questo comando di timeout è impostato nella configurazione di distribuzione e aggiornamento del tuo ambiente (nel namespace aws:elasticbeanstalk:command) e l'impostazione predefinita è su 10 minuti.

Durante gli aggiornamenti in sequenza, Elastic Beanstalk applica un timeout separato per ogni batch nell'operazione. Questo timeout è impostato come parte della configurazione dell'aggiornamento in sequenza dell'ambiente (nello spazio dei nomi aws:autoscaling:updatepolicy:rollingupdate). Se tutte le istanze nel batch sono integre nel periodo di aggiornamento continuo, l'operazione prosegue con il batch successivo. In caso contrario, l'operazione ha esito negativo.

> **Note**
>
> Se l'applicazione non supera i controlli dello stato con uno stato OK, ma è stabile a un livello diverso, puoi impostare l'opzione `HealthCheckSuccessThreshold` nel `aws:elasticbeanstalk:command namespace` per cambiare il livello a cui Elastic Beanstalk considera un'istanza integra.

Affinché un ambiente server Web sia considerato integro, ogni istanza nell'ambiente o batch deve passare 12 controlli dello stato consecutivi nel corso di due minuti. Per un ambiente con piano lavoratore, ogni istanza deve superare 18 controlli dello stato. Prima della scadenza del comando, Elastic Beanstalk non riduce lo stato di integrità di un ambiente quando i controlli dello stato non vanno a buon fine. Se le istanze nell'ambiente diventano integre entro il timeout del comando, l'operazione va a buon fine.

Richieste HTTP

Quando non è in corso un'operazione su un ambiente, l'origine principale di informazioni sull'istanza e sullo stato dell'ambiente è il log del server Web per ogni istanza. Per stabilire lo stato di un'istanza e lo stato generale dell'ambiente, Elastic Beanstalk considera il numero di richieste, il risultato di ogni richiesta e la velocità con cui ogni richiesta è stata risolta.

Sulle piattaforme basate su Linux, Elastic Beanstalk legge e analizza i log del server Web per ottenere informazioni sulle richieste HTTP. Sulla piattaforma Windows Server, Elastic Beanstalk riceve queste informazioni direttamente dal server Web IIS.

L'ambiente potrebbe non disporre di un server Web attivo. Ad esempio, la piattaforma Docker multicontainer non include un server Web. Altre piattaforme includono un server Web e l'applicazione potrebbe disabilitarlo. In questi casi, l'ambiente richiede una configurazione aggiuntiva per fornire all'agente di stato Elastic Beanstalk i log nel formato necessario per inoltrare le informazioni sullo stato al servizio Elastic Beanstalk. Per informazioni dettagliate, consulta Formato dei log dello stato migliorato.

Parametri del sistema operativo

Elastic Beanstalk monitora i parametri del sistema operativo riportati dall'agente di stato per identificare le istanze che sono sempre basse in termini di risorse di sistema.

Consulta Parametri dell'istanza per dettagli sui parametri riportati dall'agente dello stato.

Personalizzazione delle regole di controllo dello stato

Il reporting dello stato avanzato Elastic Beanstalk si basa su un set di regole per determinare lo stato dell'ambiente. Alcune di queste regole potrebbero non essere idonee per la tua applicazione specifica. Un caso comune è un'applicazione che restituisce frequenti errori HTTP 4xx come comportamento predefinito. Utilizzando una delle sue proprie regole predefinite, Elastic Beanstalk conclude che qualcosa non funziona e modifica lo stato dell'ambiente da OK a Warning (Avviso), Degraded (Degradato) o Severe (Grave), a seconda della percentuale di errori. Per gestire questo caso correttamente, Elastic Beanstalk consente di configurare questa regola e di ignorare gli errori 4xx HTTP dell'applicazione. Per informazioni dettagliate, consulta Configurazione delle regole dello stato migliorato per un ambiente.

Ruoli dello stato migliorato

Il reporting dello stato avanzato richiede due ruoli: un ruolo di servizio per Elastic Beanstalk e un profilo dell'istanza per l'ambiente. Il ruolo di servizio consente a Elastic Beanstalk di interagire AWS con altri servizi per conto dell'utente per raccogliere informazioni sulle risorse presenti nell'ambiente. Il profilo dell'istanza consente alle istanze dell'ambiente di scrivere i log su Amazon S3 e di comunicare al servizio Elastic Beanstalk informazioni avanzate sullo stato.

Quando si crea un ambiente Elastic Beanstalk utilizzando la console Elastic Beanstalk o la CLI EB, Elastic Beanstalk crea un ruolo di servizio predefinito e collega le policy gestite richieste a un profilo dell'istanza predefinito per l'ambiente.

Se utilizzi l'API, un SDK o AWS CLI per creare ambienti, devi creare questi ruoli in anticipo e specificarli durante la creazione dell'ambiente per sfruttare l'integrità migliorata. Per le istruzioni su

come creare ruoli appropriati per i tuoi ambienti, consulta Ruoli, profili di istanza e policy utente di Elastic Beanstalk Service.

Si consiglia di utilizzare policy gestite per il profilo dell'istanza e il ruolo di servizio. Le policy gestite sono policy AWS Identity and Access Management (IAM) gestite da Elastic Beanstalk. L'utilizzo di policy gestite garantisce che l'ambiente disponga di tutte le autorizzazioni necessarie per il corretto funzionamento.

Per il profilo dell'istanza, è possibile utilizzare la policy gestita `AWSElasticBeanstalkWebTier` o `AWSElasticBeanstalkWorkerTier`, rispettivamente per un ambiente di livello server Web o di livello worker. Per informazioni dettagliate su queste due policy gestite del profilo dell'istanza, consulta the section called “Profili delle istanze”.

Autorizzazione stato avanzata

Le policy gestite dal profilo dell'istanza Elastic Beanstalk contengono l'autorizzazione per l'operazione `elasticbeanstalk:PutInstanceStatistics`. Questa operazione non fa parte dell'API Elastic Beanstalk. Fa parte di un'API diversa utilizzata internamente dalle istanze di ambiente per comunicare al servizio Elastic Beanstalk informazioni avanzate sull'integrità. Questa API non viene richiamata direttamente.

Quando si crea un nuovo ambiente, l'autorizzazione per l'operazione `elasticbeanstalk:PutInstanceStatistics` è abilitata per impostazione predefinita. Per aumentare la sicurezza dell'ambiente e prevenire lo spoofing dei dati relativi all'integrità per conto dell'utente, è consigliabile tenere abilitata l'autorizzazione per questa operazione. Se utilizzi policy gestite per il profilo dell'istanza, questa caratteristica sarà disponibile per il nuovo ambiente senza ulteriori configurazioni. Tuttavia, se utilizzi un profilo dell'istanza personalizzato anziché una policy gestita, l'ambiente potrebbe mostrare lo stato di integrità No Data (Nessun dato). Ciò si verifica perché le istanze non sono autorizzate per l'operazione che comunica i dati dello stato avanzati al servizio.

Per autorizzare l'operazione, includere la seguente istruzione nel profilo dell'istanza.

```
    {
      "Sid": "ElasticBeanstalkHealthAccess",
      "Action": [
        "elasticbeanstalk:PutInstanceStatistics"
      ],
      "Effect": "Allow",
      "Resource": [
```

```
        "arn:aws:elasticbeanstalk:*:*:application/*",
        "arn:aws:elasticbeanstalk:*:*:environment/*"
      ]
    }
```

Se al momento non si desidera utilizzare l'autorizzazione di integrità avanzata, disabilitarla impostando l'opzione `EnhancedHealthAuthEnabled` nello spazio dei nomi the section called "aws:elasticbeanstalk:healthreporting:system" su `false`. Se questa opzione è disabilitata, le autorizzazioni descritte in precedenza non sono necessarie. È possibile rimuoverle dal profilo dell'istanza per l'accesso con privilegi minimi alle applicazioni e agli ambienti.

> **Note**
>
> In precedenza, l'impostazione predefinita per `EnhancedHealthAuthEnabled` era `false`, che ha avuto come risultato l'autorizzazione per l'operazione `elasticbeanstalk:PutInstanceStatistics`. Questa operazione è disabilitata per impostazione predefinita. Per abilitare questa operazione per un ambiente esistente, imposta l'opzione `EnhancedHealthAuthEnabled` nello spazio dei nomi the section called "aws:elasticbeanstalk:healthreporting:system" su `true`. È possibile configurare questa opzione utilizzando un'impostazione di opzione in un file di configurazione.

Eventi dello stato migliorato

Il sistema dello stato migliorato genera eventi quando un ambiente è in transito tra stati diversi. L'esempio seguente mostra gli eventi generati da un ambiente di transizione tra Info, OK e stati gravi:

Recent events

Show all

1

Time	Type	Details
2020-01-28 16:06:04 UTC-0800	INFO	Environment health has transitioned from Severe to Ok.
2020-01-28 16:05:04 UTC-0800	INFO	Added instance [i-03280193ba1ba4171] to your environment.
2020-01-28 16:05:04 UTC-0800	WARN	Removed instance [i-0a4a27bbbf9994ba5] from your environment due to a EC2 health check failure.
2020-01-28 16:03:04 UTC-0800	WARN	Environment health has transitioned from Ok to Severe. ELB processes are not healthy on all instances. None of the instances are sending data. ELB health is failing or not available for all instances.
2020-01-28 15:19:06 UTC-0800	INFO	Environment health has transitioned from Info to Ok. Application update completed 75 seconds ago and took 22 seconds.

Quando la transizione è verso uno stato peggiore, l'evento dello stato migliorato include un messaggio che indica la causa della transizione.

Non tutte le modifiche di stato a livello di istanza comportano la generazione di un evento da parte di Elastic Beanstalk. Per prevenire falsi allarmi, Elastic Beanstalk genera un evento correlato allo stato solo nel caso in cui un problema persiste in più controlli.

Informazioni in tempo reale sullo stato di integrità a livello di ambiente, tra cui lo stato, il colore e la causa, sono disponibili nella panoramica di controllo dell'ambiente della console Elastic Beanstalk e della CLI EB. Collegando EB CLI al proprio ambiente ed eseguendo il comando eb health, è anche possibile visualizzare gli stati in tempo reale di ciascuna delle istanze nel tuo ambiente.

Comportamento del reporting dello stato migliorato durante aggiornamenti, distribuzioni e dimensionamento

L'abilitazione del reporting dello stato migliorato può influire sul modo in cui l'ambiente si comporta durante gli aggiornamenti di configurazione e le distribuzioni. Elastic Beanstalk non completerà un batch di aggiornamenti finché tutte le istanze non superano i controlli di stato in modo coerente. dato che il reporting dello stato migliorato applica standard più elevati e monitora più fattori, le istanze che superano il controllo dello stato ELB non superano necessariamente i controlli del reporting dello stato migliorato. Vedi gli argomenti in aggiornamenti delle configurazioni in sequenza e in distribuzioni in sequenza per dettagli su come i controlli dello stato influenzano il processo di aggiornamento.

Il reporting dello stato avanzato può anche evidenziare la necessità di impostare un URL per il controllo dello stato adeguato per Elastic Load Balancing. Quando l'ambiente aumenta le risorse per soddisfare la domanda, le nuove istanze inizieranno ad accettare le richieste non appena avranno passato un numero sufficiente di controlli dello stato ELB. Se l'URL del controllo dello stato non è configurato, possono trascorrere minimo 20 secondi prima che una nuova istanza sia in grado di accettare una connessione TCP.

Se la tua applicazione non ha terminato l'avvio quando il sistema di bilanciamento del carico la dichiara abbastanza integra per ricevere il traffico, verrà visualizzata una serie di richieste non riuscite e l'ambiente inizierà a non superare i controlli dello stato. Un URL di controllo dello stato in un percorso servito dalla tua applicazione è in grado di prevenire questo problema; i controlli dello stato ELB non passeranno fino a quando una richiesta GET all'URL del controllo dello stato non restituisce il codice di stato 200.

Abilitazione del reporting dello stato avanzato Elastic Beanstalk

Questo argomento spiega come sono abilitati i report sanitari avanzati. Fornisce procedure per abilitare la funzionalità di integrità avanzata per il tuo ambiente con la console Elastic Beanstalk, l'EB CLI e con una configurazione.ebextensions.

I nuovi ambienti creati con le versioni più recenti della piattaforma includono AWS Elastic Beanstalk Health Agent, che supporta report sanitari avanzati. Se crei il tuo ambiente nella console Elastic Beanstalk o con la CLI EB, lo stato avanzato viene abilitato per impostazione predefinita. Puoi anche impostare le tue preferenze di reporting dello stato nel codice sorgente dell'applicazione, utilizzando i file di configurazione.

Il reporting sullo stato migliorato richiede un profilo dell'istanza e un ruolo del servizio con il set standard di autorizzazioni. Quando crei un ambiente nella console Elastic Beanstalk, Elastic Beanstalk crea automaticamente i ruoli richiesti. Per le istruzioni su come creare il tuo primo ambiente, consulta See Scopri come iniziare a usare Elastic Beanstalk.

Argomenti

- Abilitazione del reporting dello stato avanzato mediante la console Elastic Beanstalk
- Abilitazione del reporting sullo stato migliorato mediante l'interfaccia a riga di comando EB
- Abilitazione del reporting sullo stato migliorato mediante un file di configurazione

Abilitazione del reporting dello stato avanzato mediante la console Elastic Beanstalk

Per abilitare il reporting dello stato avanzato in un ambiente in esecuzione con la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Monitoring (Monitoraggio), scegliere Edit (Modifica).
5. In Health reporting (Reporting sullo stato), per System (Sistema), scegliere Enhanced (Migliorato).

> **Note**
>
> Le opzioni per il reporting dello stato avanzato non vengono visualizzate se utilizzi una piattaforma o una versione non supportata.

6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Per impostazione predefinita, quando si crea un nuovo ambiente con una piattaforma versione 2 (v2), la console Elastic Beanstalk è configurata sul reporting dello stato avanzato. È possibile disabilitare il reporting dello stato avanzato cambiando l'opzione relativa al reporting durante la creazione dell'ambiente.

Per disabilitare il reporting dello stato avanzato durante la creazione di un ambiente utilizzando la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Creare un'applicazione o selezionane una esistente.
3. Crea un ambiente. Nella pagina Create a new environment (Crea nuovo ambiente), prima di selezionare Create environment (Crea ambiente), scegliere Configure more options (Configura più opzioni).
4. Nella categoria di configurazione Monitoring (Monitoraggio), scegliere Edit (Modifica).
5. In Health reporting (Reporting sullo stato), per System (Sistema) scegliere Basic (Base).
6. Scegli Save (Salva).

Abilitazione del reporting sullo stato migliorato mediante l'interfaccia a riga di comando EB

Quando crei un nuovo ambiente con il comando eb create, l'interfaccia a riga di comando EB attiva il reporting sullo stato migliorato per impostazione predefinita e applica il profilo dell'istanza predefinito e il ruolo di servizio.

Puoi specificare un ruolo di servizio diverso per nome mediante l'opzione `--service-role`.

Se disponi di un ambiente in esecuzione con il reporting dello stato di base su una versione della piattaforma v2 e desideri passare al reporting sullo stato migliorato, procedi nel seguente modo.

Per abilitare lo stato migliorato in un ambiente in esecuzione utilizzando la CLI EB

1. Utilizza il comando eb config per aprire il file di configurazione nell'editor di testo predefinito.

```
~/project$ eb config
```

2. Individua lo spazio dei nomi `aws:elasticbeanstalk:environment` nella sezione delle impostazioni. Verifica che il valore di `ServiceRole` non sia nullo e che corrisponda al nome del ruolo di servizio.

```
  aws:elasticbeanstalk:environment:
    EnvironmentType: LoadBalanced
    ServiceRole: aws-elasticbeanstalk-service-role
```

3. Nello spazio dei nomi `aws:elasticbeanstalk:healthreporting:system:`, modifica il valore di `SystemType` in **enhanced**.

```
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
```

4. Salva il file di configurazione e chiudi l'editor di testo.
5. L'interfaccia a riga di comando EB avvia un aggiornamento dell'ambiente per applicare le modifiche di configurazione. Attendi il completamento dell'operazione o premi Ctrl+C per uscire in modo sicuro.

```
~/project$ eb config
Printing Status:
INFO: Environment update is starting.
INFO: Health reporting type changed to ENHANCED.
INFO: Updating environment no-role-test's configuration settings.
```

Abilitazione del reporting sullo stato migliorato mediante un file di configurazione

Puoi abilitare il reporting sullo stato migliorato includendo un file di configurazione nel bundle di origine. L'esempio seguente mostra un file di configurazione che abilita il reporting sullo stato migliorato e assegna il servizio e il profilo dell'istanza predefiniti all'ambiente:

Example.ebextensions/enhanced-health.config

```
option_settings:
```

```
  aws:elasticbeanstalk:healthreporting:system:
    SystemType: enhanced
  aws:autoscaling:launchconfiguration:
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
  aws:elasticbeanstalk:environment:
    ServiceRole: aws-elasticbeanstalk-service-role
```

Se hai creato un profilo dell'istanza o un ruolo del servizio, sostituisci il testo evidenziato con i nomi di questi ruoli.

Monitoraggio dello stato migliorato con la console di gestione dell'ambiente

Quando abiliti la reportistica avanzata sullo stato di salute in AWS Elastic Beanstalk, puoi monitorare lo stato dell'ambiente nella console di gestione dell'ambiente.

Argomenti

- Panoramica dell'ambiente
- Pagina dello stato dell'ambiente
- Pagina di monitoraggio

Panoramica dell'ambiente

Nella panoramica dell'ambiente è visualizzato lo stato di integrità dell'ambiente e sono elencati gli eventi che forniscono informazioni sulle modifiche recenti di questo stato.

Per visualizzare la panoramica dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

Per informazioni dettagliate sullo stato dell'ambiente corrente, aprire la pagina Health (Stato) scegliendo Causes (Cause). In alternativa, nel riquadro di navigazione, selezionare Health (Integrità).

Pagina dello stato dell'ambiente

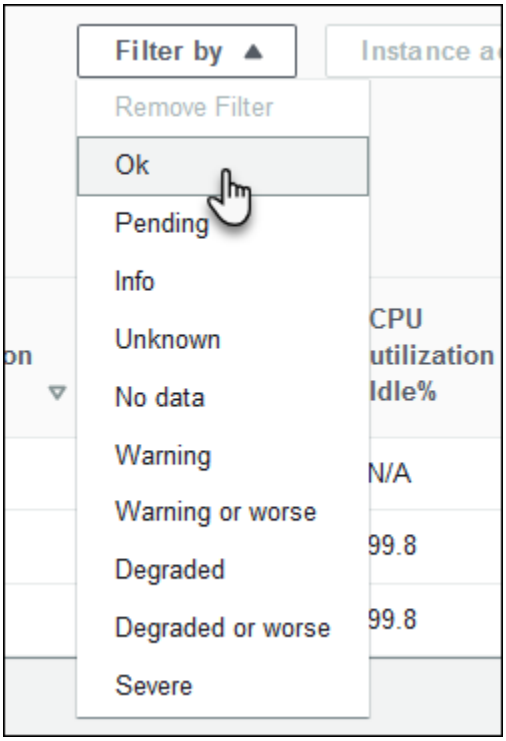
La pagina dello stato mostra lo stato di integrità, i parametri e le cause per l'ambiente e per ogni istanza Amazon EC2 nell'ambiente.

> **Note**
>
> Elastic Beanstalk permette di visualizzare la pagina Health (Stato) solo se hai abilitato il monitoraggio dello stato avanzato per l'ambiente.

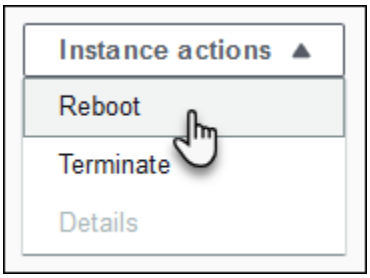
L'immagine seguente mostra la pagina Health (Stato) per un ambiente Linux.

L'immagine seguente mostra la pagina Health (Stato) per un ambiente Windows. Nota che i parametri della CPU sono diversi da quelli in un ambiente Linux.

Nella parte superiore della pagina è possibile visualizzare il numero totale di istanze di ambiente, nonché il numero di istanze per stato. Per visualizzare solo le istanze che hanno uno stato specifico, scegliere Filter By (Filtra per), quindi scegliere uno stato.



Per riavviare o terminare un'istanza danneggiata, seleziona Instance Actions (Operazioni istanza), quindi Reboot (Riavvia) o Terminate (Termina).

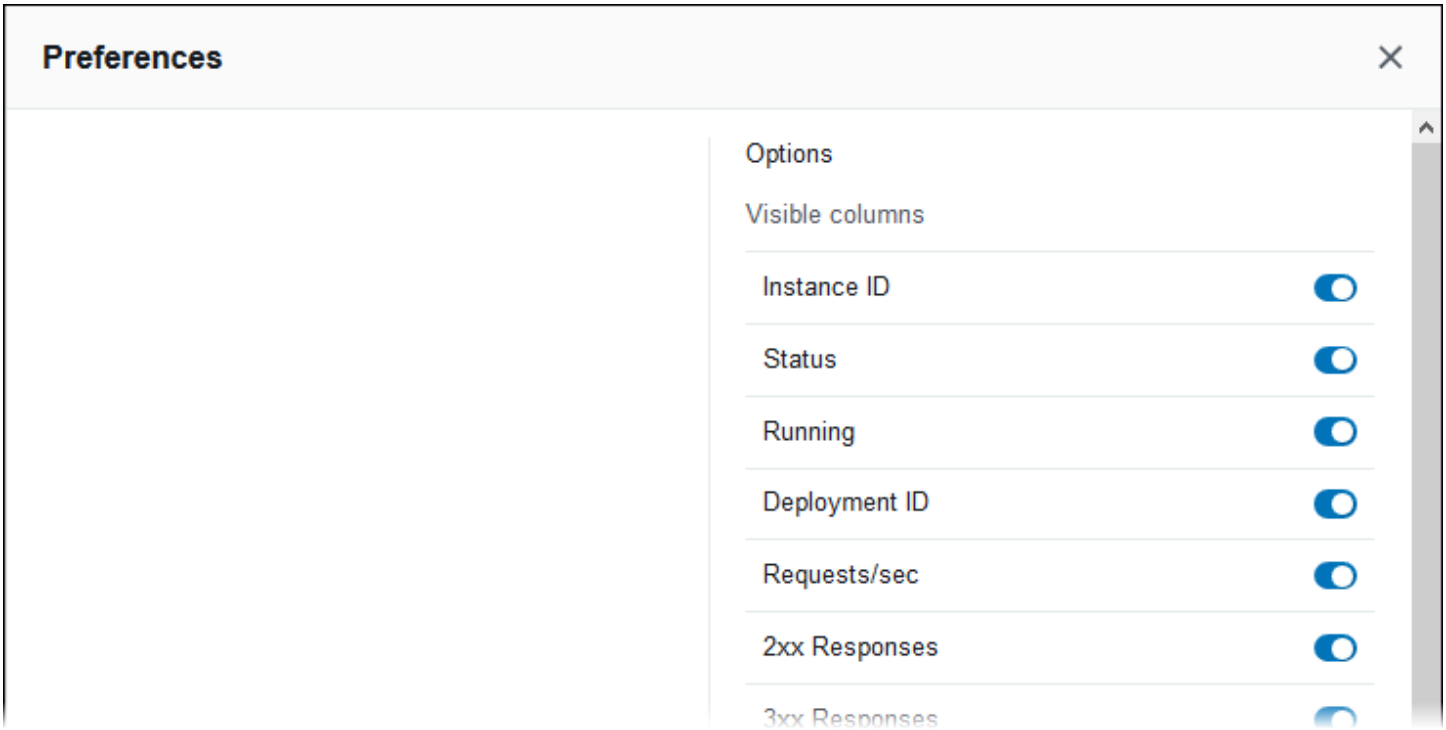


Elastic Beanstalk aggiorna la pagina Health (Stato) ogni 10 secondi. Segnala informazioni sull'ambiente e l'integrità delle istanze.

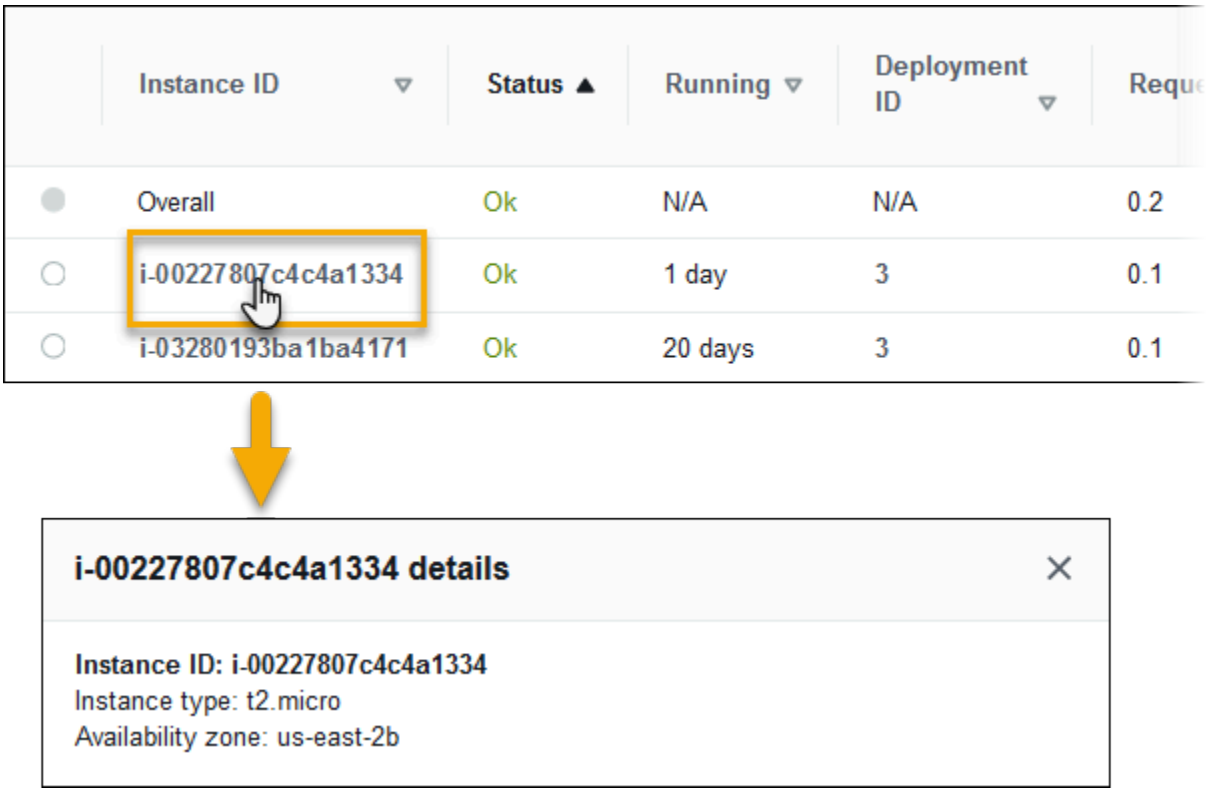
Per ogni istanza Amazon EC2 nell'ambiente, la pagina visualizza l'ID e lo stato dell'istanza, il tempo trascorso dall'avvio dell'istanza, l'ID della distribuzione più recente eseguita sull'istanza, le risposte e la latenza delle richieste fornite dall'istanza e informazioni sul carico e sull'utilizzo della CPU. La riga Overall (Generale) visualizza le informazioni di risposta media e latenza per l'intero ambiente.

La pagina mostra molti dettagli in una tabella molto ampia. Per nascondere alcune colonne, scegliere

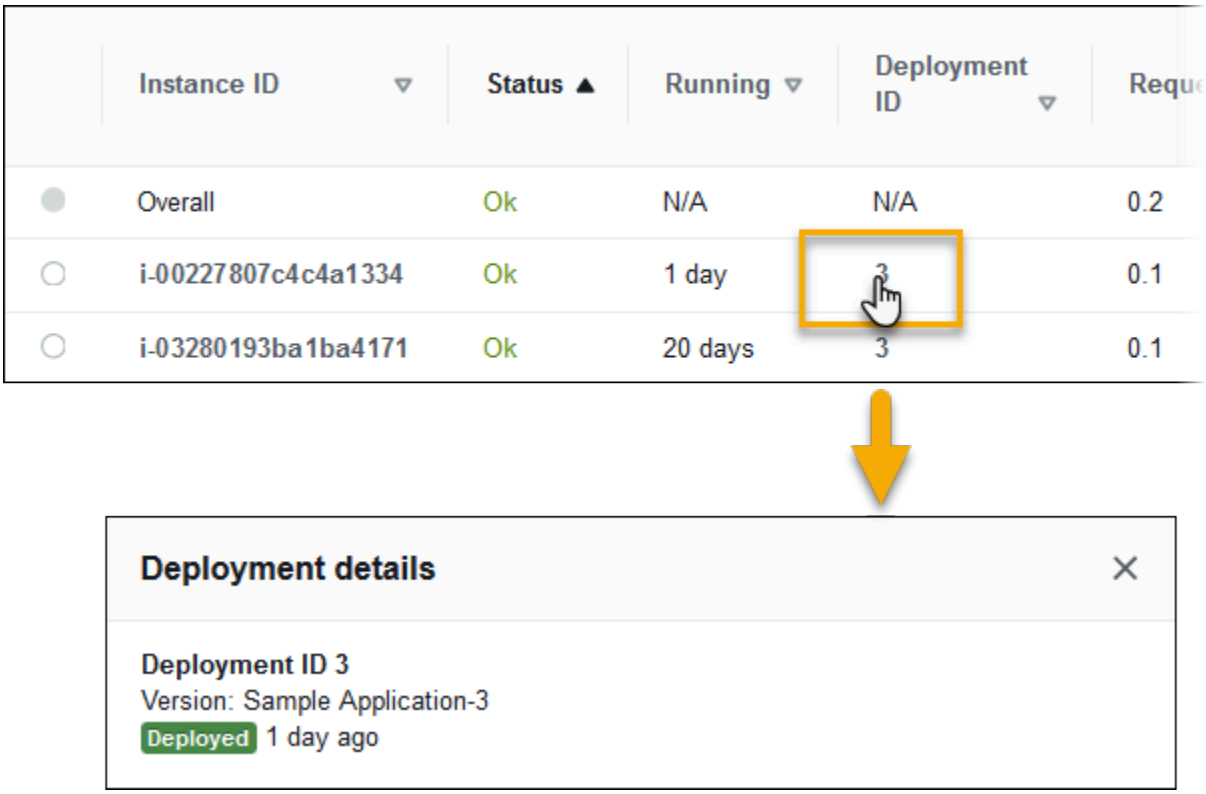
(Preferences (Preferenze)). Selezionare o cancellare i nomi delle colonne, quindi scegliere Confirm (Conferma).



Scegliere l'Instance ID (ID istanza) di qualsiasi istanza per visualizzare ulteriori informazioni sull'istanza, inclusi la zona di disponibilità e il tipo di istanza.



Scegliere Deployment ID (ID distribuzione) di qualsiasi istanza per visualizzare le informazioni sull'ultima distribuzione nell'istanza.



Le informazioni sulla distribuzione includono:

- ID distribuzione: identificatore univoco per la distribuzione. Gli ID di distribuzione iniziano da 1 e aumentano di 1 unità ogni volta che si distribuisce una nuova versione dell'applicazione o si modificano le impostazioni di configurazione che interessano il software o il sistema operativo in esecuzione sulle istanze nel proprio ambiente.
- Versione: etichetta della versione del codice sorgente dell'applicazione utilizzato nella distribuzione.
- Stato: stato della distribuzione, che può essere `In Progress`, `Deployed` o `Failed`.
- Ora: per le distribuzioni in corso, l'ora in cui la distribuzione è iniziata. Per le distribuzioni completate, l'ora in cui la distribuzione è terminata.

Se abiliti X-Ray l'integrazione nel tuo ambiente e strumentalizzi la tua applicazione con l' AWS X-Ray SDK, la pagina Health aggiunge collegamenti alla AWS X-Ray console nella riga panoramica.

sts/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	Loa ave
	100%	0.0%	0.0%	0.0%	0.002	0.002	0.002	0.002	0.001	N/A
	1	0	0	0	0.002	0.002	0.002	0.002	0.002	0.01
	1	0	0	0	0.001	0.001	0.001	0.001	0.001	0.00

Scegli un link per visualizzare le tracce relative alla statistica evidenziata nella AWS X-Ray console.

Pagina di monitoraggio

La pagina Monitoraggio mostra statistiche riassuntive e grafici per le CloudWatch metriche Amazon personalizzate generate dal sistema di reporting sanitario avanzato. Per istruzioni sull'aggiunta di grafici e statistiche a questa pagina, consulta Monitoraggio dello stato dell'ambiente nella console AWS di gestione.

Colori e stati indicanti l'integrità

Il reporting sullo stato migliorato indica l'integrità delle istanze e dell'ambiente complessivo mediante quattro colori, analogamente ai report di stato di base. I report di integrità avanzati forniscono anche sette stati di integrità, che sono descrizioni di una sola parola che forniscono un'indicazione migliore dello stato dell'ambiente.

Stato dell'istanza e stato dell'ambiente

Ogni volta che Elastic Beanstalk esegue un controllo dello stato nell'ambiente, il reporting dello stato avanzato controlla lo stato di ciascuna istanza nell'ambiente analizzando tutti i dati disponibili. Se uno qualsiasi dei controlli di livello inferiore non riesce, Elastic Beanstalk esegue il downgrade dello stato dell'istanza.

Elastic Beanstalk visualizza le informazioni sullo stato per l'intero ambiente (colore, stato e causa) nella console di gestione dell'ambiente. Queste informazioni sono anche disponibili nell'interfaccia a riga di comando di EB. I messaggi su stato e causa dell'integrità per singole istanze vengono aggiornati ogni 10 secondi e sono disponibili tramite la CLI EB quando visualizzi lo stato di integrità con eb health.

Elastic Beanstalk utilizza le modifiche dello stato dell'istanza per valutare lo stato dell'ambiente, ma non modifica immediatamente lo stato di integrità dell'ambiente. Quando i controlli dello stato di un'istanza hanno esito negativo almeno tre volte in un periodo di un minuto, Elastic Beanstalk può

eseguire il downgrade dello stato dell'ambiente. A seconda del numero di istanze nell'ambiente e del problema identificato, un'istanza non integra può far sì che Elastic Beanstalk visualizzi un messaggio informativo o modifichi lo stato di integrità dell'ambiente da verde (OK) a giallo (avviso) o rosso (degradato o grave).

OK – Verde

Questo stato viene visualizzato quando:

- Un'istanza supera i controlli dello stato e l'agente di stato non segnala problemi.
- La maggior parte delle istanze nell'ambiente supera i controlli dello stato e l'agente non segnala problemi gravi.
- Un'istanza supera i controlli dello stato e completa le richieste correttamente.

Esempio: l'ambiente è stato distribuito di recente e accetta richieste normalmente. Il 5% delle richieste restituisce errori serie 400. Distribuzione completata normalmente in ogni istanza.

Messaggio (istanza): indica che la distribuzione dell'applicazione è stata completata 23 secondi fa e ha impiegato 26 secondi.

Warning (Attenzione) – Giallo

Questo stato viene visualizzato quando:

- L'agente di stato segnala un numero moderato di errori di richiesta o altri problemi per un'istanza o un ambiente.
- L'operazione in corso su un'istanza sta impiegando molto tempo.

Esempio: lo stato di un'istanza nell'ambiente è Grave.

Messaggio (ambiente): indica che i servizi interessati riguardano 1 istanza su 5.

Degraded (Degradato) – Rosso

Questo stato viene visualizzati quando l'agente di stato segnala un numero molto elevato di errori di richiesta o altri problemi per un'istanza o un ambiente.

Esempio: l'ambiente è in fase di dimensionamento fino a 5 istanze.

Messaggio (ambiente): indica che 4 istanze attive sono al di sotto della dimensione minima 5 del gruppo Auto Scaling.

Severe (Grave) – Rosso

Questo stato viene visualizzati quando l'agente di stato segnala un numero molto elevato di errori di richiesta o altri problemi per un'istanza o un ambiente.

Esempio: Elastic Beanstalk non è in grado di contattare il sistema di bilanciamento del carico per ottenere lo stato di integrità dell'istanza.

Messaggio (ambiente): indica che l'integrità di ELB ha esito negativo o non è disponibile per tutte le istanze, che nessuna delle istanze invia dati e Impossibile assumere il ruolo «arn:aws:iam: :123456789012:role/». aws-elasticbeanstalk-service-role Verifica che il ruolo esiste e che sia configurato correttamente.

Messaggio (istanza): indica che l'integrità dell'istanza ELB non è stata disponibile per 37 minuti, che non sono presenti dati e che le ultime informazioni sull'integrità sono state restituite 37 minuti fa.

Info – Verde

Questo stato viene visualizzato quando:

- È in corso un'operazione su un'istanza.
- È in corso un'operazione su diverse istanze in un ambiente.

Esempio: una nuova versione dell'applicazione viene distribuita nelle istanze in esecuzione.

Messaggio (ambiente): indica che il comando è in esecuzione su 3 istanze su 5.

Messaggio (istanza): indica che è in corso l'esecuzione della distribuzione dell'applicazione (in esecuzione per 3 secondi).

Pending (In sospeso) – Grigio

Questo stato viene visualizzato quando è in corso un'operazione su un'istanza entro il timeout di un comando.

Esempio: hai creato l'ambiente di recente ed è in corso il bootstrap delle istanze.

Messaggio: indica che è in corso l'esecuzione dell'inizializzazione (in esecuzione per 12 secondi).

Unknown (Sconosciuto) – Grigio

Questo stato viene visualizzato quando Elastic Beanstalk e l'agente di stato segnalano una quantità di dati insufficiente su un'istanza.

Esempio: non viene ricevuto alcun dato.

Sospeso (grigio)

Questo stato viene visualizzata quando Elastic Beanstalk ha interrotto il monitoraggio dello stato dell'ambiente. L'ambiente potrebbe non funzionare correttamente. Alcune gravi condizioni di stato, se si protraggono a lungo, fanno sì che Elastic Beanstalk modifichi lo stato dell'ambiente in Sospeso.

Esempio: Elastic Beanstalk non è in grado di accedere al ruolo di servizio dell'ambiente.

Esempio: il gruppo Auto Scaling creato da Elastic Beanstalk per l'ambiente è stato eliminato.

Messaggio: lo stato dell'ambiente è passato da Ok a Grave. Non sono presenti istanze. La capacità desiderata per il gruppo Auto Scaling è impostata su 1.

Parametri dell'istanza

I parametri delle istanze forniscono informazioni sull'integrità delle istanze in un ambiente. L'agente di stato Elastic Beanstalk viene eseguito in ogni istanza. L'agente raccoglie e inoltra parametri sulle istanze a Elastic Beanstalk, che li analizza per determinare lo stato delle istanze negli ambienti.

L'agente di stato Elastic Beanstalk nell'istanza raccoglie i parametri sulle istanze dai server Web e dal sistema operativo. Per ottenere le informazioni sui server Web su piattaforme basate su Linux, Elastic Beanstalk legge e analizza i log dei server Web. Sulla piattaforma Windows Server, Elastic Beanstalk riceve queste informazioni direttamente dal server Web IIS. I server Web forniscono informazioni sulle richieste HTTP in ingresso: quante richieste sono arrivate, quante hanno restituito errori e il tempo necessario per risolverle. Il sistema operativo fornisce informazioni dello snapshot sullo stato delle risorse delle istanze: il carico della CPU e la distribuzione del tempo speso per ogni tipo di processo.

L'agente di stato raccoglie i parametri del sistema operativo e del server Web e li inoltra a Elastic Beanstalk ogni dieci secondi. Elastic Beanstalk analizza i dati e utilizza i risultati per aggiornare lo stato di ogni istanza e ambiente.

Argomenti

- Parametri del server Web

- Parametri del sistema operativo
- Acquisizione di parametri dei server Web in IIS su Windows Server

Parametri del server Web

Nelle piattaforme basate su Linux l'agente di stato Elastic Beanstalk legge i parametri dei server Web dai log generati dal container o server Web che elabora le richieste in ogni istanza nell'ambiente. Le piattaforme Elastic Beanstalk sono configurate per generare due log: uno in formato leggibile dall'utente e uno in formato leggibile dal computer. L'agente di stato inoltra i log leggibili dal computer a Elastic Beanstalk ogni 10 secondi.

Per ulteriori informazioni sul formato dei log utilizzato da Elastic Beanstalk, consulta Formato dei log dello stato migliorato.

Nella piattaforma Windows Server, Elastic Beanstalk aggiunge un modulo alla pipeline delle richieste del server Web IIS e acquisisce i parametri sugli orari delle richieste HTTP e sui codici di risposta. Il modulo invia questi parametri all'agente di stato nell'istanza usando un canale di comunicazione tra processi ad alte prestazioni. Per informazioni sull'implementazione, consulta Acquisizione di parametri dei server Web in IIS su Windows Server.

Parametri dei server Web segnalati

`RequestCount`

Numero di richieste gestite dal server Web al secondo negli ultimi 10 secondi. Mostrato come valore medio `r/sec` (richieste al secondo) nella CLI EB e in Pagina dello stato dell'ambiente.

`Status2xx`, `Status3xx`, `Status4xx`, `Status5xx`

Numero di richieste che hanno avuto ogni tipo di codice di stato negli ultimi 10 secondi. Ad esempio, le richieste di successo restituiscono 200 OK, i reindirizzamenti 301 e 404 viene restituito se l'URL inserito non corrisponde alle risorse dell'applicazione.

La CLI EB e Pagina dello stato dell'ambiente mostrano questi parametri sotto forma di numeri grezzi delle richieste per le istanze e come percentuale delle richieste generali per gli ambienti.

`p99.9`, `p99`, `p95`, `p90`, `p85`, `p75`, `p50`, `p10`

Latenza media per la percentuale x di richieste più lente negli ultimi 10 secondi, dove x è la differenza tra il numero e 100. Ad esempio, `p99 1.403` indica che l'1% più lento delle richieste negli ultimi 10 secondi aveva una latenza media di 1.403 secondi.

Parametri del sistema operativo

L'agente di stato Elastic Beanstalk comunica i seguenti parametri del sistema operativo. Elastic Beanstalk utilizza questi parametri per identificare le istanze sotto carico elevato. I parametri differiscono in base al sistema operativo.

Parametri del sistema operativo segnalati – Linux

`Running`

La quantità di tempo trascorsa a partire dal momento in cui l'istanza è stata avviata.

`Load 1`, `Load 5`

Media del carico negli ultimi periodi di 1 minuto e 5 minuti. Mostrato come valore decimale che indica il numero medio di processi in esecuzione in quel momento. Se il numero mostrato è superiore al numero di v CPUs (thread) disponibili, il resto è il numero medio di processi in attesa.

Ad esempio, se il tipo di istanza ha quattro v CPUs e il carico è 4,5, durante quel periodo di tempo c'era una media di 0,5 processi in attesa, equivalente a un processo che aspettava il 50 percento del tempo.

`User %`, `Nice %`, `System %`, `Idle %`, `I/O Wait %`

Percentuale di tempo che la CPU ha trascorso in ogni stato negli ultimi 10 secondi.

Parametri del sistema operativo segnalati – Windows

`Running`

La quantità di tempo trascorsa a partire dal momento in cui l'istanza è stata avviata.

`% User Time`, `% Privileged Time`, `% Idle Time`

Percentuale di tempo che la CPU ha trascorso in ogni stato negli ultimi 10 secondi.

Acquisizione di parametri dei server Web in IIS su Windows Server

Nella piattaforma Windows Server, Elastic Beanstalk aggiunge un modulo alla pipeline delle richieste del server Web IIS e acquisisce i parametri sugli orari delle richieste HTTP e sui codici di risposta. Il modulo invia questi parametri all'agente di stato nell'istanza usando un canale di comunicazione

tra processi ad alte prestazioni. L'agente di stato aggrega questi parametri, li combina con quelli del sistema operativo e li invia al servizio Elastic Beanstalk.

Dettagli di implementazione

Per acquisire parametri da IIS, Elastic Beanstalk implementa un `IHttpModule` gestito ed esegue la sottoscrizione agli eventi `BeginRequest` e `EndRequest`. Questo comportamento permette al modulo di segnalare la latenza delle richieste HTTP e i codici di risposta per tutte le richieste Web gestite da IIS. Per aggiungere il modulo alla pipeline delle richieste di IIS, Elastic Beanstalk registra il modulo nella sezione `<modules>` del file di configurazione IIS, `%windir%\System32\inetsrv\config\applicationHost.config`.

Il modulo Elastic Beanstalk in IIS invia i parametri delle richieste Web acquisiti all'agente di stato nell'istanza, che è un servizio Windows denominato `HealthD`. Per inviare questi dati, il modulo usa `NetNamedPipeBinding`, che fornisce un'associazione sicura e affidabile e ottimizzata per la comunicazione nel computer.

Configurazione delle regole dello stato migliorato per un ambiente

AWS Elastic Beanstalk i report avanzati sullo stato di salute si basano su una serie di regole per determinare lo stato dell'ambiente. Alcune di queste regole potrebbero non essere idonee per la tua applicazione specifica. Di seguito sono riportati alcuni esempi comuni:

- Utilizzo di strumenti di test lato client. In questo caso, sono previsti frequenti errori (4xx) del client HTTP.
- Utilizzare AWS WAF insieme all'Application Load Balancer dell'ambiente per bloccare il traffico in entrata indesiderato. In questo caso, Application Load Balancer restituisce un errore HTTP 403 per ogni messaggio in arrivo rifiutato.

Per impostazione predefinita, Elastic Beanstalk include tutti gli errori HTTP 4xx dell'applicazione quando si determina lo stato dell'ambiente. Consente di modificare lo stato di integrità dell'ambiente da OK to Warning (Avviso), Degraded (Degradato) o Severe (Grave), a seconda della percentuale di errori. Per gestire correttamente casi come gli esempi descritti, Elastic Beanstalk consente di configurare alcune regole dello stato migliorato. Puoi ignorare gli errori HTTP 4xx dell'applicazione nelle istanze dell'ambiente oppure quelli restituiti dal sistema di bilanciamento del carico dell'ambiente. In questo argomento viene descritto come apportare queste modifiche alla configurazione.

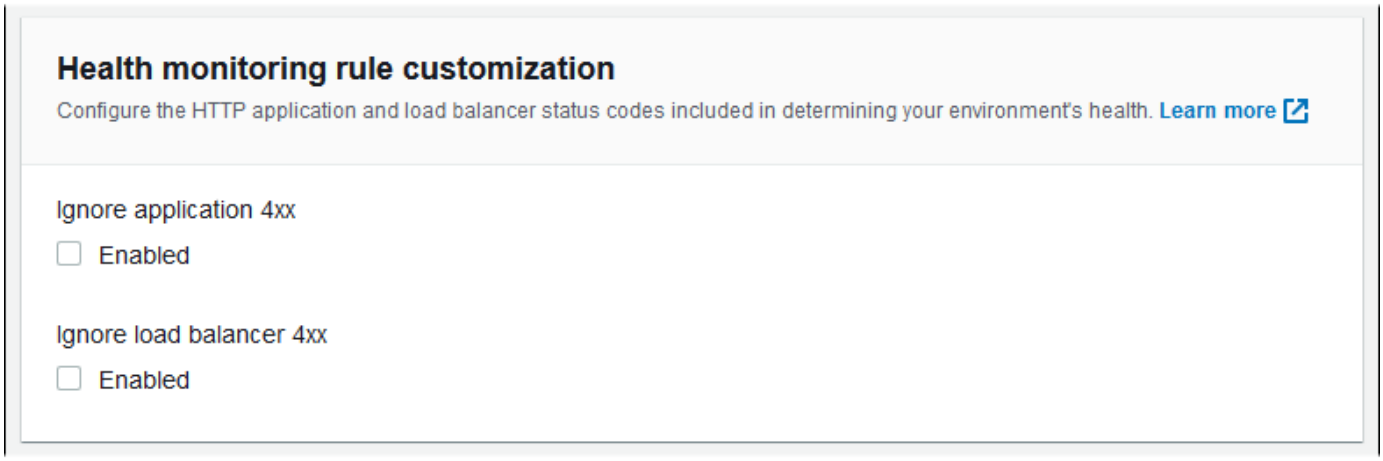
> **Note**
>
> Al momento, sono le uniche personalizzazioni disponibili per le regole dello stato migliorato. Non è possibile configurare lo stato migliorato in modo che ignori errori HTTP diversi da 4xx.

Configurazione delle regole dello stato avanzato mediante la console Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk per configurare le regole dello stato avanzato nell'ambiente.

Per configurare il controllo dei codici di stato HTTP 4xx utilizzando la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Monitoring (Monitoraggio), scegliere Edit (Modifica).
5. In Health monitoring rule customization (Personalizzazione delle regole di monitoraggio dello stato), abilitare o disabilitare le opzioni Ignore (Ignora) desiderate.



6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurazione delle regole dello stato migliorato mediante l'interfaccia a riga di comando EB

Puoi utilizzare l'interfaccia a riga di comando EB per configurare le regole dello stato avanzato salvando la configurazione dell'ambiente in locale, aggiungendo una voce che configuri le regole dello stato migliorato e caricando la configurazione in Elastic Beanstalk. Puoi applicare la configurazione salvata a un ambiente durante o dopo la creazione.

Per configurare il controllo dei codici di stato HTTP 4xx utilizzando l'interfaccia della riga di comando EB e le configurazioni salvate

1. Inizializza la tua cartella di progetto con eb init.
2. Crea un ambiente eseguendo il comando eb create.
3. Salva un modello di configurazione in locale eseguendo il comando eb config save. L'esempio seguente utilizza l'opzione `--cfg` per specificare il nome della configurazione.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-
state.cfg.yml
```

4. Apri il file di configurazione salvato in un editor di testo.
5. In `OptionSettings` > `aws:elasticbeanstalk:healthreporting:system:`, aggiungi una chiave `ConfigDocument` per elencare ciascuna regola dello stato migliorato da configurare. Il seguente `ConfigDocument` disabilita il controllo dei codici di stato HTTP 4xx dell'applicazione, mantenendo abilitato quello del sistema di bilanciamento del carico.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      Rules:
        Environment:
          Application:
            ApplicationRequests4xx:
              Enabled: false
          ELB:
            ELBRequests4xx:
              Enabled: true
      Version: 1
    SystemType: enhanced
```

```
...
```

> **Note**
>
> Puoi combinare `Rules` e `CloudWatchMetrics` nella stessa impostazione dell'opzione `ConfigDocument`. Le `CloudWatchMetrics` sono descritte in Pubblicazione di parametri Amazon CloudWatch personalizzati per un ambiente.
> Se hai già attivato `CloudWatchMetrics`, il file di configurazione che recuperi utilizzando il comando eb config save dispone già di una chiave `ConfigDocument` con una sezione `CloudWatchMetrics`. Non eliminarla, ma aggiungi una sezione `Rules` allo stesso valore dell'opzione `ConfigDocument`.

6. Salva il file di configurazione e chiudi l'editor di testo. In questo esempio, il file di configurazione aggiornato viene salvato con un nome (`02-cloudwatch-enabled.cfg.yml`) diverso rispetto al file di configurazione scaricato. In questo modo, si crea una configurazione salvata distinta quando si carica il file. Puoi utilizzare lo stesso nome del file scaricato per sovrascrivere la configurazione esistente senza crearne una nuova.
7. Usa il comando eb config put per caricare il file di configurazione aggiornato in Elastic Beanstalk.

```
$ eb config put 02-cloudwatch-enabled
```

   Quando utilizzi i comandi eb config `get` e `put` con configurazioni salvate, non includere l'estensione del nome del file.

8. Applica la configurazione salvata all'ambiente in esecuzione.

```
$ eb config --cfg 02-cloudwatch-enabled
```

   L'opzione `--cfg` specifica un file di configurazione con nome che viene applicato all'ambiente. Puoi salvare il file di configurazione in locale o in Elastic Beanstalk. Se un file di configurazione con il nome specificato esiste in entrambi i percorsi, l'interfaccia a riga di comando EB utilizza il file locale.

Configurazione delle regole dello stato migliorato mediante un documento di configurazione

Il documento di configurazione (config) per le regole dello stato migliorato è un documento in formato JSON in cui sono elencate le regole da configurare.

Nel seguente esempio è illustrato un documento di configurazione che disabilita il controllo dei codici di stato HTTP 4xx dell'applicazione e abilita quello del sistema di bilanciamento del carico.

```
{
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    }
  },
  "Version": 1
}
```

Per il AWS CLI, passi il documento come valore per la `Value` chiave in un argomento delle impostazioni delle opzioni, che a sua volta è un oggetto JSON. In questo caso, devi specificare il carattere di escape per le virgolette nel documento incorporato. Il comando seguente verifica se le impostazioni di configurazione sono valide.

```
$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
    {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "ConfigDocument",
        "Value": "{\"Rules\": { \"Environment\": { \"Application\":
 { \"ApplicationRequests4xx\": { \"Enabled\": false } }, \"ELB\": { \"ELBRequests4xx\":
 {\"Enabled\": true } } } }, \"Version\": 1 }"
```

```
    }
]'
```

Per un file di configurazione `.ebextensions` in YAML, puoi fornire il documento in formato JSON così com'è.

```
  option_settings:
    - namespace: aws:elasticbeanstalk:healthreporting:system
      option_name: ConfigDocument
      value: {
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    }
  },
  "Version": 1
}
```

Pubblicazione di parametri Amazon CloudWatch personalizzati per un ambiente

Puoi pubblicare i dati raccolti tramite la reportistica sanitaria AWS Elastic Beanstalk avanzata su Amazon CloudWatch come metriche personalizzate. La pubblicazione di metriche CloudWatch consente di monitorare i cambiamenti nelle prestazioni delle applicazioni nel tempo e identificare potenziali problemi monitorando la scalabilità dell'utilizzo delle risorse e della latenza delle richieste in base al carico.

Pubblicando le metriche su CloudWatch, le rendi disponibili anche per l'uso con grafici e allarmi di monitoraggio. Una metrica gratuita viene abilitata automaticamente quando si utilizzano i report sanitari avanzati. EnvironmentHealth Metriche personalizzate diverse dalle tariffe EnvironmentHealthstandard. CloudWatch

Per pubblicare metriche CloudWatch personalizzate per un ambiente, devi prima abilitare la reportistica avanzata sullo stato dell'ambiente. Per istruzioni, consulta Abilitazione del reporting dello stato avanzato Elastic Beanstalk.

Argomenti

- Parametri del reporting sullo stato migliorato
- Configurazione delle CloudWatch metriche utilizzando la console Elastic Beanstalk
- Configurazione di metriche CloudWatch personalizzate utilizzando l'EB CLI
- Specifica dei documenti di configurazione per parametri personalizzati

Parametri del reporting sullo stato migliorato

Quando abiliti la reportistica sanitaria avanzata nel tuo ambiente, il sistema di reporting sanitario avanzato pubblica automaticamente una metrica CloudWatch personalizzata,. EnvironmentHealth Per pubblicare metriche aggiuntive CloudWatch, configura il tuo ambiente con tali metriche utilizzando la console Elastic Beanstalk, l'EB CLI o .ebextensions.

Puoi pubblicare le seguenti metriche sanitarie avanzate dal tuo ambiente su. CloudWatch

Parametri disponibili – Tutte le piattaforme

`EnvironmentHealth`

Solo ambiente. Questa è l'unica CloudWatch metrica pubblicata dal sistema di reporting sanitario avanzato, a meno che non si configurino metriche aggiuntive. Lo stato dell'ambiente è rappresentato da uno di sette stati. Nella CloudWatch console, questi stati vengono mappati ai seguenti valori:

- 0 – OK
- 1 – Info
- 5 – Unknown (Sconosciuto)
- 10 – No data (Nessun dato)
- 15 – Warning (Avvertenza)
- 20 – Degraded (Degradato)
- 25 – Severe (Grave)

`InstancesSevere`, `InstancesDegraded`, `InstancesWarning`, `InstancesInfo`, `InstancesOk`, `InstancesPending`, `InstancesUnknown`, `InstancesNoData`

Solo ambiente. Questi parametri indicano il numero di istanze nell'ambiente con ciascuno stato di integrità. `InstancesNoData` indica il numero di istanze per cui i dati non vengono ricevuti.

`ApplicationRequestsTotal`, `ApplicationRequests5xx`, `ApplicationRequests4xx`, `ApplicationRequests3xx`, `ApplicationRequests2xx`

Istanza e ambiente. Indica il numero totale di richieste completate dall'istanza o dall'ambiente e il numero di richieste completate con ogni categoria di codice di stato.

`ApplicationLatencyP10`, `ApplicationLatencyP50`, `ApplicationLatencyP75`, `ApplicationLatencyP85`, `ApplicationLatencyP90`, `ApplicationLatencyP95`, `ApplicationLatencyP99`, `ApplicationLatencyP99.9`

Istanza e ambiente. Indica il periodo di tempo medio, in secondi, necessario per completare la percentuale x più veloce di richieste.

`InstanceHealth`

Solo istanza. Indica lo stato corrente dell'istanza. È rappresentato da uno di sette stati. Nella CloudWatch console, questi stati sono mappati ai seguenti valori:

- 0 – OK
- 1 – Info
- 5 – Unknown (Sconosciuto)
- 10 – No data (Nessun dato)
- 15 – Warning (Avvertenza)
- 20 – Degraded (Degradato)
- 25 – Severe (Grave)

Parametri disponibili – Linux

`CPUIrq`, `CPUIdle`, `CPUUser`, `CPUSystem`, `CPUSoftirq`, `CPUIowait`, `CPUNice`

Solo istanza. Indica la percentuale di tempo che la CPU ha passato in ogni stato nell'ultimo minuto.

`LoadAverage1min`

Solo istanza. Il carico medio della CPU dell'istanza nell'ultimo minuto.

`RootFilesystemUtil`

Solo istanza. Indica la percentuale di spazio su disco utilizzato.

Parametri disponibili – Windows

`CPUIdle`, `CPUUser`, `CPUPrivileged`

Solo istanza. Indica la percentuale di tempo che la CPU ha passato in ogni stato nell'ultimo minuto.

Configurazione delle CloudWatch metriche utilizzando la console Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk per configurare il tuo ambiente in cui pubblicare CloudWatch metriche di reporting sanitario avanzate e renderle disponibili per l'uso con grafici e allarmi di monitoraggio.

Per configurare metriche CloudWatch personalizzate nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Monitoring (Monitoraggio), scegliere Edit (Modifica).
5. In Health reporting (Reporting stato), seleziona i parametri di istanza e di ambiente da pubblicare in CloudWatch. Per selezionare più parametri, premere il tasto CTRL durante la selezione.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

L'abilitazione delle metriche CloudWatch personalizzate le aggiunge all'elenco delle metriche disponibili nella pagina Monitoraggio.

Configurazione di metriche CloudWatch personalizzate utilizzando l'EB CLI

Puoi utilizzare l'interfaccia a riga di comando EB per configurare parametri personalizzati salvando la configurazione dell'ambiente in locale, aggiungendo una voce che definisca i parametri da pubblicare e caricando la configurazione in Elastic Beanstalk. Puoi applicare la configurazione salvata a un ambiente durante o dopo la creazione.

Per configurare metriche CloudWatch personalizzate con l'EB CLI e le configurazioni salvate

1. Inizializza la tua cartella di progetto con eb init.
2. Crea un ambiente eseguendo il comando eb create.
3. Salva un modello di configurazione in locale eseguendo il comando eb config save. L'esempio seguente utilizza l'opzione `--cfg` per specificare il nome della configurazione.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-
state.cfg.yml
```

4. Apri il file di configurazione salvato in un editor di testo.
5. In `OptionSettings` `>aws:elasticbeanstalk:healthreporting:system:`, aggiungi una `ConfigDocument` chiave per abilitare ciascuna delle CloudWatch metriche che desideri. Ad esempio, la chiave `ConfigDocument` seguente pubblica i parametri `ApplicationRequests5xx` e `ApplicationRequests4xx` a livello di ambiente e i parametri `ApplicationRequestsTotal` a livello di istanza.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      CloudWatchMetrics:
        Environment:
          ApplicationRequests5xx: 60
          ApplicationRequests4xx: 60
        Instance:
          ApplicationRequestsTotal: 60
      Version: 1
    SystemType: enhanced
...
```

Nell'esempio, 60 indica il numero di secondi tra le misure. Attualmente, questo è il solo valore supportato.

> **Note**
>
> Puoi combinare `CloudWatchMetrics` e `Rules` nella stessa impostazione dell'opzione `ConfigDocument`. Le `Rules` sono descritte in Configurazione delle regole dello stato migliorato per un ambiente.
> Se hai già utilizzato `Rules` per configurare le regole dello stato migliorato, il file di configurazione che recuperi utilizzando il comando eb config save dispone già di una chiave `ConfigDocument` con una sezione `Rules`. Non eliminarla, ma aggiungi una sezione `CloudWatchMetrics` allo stesso valore dell'opzione `ConfigDocument`.

6. Salva il file di configurazione e chiudi l'editor di testo. In questo esempio, il file di configurazione aggiornato viene salvato con un nome (`02-cloudwatch-enabled.cfg.yml`) diverso rispetto al file di configurazione scaricato. In questo modo, si crea una configurazione salvata distinta quando si carica il file. Puoi utilizzare lo stesso nome del file scaricato per sovrascrivere la configurazione esistente senza crearne una nuova.
7. Usa il comando eb config put per caricare il file di configurazione aggiornato in Elastic Beanstalk.

```
$ eb config put 02-cloudwatch-enabled
```

   Quando utilizzi i comandi eb config `get` e `put` con configurazioni salvate, non includere l'estensione di file.

8. Applica la configurazione salvata all'ambiente in esecuzione.

```
$ eb config --cfg 02-cloudwatch-enabled
```

   L'opzione `--cfg` specifica un file di configurazione con nome che viene applicato all'ambiente. Puoi salvare il file di configurazione in locale o in Elastic Beanstalk. Se un file di configurazione con il nome specificato esiste in entrambi i percorsi, l'interfaccia a riga di comando EB utilizza il file locale.

Specifica dei documenti di configurazione per parametri personalizzati

Il documento di configurazione (config) per i parametri CloudWatch personalizzati di Amazon è un documento JSON che elenca i parametri da pubblicare a livello di ambiente e istanza. L'esempio

seguente mostra un documento di configurazione che abilita tutti i parametri personalizzati disponibili su Linux.

```
{
  "CloudWatchMetrics": {
    "Environment": {
      "ApplicationLatencyP99.9": 60,
      "InstancesSevere": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "InstancesUnknown": 60,
      "ApplicationLatencyP85": 60,
      "InstancesInfo": 60,
      "ApplicationRequests2xx": 60,
      "InstancesDegraded": 60,
      "InstancesWarning": 60,
      "ApplicationLatencyP50": 60,
      "ApplicationRequestsTotal": 60,
      "InstancesNoData": 60,
      "InstancesPending": 60,
      "ApplicationLatencyP10": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "InstancesOk": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60
    },
    "Instance": {
      "ApplicationLatencyP99.9": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "ApplicationLatencyP85": 60,
      "CPUUser": 60,
      "ApplicationRequests2xx": 60,
      "CPUIdle": 60,
      "ApplicationLatencyP50": 60,
      "ApplicationRequestsTotal": 60,
      "RootFilesystemUtil": 60,
      "LoadAverage1min": 60,
      "CPUIrq": 60,
```

```
      "CPUNice": 60,
      "CPUIowait": 60,
      "ApplicationLatencyP10": 60,
      "LoadAverage5min": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "CPUSystem": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60,
      "InstanceHealth": 60,
      "CPUSoftirq": 60
    }
  },
  "Version": 1
}
```

Per il AWS CLI, si passa il documento come valore per la `Value` chiave in un argomento delle impostazioni delle opzioni, che a sua volta è un oggetto JSON. In questo caso, devi specificare il carattere di escape per le virgolette nel documento incorporato.

```
$ aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings '[
    {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "ConfigDocument",
        "Value": "{\"CloudWatchMetrics\": {\"Environment\":
 {\"ApplicationLatencyP99.9\": 60,\"InstancesSevere\": 60,\"ApplicationLatencyP90\":
 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\": 60,\"InstancesUnknown
\": 60,\"ApplicationLatencyP85\": 60,\"InstancesInfo\": 60,\"ApplicationRequests2xx
\": 60,\"InstancesDegraded\": 60,\"InstancesWarning\": 60,\"ApplicationLatencyP50\":
 60,\"ApplicationRequestsTotal\": 60,\"InstancesNoData\": 60,\"InstancesPending
\": 60,\"ApplicationLatencyP10\": 60,\"ApplicationRequests5xx\": 60,
\"ApplicationLatencyP75\": 60,\"InstancesOk\": 60,\"ApplicationRequests3xx\": 60,
\"ApplicationRequests4xx\": 60},\"Instance\": {\"ApplicationLatencyP99.9\": 60,
\"ApplicationLatencyP90\": 60,\"ApplicationLatencyP99\": 60,\"ApplicationLatencyP95\":
 60,\"ApplicationLatencyP85\": 60,\"CPUUser\": 60,\"ApplicationRequests2xx\":
 60,\"CPUIdle\": 60,\"ApplicationLatencyP50\": 60,\"ApplicationRequestsTotal\":
 60,\"RootFilesystemUtil\": 60,\"LoadAverage1min\": 60,\"CPUIrq\": 60,\"CPUNice
\": 60,\"CPUIowait\": 60,\"ApplicationLatencyP10\": 60,\"LoadAverage5min\": 60,
\"ApplicationRequests5xx\": 60,\"ApplicationLatencyP75\": 60,\"CPUSystem\": 60,
\"ApplicationRequests3xx\": 60,\"ApplicationRequests4xx\": 60,\"InstanceHealth\": 60,
\"CPUSoftirq\": 60}},\"Version\": 1}"
    }
```

```
]'
```

Per un file di configurazione `.ebextensions` in YAML, puoi fornire il documento in formato JSON così com'è.

```
  option_settings:
    - namespace: aws:elasticbeanstalk:healthreporting:system
      option_name: ConfigDocument
      value: {
  "CloudWatchMetrics": {
    "Environment": {
      "ApplicationLatencyP99.9": 60,
      "InstancesSevere": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "InstancesUnknown": 60,
      "ApplicationLatencyP85": 60,
      "InstancesInfo": 60,
      "ApplicationRequests2xx": 60,
      "InstancesDegraded": 60,
      "InstancesWarning": 60,
      "ApplicationLatencyP50": 60,
      "ApplicationRequestsTotal": 60,
      "InstancesNoData": 60,
      "InstancesPending": 60,
      "ApplicationLatencyP10": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "InstancesOk": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60
    },
    "Instance": {
      "ApplicationLatencyP99.9": 60,
      "ApplicationLatencyP90": 60,
      "ApplicationLatencyP99": 60,
      "ApplicationLatencyP95": 60,
      "ApplicationLatencyP85": 60,
      "CPUUser": 60,
      "ApplicationRequests2xx": 60,
      "CPUIdle": 60,
      "ApplicationLatencyP50": 60,
```

```
      "ApplicationRequestsTotal": 60,
      "RootFilesystemUtil": 60,
      "LoadAverage1min": 60,
      "CPUIrq": 60,
      "CPUNice": 60,
      "CPUIowait": 60,
      "ApplicationLatencyP10": 60,
      "LoadAverage5min": 60,
      "ApplicationRequests5xx": 60,
      "ApplicationLatencyP75": 60,
      "CPUSystem": 60,
      "ApplicationRequests3xx": 60,
      "ApplicationRequests4xx": 60,
      "InstanceHealth": 60,
      "CPUSoftirq": 60
    }
  },
  "Version": 1
}
```

Utilizzo di report dello stato avanzato con l'API Elastic Beanstalk

Poiché la reportistica sanitaria AWS Elastic Beanstalk avanzata richiede ruoli e stack di soluzioni, è necessario aggiornare gli script e il codice utilizzati prima del rilascio della reportistica sanitaria avanzata prima di poterla utilizzare. Per mantenere la compatibilità con le versioni precedenti, il reporting dello stato avanzato non è abilitato per impostazione predefinita quando si crea un ambiente utilizzando l'API Elastic Beanstalk.

Puoi configurare la reportistica sanitaria avanzata impostando il ruolo del servizio, il profilo dell'istanza e le opzioni di CloudWatch configurazione di Amazon per il tuo ambiente. Questa operazione può essere eseguita in due modi: impostando le opzioni di configurazione nella cartella `.ebextensions` con le configurazioni salvate oppure configurandole direttamente nel parametro `create-environment` della chiamata `option-settings`.

Per utilizzare l'API o l'interfaccia a riga di AWS comando (CLI) per creare un ambiente che supporti l'integrità avanzata, devi: SDKs

- Crea un ruolo di servizio e un profilo di istanza con le autorizzazioni opportune
- Crea un nuovo ambiente con una nuova versione della piattaforma
- Imposta il tipo di sistema di stato, il profilo di istanza e le opzioni di configurazione del ruolo del servizio

Utilizza le seguenti opzioni di configurazione negli spazi dei nomi `aws:elasticbeanstalk:healthreporting:system`, `aws:autoscaling:launchconfiguration` e `aws:elasticbeanstalk:environment` per configurare l'ambiente per il reporting dello stato avanzato.

Opzioni di configurazione dello stato migliorato

SystemType

Spazio dei nomi: `aws:elasticbeanstalk:healthreporting:system`

Per abilitare il reporting dello stato avanzato, imposta questo valore su **`enhanced`**.

IamInstanceProfile

Spazio dei nomi: `aws:autoscaling:launchconfiguration`

Imposta questo valore sul nome di un profilo di istanza configurato per l'uso con Elastic Beanstalk.

ServiceRole

Spazio dei nomi: `aws:elasticbeanstalk:environment`

Imposta questo valore sul nome di un ruolo di servizio configurato per l'uso con Elastic Beanstalk.

ConfigDocument (facoltativo)

Spazio dei nomi: `aws:elasticbeanstalk:healthreporting:system`

Un documento JSON che definisce le metriche relative all'istanza e all'ambiente in cui pubblicare. CloudWatch Esempio:

```
{
  "CloudWatchMetrics":
    {
    "Environment":
      {
      "ApplicationLatencyP99.9":60,
      "InstancesSevere":60
      }
    "Instance":
      {
```

```
      "ApplicationLatencyP85":60,
      "CPUUser": 60
      }
    }
  "Version":1
}
```

> **Note**
>
> I documenti config potrebbero richiedere una formattazione speciale, come le virgolette di escape, a seconda di come vengono forniti in Elastic Beanstalk. Per esempi, consulta Specifica dei documenti di configurazione per parametri personalizzati.

Formato dei log dello stato migliorato

AWS Elastic Beanstalk le piattaforme utilizzano un formato di registro del server Web personalizzato per inoltrare in modo efficiente le informazioni sulle richieste HTTP al sistema di segnalazione sanitaria avanzato. che analizza i log, identifica i problemi e imposta lo stato dell'istanza e dell'ambiente di conseguenza. Se disabiliti il server proxy Web nel tuo ambiente e servi le richieste direttamente dal container Web, è comunque possibile utilizzare appieno il reporting dello stato avanzato configurando il server in modo che i log siano pubblicati nel percorso e nel formato impiegati dall'agente di stato Elastic Beanstalk.

> **Note**
>
> Le informazioni contenute in questa pagina si applicano solo alle piattaforme basate su Linux. Nella piattaforma Windows Server, Elastic Beanstalk riceve le informazioni sulle richieste HTTP direttamente dal server Web IIS. Per informazioni dettagliate, vedi Acquisizione di parametri dei server Web in IIS su Windows Server.

Configurazione dei log del server Web

Le piattaforme Elastic Beanstalk sono configurate per produrre due log con informazioni sulle richieste HTTP. Il primo è in formato Verbose e fornisce informazioni dettagliate sulla richiesta, incluse le informazioni sull'utente agente del richiedente e un timestamp leggibile.

/.log var/log/nginx/access

L'esempio seguente è da un proxy nginx in esecuzione in un ambiente con server Web Ruby, ma il formato è simile a quello di Apache.

```
172.31.24.3 - - [23/Jul/2015:00:21:20 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:21 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
172.31.24.3 - - [23/Jul/2015:00:21:22 +0000] "GET / HTTP/1.1" 200 11 "-" "curl/7.22.0
 (x86_64-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23
 librtmp/2.3" "177.72.242.17"
```

Il secondo log è in formato Terse. Include informazioni rilevanti solo per il reporting sullo stato migliorato. Questo log viene pubblicato su una sottocartella denominata `healthd` e ruota su base oraria. I vecchi log vengono eliminati immediatamente dopo la rotazione.

/.log.2015-07-23-00 var/log/nginx/healthd/application

L'esempio seguente mostra un log nel formato leggibile dal computer.

```
1437609879.311"/"200"0.083"0.083"177.72.242.17
1437609879.874"/"200"0.347"0.347"177.72.242.17
1437609880.006"/bad/path"404"0.001"0.001"177.72.242.17
1437609880.058"/"200"0.530"0.530"177.72.242.17
1437609880.928"/bad/path"404"0.001"0.001"177.72.242.17
```

Il formato di log dello stato migliorato include le informazioni riportate di seguito:

- L'ora della richiesta, in tempo Unix
- Il percorso della richiesta
- Il codice di stato HTTP per il risultato
- L'orario della richiesta

- L'orario di upstream
- L'intestazione HTTP `X-Forwarded-For`

Per i proxy nginx, gli orari sono stampati in secondi con virgola mobile, con tre decimali. Per Apache, vengono utilizzati interi microsecondi.

> **Note**
>
> Se viene visualizzato un avviso simile a questo in un file di log, dove `DATE-TIME` indica data e ora, e si sta utilizzando un proxy personalizzato, ad esempio in un ambiente Docker multicontainer, è necessario utilizzare un .ebextension per configurare l'ambiente in modo che `healthd` sia in grado di leggere i file di log:
>
> ```
> W, [DATE-TIME #1922] WARN -- : log file "/var/log/nginx/healthd/
> application.log.DATE-TIME" does not exist
> ```
>
> È possibile iniziare con .ebextension nel Docker multicontainer di esempio.

/etc/nginx/conf.d/webapp_healthd.conf

L'esempio seguente mostra la configurazione di log per nginx con il formato di log `healthd` evidenziato.

```
upstream my_app {
  server unix:///var/run/puma/my_app.sock;
}

log_format healthd '$msec"$uri"'
                '$status"$request_time"$upstream_response_time"'
                '$http_x_forwarded_for';

server {
  listen 80;
  server_name _ localhost; # need to listen to localhost for worker tier

  if ($time_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2})") {
    set $year $1;
    set $month $2;
    set $day $3;
```

```
    set $hour $4;
  }

  access_log  /var/log/nginx/access.log  main;
  access_log /var/log/nginx/healthd/application.log.$year-$month-$day-$hour healthd;

  location / {
    proxy_pass http://my_app; # match the name of upstream directive which is defined
 above
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  }

  location /assets {
    alias /var/app/current/public/assets;
    gzip_static on;
    gzip on;
    expires max;
    add_header Cache-Control public;
  }

  location /public {
    alias /var/app/current/public;
    gzip_static on;
    gzip on;
    expires max;
    add_header Cache-Control public;
  }
}
```

/etc/httpd/conf.d/healthd.conf

L'esempio seguente mostra la configurazione di log per Apache.

```
LogFormat "%{%s}t\"%U\"%s\"%D\"%D\"%{X-Forwarded-For}i" healthd
CustomLog "|/usr/sbin/rotatelogs /var/log/httpd/healthd/application.log.%Y-%m-%d-%H
 3600" healthd
```

Generazione di log per il reporting sullo stato migliorato

Per fornire i log all'agente di stato, è necessario:

- Produrre i log nel formato corretto, come illustrato nella sezione precedente

- Produrre i log su `/var/log/nginx/healthd/`
- Assegnare un nome ai log utilizzando il formato seguente: `application.log.$year-$month-$day-$hour`
- Ruotare i log una volta all'ora
- Non troncare i log

Notifiche e risoluzione dei problemi

> **Prova Amazon Q Developer CLI per la risoluzione dei problemi assistita dall'intelligenza artificiale**
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Troubleshooting Elastic Beanstalk Environments with Amazon Q Developer CLI nei blog. AWS

Questa pagina elenca i messaggi relativi a problemi comuni e i collegamenti a ulteriori informazioni. I messaggi vengono visualizzati nella Pannello di panoramica dell'ambiente console Elastic Beanstalk e vengono registrati negli eventi in cui i problemi di salute persistono in diversi controlli.

Distribuzioni

Elastic Beanstalk monitora l'ambiente per verificare che venga mantenuta la coerenza in seguito alle distribuzioni. Se una distribuzione in sequenza ha esito negativo, la versione dell'applicazione in esecuzione sulle istanze nel proprio ambiente potrebbe variare. Questa situazione può verificarsi se un'implementazione riesce su uno o più batch ma non su tutti.

È stata rilevata una versione dell'applicazione errata in 2 istanze su 5. Versione prevista «v1" (distribuzione 1).

Versione dell'applicazione errata sulle istanze di ambiente. Versione prevista «v1" (implementazione 1).

La versione dell'applicazione prevista non è in esecuzione su alcune o tutte le istanze in un ambiente.

Versione dell'applicazione non corretta «v2" (distribuzione 2). Versione prevista «v1" (distribuzione 1).

L'applicazione distribuita in un'istanza differisce da quella prevista. Se la distribuzione non riesce, la versione prevista viene reimpostata su quella dell'ultima distribuzione riuscita. Nell'esempio precedente la prima distribuzione (versione "v1") è riuscita, ma la seconda (versione "v2") non è riuscita. Le istanze che eseguono "v2" sono considerate non integre.

Per risolvere il problema, avvia un'altra distribuzione. Puoi ridistribuire una versione precedente che sei certo funzioni correttamente o configurare l'ambiente in modo che vengano ignorati i controlli dello stato durante la distribuzione e ridistribuire la nuova versione per imporre il completamento della distribuzione.

Puoi inoltre identificare e terminare le istanze che eseguono la versione sbagliata dell'applicazione. Le istanze terminate verranno sostituite da Elastic Beanstalk che avvierà istanze con la versione corretta. Utilizza il comando dello stato CLI EB per identificare le istanze che eseguono la versione sbagliata dell'applicazione.

Server applicazioni

15% of requests are erroring with HTTP 4xx

20% of the requests to the ELB are erroring with HTTP 4xx.

Un'elevata percentuale di richieste HTTP a un'istanza o un ambiente non riesce con errori 4xx.

Un codice di stato della serie 400 indica che l'utente ha eseguito una richiesta non valida, ad esempio ha richiesto una pagina che non esiste (404 File Not Found) o a cui l'utente non ha accesso (403 Forbidden). Un numero limitato di 404 non è insolito, ma un numero elevato potrebbe indicare che sono presenti link interni o esterni a pagine non disponibili. Questi problemi possono essere risolti correggendo i link interni non validi e aggiungendo reindirizzamenti per quelli esterni.

5% of the requests are failing with HTTP 5xx

3% of the requests to the ELB are failing with HTTP 5xx.

Un'elevata percentuale di richieste HTTP a un'istanza o un ambiente non riesce con codici di stato della serie 500.

Un codice di stato della serie 500 indica che il server di applicazioni ha riscontrato un errore interno. Questi problemi indicano che si è verificato un errore nel codice dell'applicazione che deve essere individuato e risolto in tempi brevi.

95% of CPU is in use

Su un'istanza, l'agente di stato sta segnalando una percentuale di utilizzo della CPU estremamente elevata e imposta lo stato dell'istanza dello stato su Warning (Avviso) o Degraded (Degradata).

Scala l'ambiente per ridurre il carico delle istanze.

Istanza lavoratore

20 messages waiting in the queue (25 seconds ago)

Le richieste vengono aggiunte alla coda dell'ambiente lavoratore più velocemente di quanto possono essere elaborate. Scala l'ambiente per aumentare la capacità.

5 messages in Dead Letter Queue (15 seconds ago)

Le richieste lavoratore falliscono ripetutamente e vengono aggiunte alla the section called “Code di lettere morte”. Controlla le richieste nella coda DLQ per vedere il motivo per cui non sono riuscite.

Altre risorse

4 active instances is below Auto Scaling group minimum size 5

Il numero di istanze in esecuzione nell'ambiente è inferiore al minimo configurato per il gruppo Auto Scaling.

Auto Scaling group (groupname) notifications have been deleted or modified

Le notifiche configurate per il gruppo Auto Scaling sono state modificate al di fuori di Elastic Beanstalk.

AI-powered analisi dell'ambiente

AWS Elastic Beanstalk l' AI-powered analisi identifica le cause profonde e consiglia soluzioni per i problemi di salute ambientale. Quando il tuo ambiente presenta problemi, puoi richiedere un'analisi AI utilizzando le operazioni `RequestEnvironmentInfo` e l'`RetrieveEnvironmentInfo`API con il tipo di `analyze` informazioni per ottenere AI-generated approfondimenti e soluzioni consigliate.

> **Note**
>
> L'analisi AI è disponibile nelle versioni supportate della piattaforma Amazon Linux 2 e AL2023 rilasciate a partire dal 26 febbraio 2026. Per le piattaforme Windows Server, l'analisi AI è disponibile nelle versioni della piattaforma rilasciate a partire dal 22 aprile 2026.

Come funziona

Quando richiedi un'analisi AI, Elastic Beanstalk esegue uno script su un'istanza del tuo ambiente che raccoglie gli eventi recenti, lo stato dell'istanza e i log (fino a 170.000 token di dati). Invia quindi questi dati ad Amazon Bedrock nel tuo account e restituisce approfondimenti e passaggi successivi consigliati.

Prerequisiti

Prima di utilizzare l'analisi AI, verifica che il tuo ambiente soddisfi i seguenti requisiti:

- Ambiente che esegue una versione della piattaforma supportata
- Profilo dell'istanza con le autorizzazioni richieste (vedi Autorizzazioni richieste sotto)
- Dettagli dei casi d'uso antropici (aree commerciali): nelle aree commerciali, l'analisi dell'intelligenza artificiale utilizza modelli Anthropic Claude tramite Amazon Bedrock. Anthropic richiede l'invio di un modulo monouso con i dettagli dei casi d'uso prima di poter richiamare i relativi modelli. Per inviare questo modulo, seleziona un modello Anthropic dal catalogo modelli nella console Amazon Bedrock o chiama l'`PutUseCaseForModelAccess`API. Devi eseguire questa operazione solo una volta per account. AWS Se invii il modulo dall'account di gestione AWS Organizations, questo copre automaticamente tutti gli account dei membri dell'organizzazione. Per ulteriori informazioni, consulta Access Amazon Bedrock Foundation Models.
  - Anche dopo aver inviato il modulo con i dettagli del caso d'uso, la prima chiamata di un modello Anthropic Claude nel tuo account richiede le autorizzazioni AWS Marketplace per completare un abbonamento automatico al modello. Se il profilo dell'istanza non dispone di queste autorizzazioni, l'analisi AI fallisce con il seguente errore:

    ```
    AccessDeniedException: Model access is denied due to IAM user or
    service role is not authorized to perform the required AWS Marketplace
    actions (aws-marketplace:ViewSubscriptions, aws-marketplace:Subscribe)
    to enable access to this model.
    ```

    Per risolvere il problema `aws-marketplace:Subscribeaws-marketplace:Unsubscribe`, aggiungi le `aws-marketplace:ViewSubscriptions` autorizzazioni al tuo profilo di istanza. Queste autorizzazioni sono necessarie solo per la prima chiamata in un account. Dopo aver abilitato il modello, puoi rimuoverli. Per ulteriori informazioni, consulta Concedere le autorizzazioni IAM per richiedere l'accesso ai modelli Amazon Bedrock Foundation.
- GovCloud regioni: se utilizzi regioni AWS GovCloud (Stati Uniti), l'analisi AI utilizza il modello NVIDIA Nemotron anziché Anthropic Claude. È necessario richiedere l'accesso al modello

NVIDIA Nemotron 3 Super 120B in Amazon Bedrock prima di utilizzare l'analisi AI. Per richiedere l'accesso, apri la pagina di accesso ad Amazon Bedrock Model nella GovCloud console e richiedi l'accesso a NVIDIA Nemotron 3 Super 120B. Il modulo con i dettagli del caso d'uso di Anthropic e le autorizzazioni AWS Marketplace non sono necessari per GovCloud le aree geografiche. Per ulteriori informazioni sull'abilitazione dell'accesso ai modelli in GovCloud, consulta Accedere ai modelli Amazon Bedrock Foundation in AWS GovCloud (Stati Uniti).

Autorizzazioni richieste

Per utilizzare l'analisi AI, il profilo dell'istanza Amazon EC2 per il tuo ambiente deve disporre delle autorizzazioni per richiamare Amazon Bedrock. Aggiungi le seguenti autorizzazioni al tuo profilo di istanza:

- `bedrock:InvokeModel`
- `bedrock:ListFoundationModels`
- `elasticbeanstalk:DescribeEvents`
- `elasticbeanstalk:DescribeEnvironmentHealth`

Per ulteriori informazioni sulla configurazione dei profili di istanza, consulta. Gestione dei profili dell'istanza Elastic Beanstalk

> **Note**
>
> Nelle aree commerciali, se è la prima volta che viene richiamato un modello Anthropic Claude nel tuo account, avrai bisogno anche delle autorizzazioni AWS Marketplace (`aws-marketplace:Subscribe`,, `aws-marketplace:Unsubscribe` and). `aws-marketplace:ViewSubscriptions` Queste autorizzazioni non sono richieste nelle regioni. GovCloud Per informazioni dettagliate, vedi Prerequisiti.

Utilizzo dell'analisi AI nella console

Dalla panoramica dell'ambiente

Quando lo stato di salute dell'ambiente è Avviso, Degrado o Grave, nella sezione panoramica dell'ambiente viene visualizzato un pulsante di analisi AI. Fai clic su questo pulsante per avviare un'analisi AI del tuo ambiente.

Dalla pagina Logs

Puoi anche accedere all'analisi AI dalla pagina Registri nel riquadro di navigazione. Fai clic sul pulsante AI Analysis per richiedere un' AI-powered analisi dello stato attuale del tuo ambiente.

Utilizzo dell'analisi AI con AWS CLI

Puoi utilizzare l'API Elastic Beanstalk AWS CLI tramite per richiedere e recuperare l'analisi AI a livello di codice.

Richiedi l'analisi AI

Usa l'`RequestEnvironmentInfo`operazione con il `InfoType` parametro impostato su`analyze`.

Example AWS CLI - Richiedi l'analisi AI

```
aws elasticbeanstalk request-environment-info \
    --environment-name my-env \
    --info-type analyze \
    --region us-east-1
```

Recupera l'analisi AI

Utilizzate l'`RetrieveEnvironmentInfo`operazione con il `InfoType` parametro impostato per `analyze` recuperare i risultati dell'analisi.

Example AWS CLI - Recupera l'analisi AI

```
aws elasticbeanstalk retrieve-environment-info \
    --environment-name my-env \
    --info-type analyze \
    --region us-east-1
```

La risposta include un' AI-generated analisi dello stato attuale dell'ambiente, insieme alle soluzioni consigliate per eventuali problemi identificati.

Utilizzo dell'analisi AI con l'EB CLI

Se utilizzi l'EB CLI, puoi richiedere l'analisi AI con l'opzione `--analyze` (`-ai`) del eb logs comando. Il comando richiede l'analisi, attende il completamento e visualizza i risultati.

Example EB CLI - Richiedi analisi AI

```
$ eb logs --analyze
```

L'`--analyze`opzione non è compatibile con`--instance`, `--all``--zip`, o`--log-group`. Per il riferimento completo ai comandi, vedereeb logs.

> **Note**
>
> L'`--analyze`opzione richiede EB CLI versione 3.27 o successiva.

Considerazioni importanti

- Prezzi: l'analisi AI utilizza Amazon Bedrock per elaborare i dati dell'ambiente e per le chiamate dei modelli si applicano i prezzi standard di Amazon Bedrock. Per i dettagli sui prezzi, consultare Prezzi di Amazon Bedrock.
- Requisiti della piattaforma: l'analisi AI è disponibile nelle versioni della piattaforma basate su Amazon Linux 2 e AL2023 rilasciate a partire dal 26 febbraio 2026. Per le piattaforme Windows Server, l'analisi AI è disponibile nelle versioni della piattaforma rilasciate a partire dal 22 aprile 2026. Per utilizzare questa funzionalità, aggiorna l'ambiente a una versione della piattaforma supportata. Per ulteriori informazioni, consulta Aggiornamento della versione della piattaforma dell'ambiente Elastic Beanstalk.
- Autorizzazioni: prima di utilizzare l'analisi AI, assicurati che il profilo dell'istanza disponga delle autorizzazioni Amazon Bedrock (and`bedrock:ListFoundationModels`) `bedrock:InvokeModel` e delle autorizzazioni Elastic Beanstalk (e) richieste. `elasticbeanstalk:DescribeEvents` `elasticbeanstalk:DescribeEnvironmentHealth`
- Privacy dei dati: l'analisi invia gli eventi e i log dell'ambiente ad Amazon Bedrock del tuo account per l'elaborazione. Per informazioni su come Amazon Bedrock gestisce i tuoi dati, consulta Amazon Bedrock Security and Compliance.
- Quote di servizio: l'analisi dell'intelligenza artificiale utilizza i modelli Amazon Bedrock Foundation, che hanno quote predefinite per richieste al minuto e token al minuto. Nelle regioni commerciali, vengono utilizzati modelli Anthropic Claude. Nelle GovCloud regioni, viene utilizzato il modello NVIDIA Nemotron. Se riscontri errori di limitazione, puoi richiedere un aumento della quota. Per ulteriori informazioni, consulta la sezione Richiesta di un aumento di quota.

Versioni delle piattaforma supportate

L'analisi AI è supportata nelle versioni delle piattaforme basate su Amazon Linux 2 e AL2023 rilasciate a partire dal 26 febbraio 2026. Per le piattaforme Windows Server, l'analisi AI è supportata nelle versioni della piattaforma rilasciate a partire dal 22 aprile 2026. Per verificare la versione della tua piattaforma, consulta le note di rilascio di Elastic Beanstalk.

Gestione degli allarmi

Questo argomento illustra i passaggi per creare allarmi per le metriche che stai monitorando. Fornisce inoltre istruzioni per visualizzare gli allarmi esistenti e controllarne lo stato.

Puoi creare allarmi per i parametri che monitori mediante la console Elastic Beanstalk. Gli allarmi consentono di monitorare le modifiche all' AWS Elastic Beanstalk ambiente in modo da poter identificare e mitigare facilmente i problemi prima che si verifichino. Ad esempio, puoi impostare un allarme che genera una notifica quando l'utilizzo della CPU in un ambiente supera una determinata soglia, in modo da essere avvertito prima che si verifichi un potenziale problema. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon CloudWatch.

> **Note**
>
> Elastic CloudWatch Beanstalk lo utilizza per il monitoraggio e gli allarmi CloudWatch , il che significa che i costi vengono applicati AWS all'account per tutti gli allarmi utilizzati.

Per ulteriori informazioni sul monitoraggio di specifici parametri, consulta Report dello stato di base.

Per controllare lo stato degli allarmi

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione, seleziona Alarms (Allarmi).

   La pagina visualizza un elenco di allarmi esistenti. Se alcuni allarmi sono in stato di allarme, vengono contrassegnati con l'icona di avviso ().

4. Per filtrare gli allarmi, scegliere il menu a discesa e quindi selezionare un filtro.
5. Per modificare o eliminare un avviso, scegli rispettivamente l'icona di modifica
   ( )
   o l'icona di eliminazione
   ( ).

Per creare un allarme

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, scegli Monitoring (Monitoraggio).
4. Individua la metrica per la quale desideri creare un allarme, quindi scegli l'icona della sveglia ().
   Viene visualizzata la pagina Add alarm (Aggiungi allarme).
5. Inserisci i dettagli sull'allarme:
   - Name (Nome): un nome per l'allarme.
   - Description (Descrizione) (facoltativo): una breve descrizione dell'allarme.
   - Period (Periodo): l'intervallo di tempo tra le letture.
   - Threshold (Soglia): descrive il comportamento e il valore che il parametro deve superare per attivare un allarme.
   - Change state after (Cambia stato dopo): il periodo di tempo successivo al superamento della soglia dopo il quale lo stato dell'allarme cambia.
   - Notify (Notifica): l'argomento di Amazon SNS che viene avvertito quando lo stato di un allarme cambia.
   - Notify when state changes to (Notifica quando lo stato diventa):
     - OK: il parametro rientra nella soglia definita.
     - Alarm (Allarme): il parametro supera la soglia definita.
     - Insufficient data (Dati insufficienti): l'allarme è stato appena attivato, il parametro non è disponibile o la quantità di dati non è sufficiente per determinare lo stato dell'allarme.
6. Scegliere Aggiungi. Lo stato dell'ambiente diventa grigio durante gli aggiornamenti dell'ambiente. È possibile visualizzare l'avviso creato scegliendo Alarms (Allarmi) nel riquadro di navigazione.

Visualizzazione della cronologia delle modifiche di un ambiente Elastic Beanstalk

Questo argomento spiega come utilizzare Elastic Beanstalk Console per visualizzare una cronologia delle modifiche alla configurazione che sono state apportate agli ambienti Elastic Beanstalk.

Elastic Beanstalk recupera la cronologia delle modifiche dagli eventi registrati in AWS CloudTrail e le visualizza in un elenco in cui puoi facilmente navigare e che puoi filtrare.

Il pannello Cronologia delle modifiche visualizza le seguenti informazioni per le modifiche apportate agli ambienti:

- La data e l'ora in cui è stata apportata una modifica
- L'utente IAM responsabile di una modifica apportata
- Lo strumento di origine (interfaccia a riga di comando di Elastic Beanstalk (EB CLI) o console) utilizzato per apportare la modifica
- Il parametro di configurazione e i nuovi valori impostati

I dati sensibili che fanno parte della modifica, ad esempio i nomi degli utenti del database interessati dalla modifica, non vengono visualizzati nel pannello.

Per visualizzare la cronologia delle modifiche

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel riquadro di navigazione, scegli Change history (Cronologia delle modifiche).

   La pagina Cronologia delle modifiche mostra un elenco delle modifiche alla configurazione apportate agli ambienti Elastic Beanstalk.

Tieni presente i seguenti punti sulla navigazione tra le informazioni in questa pagina:

- È possibile scorrere l'elenco scegliendo < (precedente) o > (successivo) oppure scegliendo un numero di pagina specifico.
- Nella colonna Modifiche alla configurazione, seleziona l'icona a forma di freccia per passare dall'espansione alla compressione dell'elenco delle modifiche sotto l'intestazione Modifiche apportate.

- Utilizza la barra di ricerca per filtrare i risultati dall'elenco della cronologia delle modifiche. È possibile immettere qualsiasi stringa per restringere l'elenco delle modifiche visualizzate.

Tieni presente quanto segue per filtrare i risultati visualizzati:

- Il filtro di ricerca non distingue tra maiuscole e minuscole.
- È possibile filtrare le modifiche visualizzate in base alle informazioni nella colonna Modifiche alla configurazione, anche quando non è visibile a causa della compressione all'interno delle modifiche apportate.
- È possibile filtrare solo i risultati visualizzati. Tuttavia, il filtro rimane attivo anche se scegli di passare a un'altra pagina per visualizzare più risultati. I risultati filtrati vengono aggiunti anche al set di risultati della pagina successiva.

Negli esempi seguenti viene illustrato come filtrare i dati mostrati nella schermata precedente:

- Entra **`GettingStartedApp-env`** nella casella di ricerca per restringere i risultati e includere solo le modifiche apportate all'ambiente denominato GettingStartedApp-env.
- Immetti **`example3`** nella casella di ricerca per limitare i risultati in modo da includere solo le modifiche apportate dagli utenti IAM il cui nome utente contiene la stringa esempio3.
- Immetti **`2020-10`** nella casella di ricerca per limitare i risultati in modo da includere solo le modifiche apportate nel mese di ottobre 2020. Modifica il valore di ricerca su **`2020-10-16`** per filtrare ulteriormente i risultati visualizzati in modo da includere solo le modifiche apportate il giorno del 16 ottobre 2020.
- Immetti **`proxy:staticfiles`** nella casella di ricerca per limitare i risultati in modo da includere solo le modifiche apportate allo spazio dei nomi denominato aws:elasticbeanstalk:environment:proxy:staticfiles. Le righe visualizzate sono il risultato dell'applicazione del filtro. Questo è valido anche per i risultati che sono compressi all'interno delle modifiche apportate.

Visualizzazione del flusso di eventi di un ambiente Elastic Beanstalk

Questo argomento spiega come accedere agli eventi e alle notifiche associati all'applicazione.

Visualizzazione degli eventi con la console Elastic Beanstalk

Per visualizzare gli eventi con la console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel pannello di navigazione seleziona Events (Eventi).

   La pagina Events (Eventi) mostra un elenco di tutti gli eventi registrati per l'ambiente. È possibile sfogliare l'elenco scegliendo **<** previous (precedente), **>** next (successivo) o numeri di pagina. Puoi filtrare gli eventi in base al tipo usando l'elenco a discesa Severity (Gravità).

Visualizzazione degli eventi con gli strumenti da riga di comando

La CLI EB e AWS CLI forniscono entrambe i comandi per il recupero di eventi. Se gestisci l'ambiente utilizzando la CLI EB, utilizza eb events per stampare un elenco di eventi. Questo comando include anche l'opzione `--follow`, che continua a mostrare nuovi eventi finché non premi CTRL+C per interrompere l'output.

Per recuperare gli eventi utilizzando AWS CLI, usa il `describe-events` comando e specifica l'ambiente per nome o ID:

```
$ aws elasticbeanstalk describe-events --environment-id e-gbjzqccra3
{
    "Events": [
        {
            "ApplicationName": "elastic-beanstalk-example",
            "EnvironmentName": "elasticBeanstalkExa-env",
            "Severity": "INFO",
            "RequestId": "a4c7bfd6-2043-11e5-91e2-9114455c358a",
            "Message": "Environment update completed successfully.",
            "EventDate": "2015-07-01T22:52:12.639Z"
        },
...
```

Per ulteriori informazioni sugli strumenti da riga di comando, vedere???.

Creazione di elenchi e connessioni alle istanze dei server

Questo argomento spiega come visualizzare un elenco delle EC2 istanze Amazon che eseguono l'ambiente applicativo Elastic Beanstalk e come connettersi ad esse.

Puoi visualizzare un elenco di EC2 istanze Amazon che eseguono il tuo ambiente AWS Elastic Beanstalk applicativo tramite la console Elastic Beanstalk. È possibile connettersi a queste istanze tramite qualsiasi client SSH. È possibile connettere le istanze in esecuzione su Windows tramite Remote Desktop.

> **Important**
>
> Prima di poter accedere alle istanze Amazon fornite da Elastic Beanstalk, devi creare una coppia di chiavi EC2 Amazon e configurare le istanze Amazon fornite da Elastic Beanstalk per utilizzare la EC2 coppia di chiavi Amazon. EC2instances EC2 Puoi configurare le tue coppie di EC2 chiavi Amazon utilizzando la console AWS di gestione. Per istruzioni sulla creazione di una coppia di key pair per Amazon EC2, consulta la Amazon EC2 Getting Started Guide. Per ulteriori informazioni su come configurare le EC2 istanze Amazon per l'utilizzo di una coppia di EC2 chiavi Amazon, consultaCoppia di chiavi EC2.
> Per impostazione predefinita, Elastic Beanstalk non abilita le EC2 connessioni remote alle istanze in un contenitore Windows ad eccezione di quelle nei contenitori Windows legacy. (Elastic EC2 Beanstalk configura le istanze nei contenitori Windows legacy per utilizzare la porta 3389 per le connessioni RDP.) È possibile abilitare le connessioni remote alle EC2 istanze che eseguono Windows aggiungendo una regola a un gruppo di sicurezza che autorizza il traffico in entrata verso le istanze. È vivamente consigliabile rimuovere la regola al termine della connessione in remoto. Puoi aggiungere di nuovo la regola quando hai bisogno di accedere nuovamente da remoto. Per ulteriori informazioni, consulta Aggiunta di una regola per il traffico RDP in entrata verso un'istanza di Windows e Connessione all'istanza Windows nella Guida per l'utente di Amazon Elastic Compute Cloud per Microsoft Windows.

Per visualizzare e connettersi alle EC2 istanze Amazon per un ambiente

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione della console, seleziona Load Balancers (Sistemi di bilanciamento del carico).

3. Il nome dei sistemi di bilanciamento del carico creati da Elastic Beanstalk include awseb. Trova il sistema di bilanciamento del carico per il tuo ambiente e fai clic su di esso.
4. Scegli la scheda Instances (Istanze) nel riquadro inferiore della console.

   Viene visualizzato un elenco delle istanze che il sistema di bilanciamento del carico utilizza per l'ambiente Elastic Beanstalk. Annota un ID dell'istanza a cui desideri connetterti.
5. Nel riquadro di navigazione della EC2 console Amazon, scegli Istanze e trova l'ID dell'istanza nell'elenco.
6. Fai clic con il pulsante destro del mouse sull'ID dell' EC2 istanza Amazon in esecuzione nel sistema di bilanciamento del carico del tuo ambiente, quindi seleziona Connect dal menu contestuale.
7. Annota l'indirizzo DNS pubblico dell'istanza nella scheda Description (Descrizione).
8. Connettiti a un'istanza che esegue Linux utilizzando il client SSH di tua scelta, quindi digita ssh -i .ec2/mykeypair.pem EC2-user@<public-DNS - >. of-the-instance

Per ulteriori informazioni sulla connessione a un'istanza Amazon EC2 Linux, consulta Getting Started with Amazon EC2 Linux Instances nella Amazon EC2 User Guide.

Se il tuo ambiente Elastic Beanstalk utilizza la piattaforma .NET on Windows Server, consulta la sezione Getting Started with EC2 Amazon Windows Instances nella Amazon User Guide. EC2

Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk

Questo argomento spiega i tipi di log delle istanze forniti da Elastic Beanstalk. Fornisce inoltre istruzioni dettagliate per recuperarli e gestirli.

Le istanze Amazon EC2 nell'ambiente Elastic Beanstalk generano log che puoi utilizzare per risolvere i problemi relativi ai file dell'applicazione o di configurazione. I log creati dal server Web, dal server delle applicazioni, dagli script CloudFormation della piattaforma Elastic Beanstalk e vengono archiviati localmente su singole istanze. Puoi recuperarle con facilità utilizzando la console di gestione dell'ambiente o EB CLI. Puoi anche configurare il tuo ambiente per lo streaming dei log su Amazon CloudWatch Logs in tempo reale.

I log di coda includono le ultime 100 righe dei file di log più utilizzati, ovvero log operativi di Elastic Beanstalk e log generati dal server Web o dal server di applicazioni. Quando richiedi i log di coda

nella console di gestione dell'ambiente o con eb logs, un'istanza nel tuo ambiente concatena le voci di log più recenti in unico file di testo e carica il file in Amazon S3.

I registri di bundle sono registri completi per un'ampia gamma di file di log, inclusi i log di yum e cron e vari log di CloudFormation. Quando richiedi i log di bundle, un'istanza nel tuo ambiente comprime i file di log completi in un archivio ZIP e carica il file in Amazon S3.

Per caricare i log ruotati in Amazon S3, le istanze nell'ambiente devono disporre di un profilo di istanza con l'autorizzazione di scrittura sul bucket Elastic Beanstalk in Amazon S3. Queste autorizzazioni sono incluse nel profilo dell'istanza predefinito che Elastic Beanstalk chiede di creare quando un ambiente viene avviato nella console Elastic Beanstalk per la prima volta.

Per recuperare i log delle istanze

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione scegli Logs (Log).
4. Scegliere Request Logs (Registri richieste), quindi scegliere il tipo di registri da recuperare. Per ottenere i log di coda, scegli Last 100 Lines (Ultime 100 righe). Per ottenere i log di bundle, scegli Full Logs (Log completi).
5. Quando Elastic Beanstalk termina di recuperare i log, scegli Download (Scarica).

Elastic Beanstalk archivia i log di coda e bundle in un bucket Amazon S3 e genera un URL Amazon S3 prefirmato che puoi utilizzare per accedere ai tuoi log. Elastic Beanstalk elimina i file da Amazon S3 dopo 15 minuti.

> **Warning**
>
> Chiunque sia in possesso dell'URL Amazon S3 prefirmato può accedere ai file prima che siano eliminati. Assicurati di rendere l'URL disponibile solo a persone affidabili.

> **Note**
>
> La policy dell'utente deve disporre dell'autorizzazione `s3:DeleteObject`. Elastic Beanstalk utilizza le autorizzazioni utente per eliminare i log da Amazon S3.

Per conservare i log, puoi configurare l'ambiente in modo da pubblicare automaticamente i log in Amazon S3 dopo la rotazione. Per abilitare la rotazione dei log in Amazon S3, segui la procedura descritta in Configurazione della visualizzazione dei log delle istanze. Le istanze nell'ambiente tenteranno di caricare i log che sono stati ruotati una volta ogni ora.

Se l'applicazione genera log in una posizione che non è inclusa nella configurazione di default per la piattaforma dell'ambiente, puoi estendere tale configurazione utilizzando i file di configurazione (`.ebextensions`). Puoi aggiungere i file di log dell'applicazione ai log di coda, ai log di bundle o alla rotazione dei log.

Per lo streaming dei log in tempo reale e lo storage a lungo termine, configura il tuo ambiente per lo streaming dei log su Amazon CloudWatch Logs.

Per AI-powered analizzare i log dell'ambiente, gli eventi e lo stato delle istanze per identificare le cause principali e le soluzioni ai problemi di salute, consulta. AI-powered analisi dell'ambiente

Sections

- Posizione dei log sulle istanze Amazon EC2
- Posizione dei log in Amazon S3
- Impostazioni della rotazione dei log in Linux
- Estensione della configurazione predefinita per i task dei log
- Streaming di file di registro su Amazon CloudWatch Logs

Posizione dei log sulle istanze Amazon EC2

I log sono archiviati in posizioni standard sulle istanze Amazon EC2 del tuo ambiente. Elastic Beanstalk genera i seguenti log.

Amazon Linux 2

- `/var/log/eb-engine.log`

Amazon Linux AMI (AL1)

> **Note**
>
> Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate sull'AMI Amazon Linux (AL1) come ritirato. Per ulteriori informazioni sulla migrazione

a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

- `/var/log/eb-activity.log`
- `/var/log/eb-commandprocessor.log`

Windows Server

- `C:\Program Files\Amazon\ElasticBeanstalk\logs\`
- `C:\cfn\log\cfn-init.log`

Questi log contengono messaggi sulle attività di distribuzione, inclusi i messaggi relativi ai file di configurazione (`.ebextensions`).

Ogni server Web e di applicazione archivia i log nella relativa cartella:

- Apache – `/var/log/httpd/`
- IIS – `C:\inetpub\wwwroot\`
- Node.js – `/var/log/nodejs/`
- nginx – `/var/log/nginx/`
- Passenger – `/var/app/support/logs/`
- Puma – `/var/log/puma/`
- Python – `/opt/python/log/`
- Tomcat – `/var/log/tomcat/`

Posizione dei log in Amazon S3

Quando richiedi i log di coda o i log di bundle dal tuo ambiente o se le istanze caricano log ruotati, i log sono archiviati nel tuo bucket Elastic Beanstalk in Amazon S3. Elastic Beanstalk crea un `elasticbeanstalk-`*`region`*`-`*`account-id` bucket denominato AWS per ogni regione in cui vengono creati gli ambienti. In questo bucket, i log sono archiviati in `resources/environments/logs/`*`logtype`*`/`*`environment-id`*`/`*`instance-id`*.

Ad esempio, i log dell'istanza`i-0a1fd158`, nell' AWS ambiente Elastic `e-mpcwnwheky` Beanstalk `us-west-2` nella regione dell'account, vengono archiviati `123456789012` nelle seguenti posizioni:

- Log di coda:

  ```
  s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/tail/e-mpcwnwheky/i-0a1fd158
  ```

- Log di bundle:

  ```
  s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/bundle/e-mpcwnwheky/i-0a1fd158
  ```

- Log ruotati:

  ```
  s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/publish/e-mpcwnwheky/i-0a1fd158
  ```

> **Note**
>
> Puoi trovare il tuo ID ambiente nella console di gestione dell'ambiente.

Elastic Beanstalk elimina automaticamente i log di coda e bundle da Amazon S3 15 minuti dopo la loro creazione. I log ruotati persistono finché non li elimini o li sposti in Amazon Glacier.

Impostazioni della rotazione dei log in Linux

Sulle piattaforme Linux, Elastic Beanstalk usa `logrotate` per ruotare periodicamente i log. Dopo la rotazione in locale di un log, l'attività di rotazione lo seleziona e lo carica in Amazon S3, se previsto dalla configurazione. Per impostazione predefinita, i log ruotati in locale non sono visualizzati nei log di coda o di bundle.

I file di configurazione Elastic Beanstalk per `logrotate` sono disponibili in `/etc/logrotate.elasticbeanstalk.hourly/`. Queste impostazioni di rotazione sono specifiche per la piattaforma e potrebbero cambiare nelle versioni future della piattaforma. Per ulteriori informazioni sulle impostazioni e sugli esempi di configurazione disponibili, esegui `man logrotate`.

I file di configurazione sono richiamati da processi cron in `/etc/cron.hourly/`. Per ulteriori informazioni su `cron`, utilizza `man cron`.

Estensione della configurazione predefinita per i task dei log

Elastic Beanstalk utilizza file in sottocartelle di `/opt/elasticbeanstalk/tasks` (Linux) o `C:\Program Files\Amazon\ElasticBeanstalk\config` (Windows Server) sull'istanza Amazon EC2 per configurare le attività per i log di coda, i log di bundle e la rotazione dei log.

Su Amazon Linux:

- Log di coda:

  `/opt/elasticbeanstalk/tasks/taillogs.d/`
- Log di bundle:

  `/opt/elasticbeanstalk/tasks/bundlelogs.d/`
- Log ruotati:

  `/opt/elasticbeanstalk/tasks/publishlogs.d/`

Su Windows Server:

- Log di coda:

  `c:\Program Files\Amazon\ElasticBeanstalk\config\taillogs.d\`
- Log di bundle:

  `c:\Program Files\Amazon\ElasticBeanstalk\config\bundlelogs.d\`
- Log ruotati:

  `c:\Program Files\Amazon\ElasticBeanstalk\config\publogs.d\`

Ad esempio, il file `eb-activity.conf` in Linux aggiunge due file di log all'attività per i log di coda.

`/opt/elasticbeanstalk/tasks/taillogs.d/eb-activity.conf`

```
/var/log/eb-commandprocessor.log
/var/log/eb-activity.log
```

Puoi utilizzare i file di configurazione dell'ambiente (`.ebextensions`) per aggiungere i tuoi file `.conf` a queste cartelle. Un file `.conf` elenca i file di log specifici della tua applicazione aggiunti da Elastic Beanstalk alle attività per i file di log.

Utilizza la sezione `files` per aggiungere file di configurazione ai task che desideri modificare. Ad esempio, il testo della configurazione seguente aggiunge un file di configurazione del log a ogni istanza nel tuo ambiente. Questo file di configurazione del log, `cloud-init.conf`, aggiunge `/var/log/cloud-init.log` ai log di coda.

```
files:
  "/opt/elasticbeanstalk/tasks/taillogs.d/cloud-init.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/log/cloud-init.log
```

Aggiungi questo testo a un file con estensione `.config` nel bundle di origine in una cartella denominata `.ebextensions`.

```
~/workspace/my-app
|-- .ebextensions
|   `-- tail-logs.config
|-- index.php
`-- styles.css
```

Sulle piattaforme Linux, puoi anche utilizzare caratteri jolly nelle configurazioni per i task dei log. Questo file di configurazione aggiunge tutti i file con l'estensione `.log` dalla cartella `log` nella directory principale dell'applicazione ai log di bundle.

```
files:
  "/opt/elasticbeanstalk/tasks/bundlelogs.d/applogs.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/app/current/log/*.log
```

Le configurazioni per i task dei log non supportano i caratteri jolly sulle piattaforme Windows.

> **Note**
>
> Per acquisire familiarità con le procedure di personalizzazione dei log, è possibile distribuire un'applicazione di esempio utilizzando l'interfaccia della riga di comando EB. A tale scopo, la CLI EB crea una directory dell'applicazione locale che contiene una sottodirectory `.ebextentions` con una configurazione di esempio. È inoltre possibile utilizzare i file di registro dell'applicazione di esempio per esplorare la funzionalità di recupero dei log descritta in questo argomento.

Per ulteriori informazioni sull'utilizzo dei file di configurazione, consulta Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Come per i log di coda e i log di bundle, puoi estendere la rotazione dei log utilizzando un file di configurazione. Ogni volta che Elastic Beanstalk ruota i propri log e li carica su Amazon S3, ruota e carica anche i tuoi log aggiuntivi. Il comportamento dell'estensione della rotazione dei log varia a seconda del sistema operativo della piattaforma. Le sezioni seguenti descrivono i due casi.

Estensione della rotazione dei log in Linux

Come spiegato in Impostazioni della rotazione dei log in Linux, Elastic Beanstalk utilizza `logrotate` per ruotare i log sulle piattaforme Linux. Quando configuri i file di log della tua applicazione per la rotazione, l'applicazione non deve creare copie dei file di log. Elastic Beanstalk configura `logrotate` per creare una copia dei file di log dell'applicazione per ogni rotazione. Pertanto, l'applicazione deve mantenere i file di log sbloccati quando non scrive attivamente su tali log.

Estensione della rotazione dei log in Windows Server

In Windows Server, quando configuri i file di log della tua applicazione per la rotazione dei log, l'applicazione deve ruotare i file di log periodicamente. Elastic Beanstalk cerca file con nomi che iniziano con il modello configurato e li seleziona per caricarli in Amazon S3. Inoltre, i punti nei nomi di file vengono ignorati e Elastic Beanstalk considera che il nome fino al punto è il nome di file di log di base.

Elastic Beanstalk carica tutte le versioni di un file di log di base eccetto quella più recente, in quanto la considera come il file di log attivo dell'applicazione, che può eventualmente essere bloccato. L'applicazione può quindi mantenere il file di log attivo bloccato tra le rotazioni.

Ad esempio, l'applicazione scrive su un file di log denominato `my_log.log` e tu indichi questo nome nel file `.conf`. L'applicazione ruota periodicamente il file. Durante il ciclo di rotazione di Elastic Beanstalk, trova i seguenti file nella cartella del file di log: `my_log.log`, `my_log.0800.log`, `my_log.0830.log`. Elastic Beanstalk considera tutti questi file come versioni del nome di base `my_log`. Il file `my_log.log` presenta l'ora di modifica più recente, quindi Elastic Beanstalk carica solo gli altri due file, ovvero `my_log.0800.log` e `my_log.0830.log`.

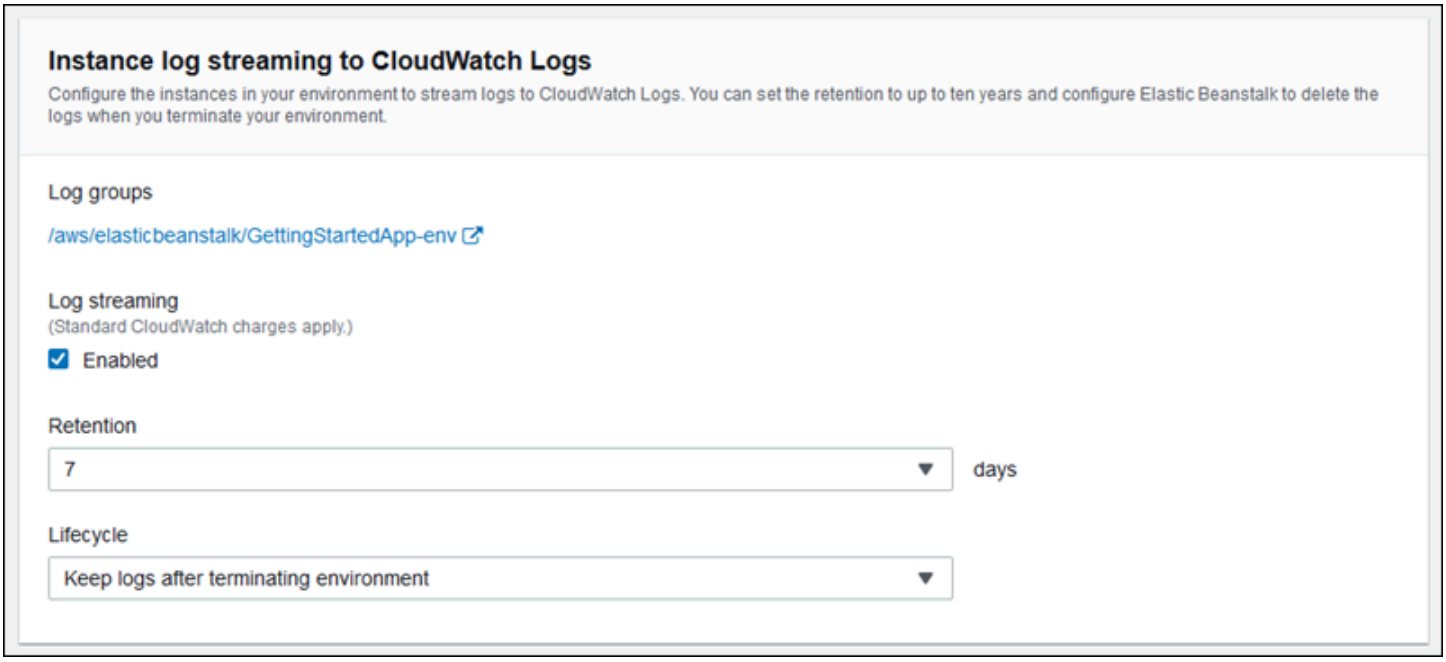
Streaming di file di registro su Amazon CloudWatch Logs

Puoi configurare il tuo ambiente per lo streaming dei log su Amazon CloudWatch Logs nella console Elastic Beanstalk o utilizzando le opzioni di configurazione. Con CloudWatch Logs, ogni istanza del tuo ambiente trasmette i log a gruppi di log che puoi configurare per essere conservati per settimane o anni, anche dopo la chiusura dell'ambiente.

Il set di log trasmessi in streaming varia a seconda dell'ambiente, ma include sempre `eb-engine.log` e i log di accesso del server proxy nginx o Apache che viene eseguito prima dell'applicazione.

Quando lo streaming dei log è abilitato, puoi visualizzare questi gruppi di log direttamente nella console Elastic Beanstalk, nella pagina Logs dell'ambiente. Scegli un gruppo di log per aprire i relativi eventi di registro in un visualizzatore integrato, dove puoi passare da un flusso di log all'altro e cercare e filtrare gli eventi.

Puoi configurare lo streaming dei log nella console Elastic Beanstalk durante la creazione dell'ambiente o per un ambiente esistente. È possibile impostare le seguenti opzioni dalla console: abilitare /disabilitare lo streaming dei log su CloudWatch Logs, impostare il numero di giorni di conservazione e selezionare una delle opzioni del ciclo di vita. In questo esempio, i log vengono salvati per un massimo di sette giorni, anche quando l'ambiente viene terminato.



Il seguente file di configurazione attiva lo streaming dei log con una conservazione di 180 giorni, anche se l'ambiente viene terminato.

Example. ebextensions/log-streaming.config

```
option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:
    StreamLogs: true
    DeleteOnTerminate: false
    RetentionInDays: 180
```

Visualizzazione dei log di distribuzione per un ambiente Elastic Beanstalk

Elastic Beanstalk genera un registro di distribuzione per ogni distribuzione nell'ambiente. Il registro di distribuzione fornisce una visione cronologica consolidata di ciò che è accaduto durante una distribuzione, tra cui l'installazione delle dipendenze, l'output della build, l'avvio dell'applicazione e gli eventuali errori riscontrati. È possibile utilizzare i log di distribuzione per diagnosticare rapidamente le distribuzioni non riuscite senza dover inserire SSH nelle istanze o correlare più file di registro.

I log di distribuzione vengono scritti localmente su ogni istanza. Per le distribuzioni attivate tramite console, CLI, API o aggiornamenti gestiti, un'istanza carica continuamente il proprio registro su

Amazon S3 durante la distribuzione. La console Elastic Beanstalk legge il registro da Amazon S3, così puoi monitorare l'avanzamento senza connetterti all'istanza.

I log di distribuzione sono progettati per essere concisi. In caso di successo, il registro mostra solo messaggi di riepilogo (ad esempio, quali comandi sono stati eseguiti e completati). In caso di errore, il registro include fino a 50 righe di output della fase non riuscita, in modo da poter visualizzare l'errore senza passare al setaccio l'output dettagliato.

> **Note**
>
> I log di distribuzione sono disponibili nelle versioni della piattaforma Amazon Linux 2 e Amazon Linux 2023 rilasciate a partire dall'11 marzo 2026 e nelle versioni della piattaforma Windows rilasciate a partire dal 22 aprile 2026.

Operazioni supportate

I log di distribuzione vengono generati per le seguenti operazioni:

- Distribuzioni delle applicazioni: distribuzione di una nuova versione dell'applicazione nell'ambiente.
- Aggiornamenti della configurazione: modifica delle impostazioni di configurazione dell'ambiente applicate alle istanze esistenti e aggiornamenti che aggiungono nuove istanze all'ambiente.
- Creazione dell'ambiente: la distribuzione iniziale quando si crea un nuovo ambiente.
- Riavvia l'app server: riavvio del server delle applicazioni sulle istanze.
- Aggiornamenti gestiti della piattaforma: aggiornamenti della piattaforma che Elastic Beanstalk applica automaticamente durante una finestra di manutenzione pianificata.

Le operazioni che non modificano lo stato dell'applicazione o della configurazione sulle istanze, come la richiesta di log, lo scambio di CNAME o l'aggiornamento dei tag, non generano registri di distribuzione.

Contenuto del registro di distribuzione

Un registro di distribuzione acquisisce le seguenti informazioni durante una distribuzione:

- Ciclo di vita della distribuzione: messaggi di inizio e completamento per ogni fase di distribuzione, ad esempio e. `Starting Application deployment` `Completed Application deployment`

- Output.ebextensions: in caso di successo, i nomi dei comandi eseguiti. In caso di errore, vengono visualizzate le ultime 50 righe di `cfn-init` output per facilitare la diagnosi del problema.
- Output degli hook della piattaforma: in caso di successo, i nomi degli script hook eseguiti. In caso di errore, vengono emesse le ultime 50 righe dell'hook.
- Installazione in base alle dipendenze: output di gestori di pacchetti come npm installpip install,composer install, ebundle install. In caso di successo, viene registrato solo un messaggio di completamento. In caso di errore, vengono incluse le ultime 50 righe di output.
- Output di compilazione: output di comandi di compilazione come docker buildgo build, e build Java. In caso di errore, vengono incluse le ultime 50 righe di output.
- Output di avvio dell'applicazione: output iniziale dell'applicazione dopo l'avvio. La fonte dipende dalla piattaforma in uso:
  - Docker: registri del contenitore da o docker logs docker compose logs
  - Java SE, Go, Python Node.js, Ruby, .NET — Elabora i log stdout
  - Uscita del registro Tomcat — Catalina
  - PHP: log degli errori del PHP-FPM master e del pool
  - ECS: registri dei container provenienti da ogni contenitore di attività

> **Note**
>
> L'output dell'applicazione viene acquisito a partire da 2 secondi dopo l'avvio dell'applicazione. Sono inclusi solo i messaggi di avvio iniziali: se l'applicazione impiega più tempo a produrre l'output, questo non verrà visualizzato nel registro di distribuzione. Per visualizzare i log completi dell'applicazione, richiedi i log dei pacchetti o connettiti direttamente all'istanza. Per ulteriori informazioni, consulta Visualizzazione dei log dalle istanze.

Quando una fase di distribuzione fallisce, il registro la contrassegna con `[ERROR]` e include fino a 50 righe di output della fase non riuscita. Se il registro di distribuzione non contiene dettagli sufficienti, è possibile recuperare i registri completi dell'istanza (inclusi **`eb-engine.log`** i registri delle applicazioni) dalla scheda Registri. **`eb-hooks.log`** Per ulteriori informazioni, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Visualizzazione dei log di distribuzione nella console

La console Elastic Beanstalk offre una scheda Distribuzioni nella dashboard dell'ambiente in cui è possibile visualizzare la cronologia e i log di distribuzione. La cronologia delle implementazioni mostra le implementazioni degli ultimi 42 giorni (6 settimane).

Visualizzazione della cronologia delle distribuzioni

Per visualizzare la cronologia delle distribuzioni

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nella dashboard dell'ambiente, scegli la scheda Distribuzioni.

   La scheda Distribuzioni mostra una tabella delle distribuzioni per l'ambiente. Ogni riga include le seguenti informazioni:

   - ID richiesta: l'identificatore univoco per la distribuzione.
   - Stato: riuscita, non riuscita o in corso.
   - Tipo: il tipo di distribuzione, ad esempio Environment Creation, Application Deployment, Configuration Update, Managed Platform Update, Restart App Server, Rebuild Environment, Restore Environment, Swap Environment Domain o Terminate Environment.
   - Politica: la politica di distribuzione, ad esempio All at once, Rolling, Rolling with additional batch, Immutable o Traffic splitting.
   - Ora di inizio: quando è iniziata la distribuzione.
   - Durata: quanto tempo è stato necessario per completare l'implementazione.

Quando è in corso una distribuzione, la scheda richiede automaticamente gli aggiornamenti. Puoi anche scegliere il pulsante di aggiornamento per ricaricare manualmente l'elenco.

Visualizzazione dei dettagli e dei registri di distribuzione

Per visualizzare i dettagli della distribuzione

1. Nella scheda Distribuzioni, scegli il link Request ID per la distribuzione che desideri ispezionare.

2. La pagina dei dettagli della distribuzione mostra una sezione di riepilogo con l'ID della richiesta, lo stato, il tipo di distribuzione, l'ora di inizio, la durata e la politica di distribuzione. La politica di distribuzione (ad esempio, All at once, Rolling, Rolling with additional batch, Immutable o Traffic splitting) viene visualizzata quando può essere determinata dagli eventi di distribuzione.
3. Sotto il riepilogo, scegli una delle seguenti schede:
   - Eventi: una cronologia degli eventi relativi a questa distribuzione, filtrata per mostrare solo gli eventi per la distribuzione selezionata.
   - Registri di distribuzione: il registro di distribuzione consolidato dell'istanza. È possibile cercare, filtrare per livello di registro e scaricare il file di registro.

Per le distribuzioni in corso, la scheda dei registri si aggiorna automaticamente per mostrare le nuove voci di registro man mano che vengono scritte. Al termine di una distribuzione, la console recupera lo stato finale del registro per garantire la visualizzazione dell'output completo.

> **Important**
>
> La visualizzazione dei log di distribuzione nella console richiede l'`s3:GetObject`autorizzazione sul `elasticbeanstalk-`*`region`*`-`*`account-id` bucket di storage Amazon S3 dell'ambiente (). Se la tua policy IAM non include questa autorizzazione, la cronologia e gli eventi di distribuzione saranno ancora disponibili, ma la scheda dei log mostrerà un errore.

File di registro di distribuzione sulle istanze

I log di distribuzione vengono scritti `/var/log/deployments/` nella directory di ogni istanza. Il nome del file di registro dipende da come è stata attivata la distribuzione:

- Workflow-controlled distribuzioni (attivate tramite console, CLI o API): *`request-id`* dov'è l'ID univoco della `eb-deployment-`*`request-id`*`.log` richiesta di distribuzione.
- Self-startup implementazioni (avvio dell'istanza) —`eb-deployment-`*`timestamp`*`-`*`instance-id`*`.log`, dove *`timestamp`* è in formato UTC (ad esempio`20260317T151315Z`) ed *`instance-id`* è l'ID dell'istanza Amazon EC2.

Elastic Beanstalk ruota automaticamente questi file, mantenendo i 50 registri di distribuzione più recenti su ogni istanza.

Per le distribuzioni controllate dal flusso di lavoro, il log viene caricato su Amazon S3 nel seguente percorso:

```
s3://elasticbeanstalk-region-account-id/resources/environments/logs/
deployments/environment-id/log-filename
```

Per le distribuzioni ad avvio automatico, il log viene caricato su Amazon S3 in una sottodirectory: `selfstartup/`

```
s3://elasticbeanstalk-region-account-id/resources/environments/logs/
deployments/environment-id/selfstartup/log-filename
```

Per le distribuzioni controllate dal flusso di lavoro, la prima istanza che inizia il caricamento rivendica il ruolo dell'intera distribuzione. Tale istanza carica il registro su Amazon S3 per tutta la durata della distribuzione. Per le distribuzioni ad avvio automatico, ogni istanza carica il proprio registro in modo indipendente. Tutte le istanze continuano a scrivere i log di distribuzione localmente.

> **Important**
>
> Il caricamento dei log di distribuzione su Amazon S3 `s3:PutObject` richiede l'autorizzazione sul bucket di storage Amazon S3 dell'ambiente nel profilo dell'istanza e la configurazione VPC deve consentire la connettività ad Amazon S3.

I caricamenti dei log di distribuzione sono limitati a 1 MB per file. Se un log di distribuzione supera questa dimensione, la versione caricata viene troncata con un messaggio che indica che il log completo è disponibile sull'istanza.

Disabilitazione dei caricamenti dei log S3

Per evitare che i log di distribuzione vengano caricati su Amazon S3, imposta la seguente proprietà di ambiente sul tuo ambiente:

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  EB_DEPLOYMENT_LOG_S3_DISABLED
    value:  true
```

Quando questa proprietà di ambiente è impostata, i log di distribuzione vengono ancora scritti localmente `/var/log/deployments/` su ogni istanza, ma non vengono caricati su Amazon S3 e non saranno disponibili nella scheda Distribuzioni della console. È inoltre possibile impostare questa proprietà nella pagina Configurazione in Software oppure utilizzando l'EB CLI o. AWS CLI

Utilizzo di Elastic Beanstalk con altri servizi AWS

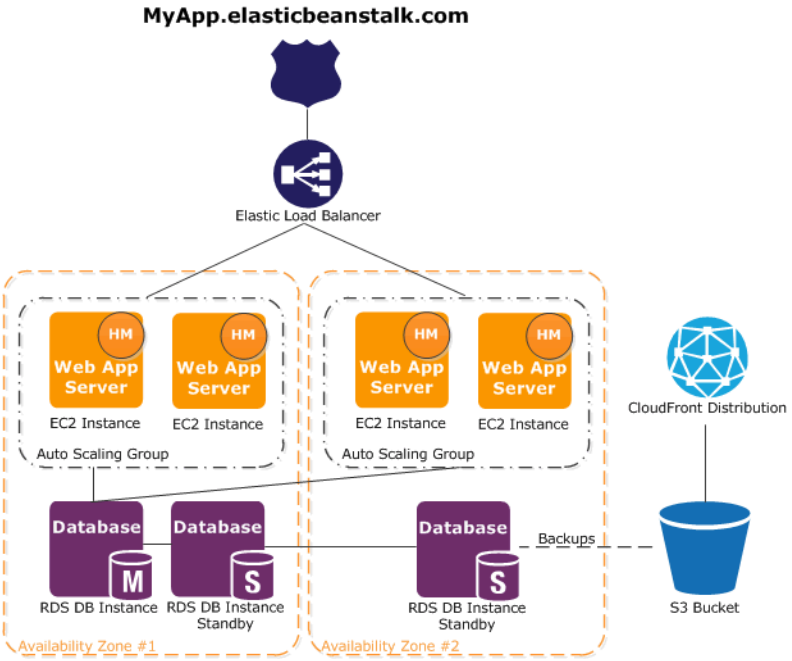
Gli argomenti di questa sezione descrivono i molti modi in cui è possibile utilizzare AWS servizi aggiuntivi con l'applicazione Elastic Beanstalk. Per implementare gli ambienti dell'applicazione, Elastic Beanstalk gestisce le risorse AWS di altri servizi o ne utilizza le funzionalità. Inoltre, Elastic Beanstalk AWS si integra con servizi che non utilizza direttamente come parte dei tuoi ambienti.

Argomenti

- Panoramica dell'architettura
- Utilizzo di Elastic Beanstalk con Amazon CloudFront
- Registrazione delle chiamate API Elastic Beanstalk con AWS CloudTrail
- Utilizzo di Elastic Beanstalk con Amazon CloudWatch
- Utilizzo di Elastic CloudWatch Beanstalk con Amazon Logs
- Utilizzo di Elastic Beanstalk con Amazon EventBridge
- Ricerca e tracciamento delle risorse Elastic Beanstalk con AWS Config
- Utilizzo di Elastic Beanstalk con Amazon DynamoDB
- Utilizzo di Elastic Beanstalk con Amazon ElastiCache
- Utilizzo di Elastic Beanstalk con Amazon Elastic File System
- Utilizzo di Elastic Beanstalk con AWS Identity and Access Management
- Utilizzo di Elastic Beanstalk con Amazon RDS
- Utilizzo di Elastic Beanstalk con Amazon S3
- Utilizzo di Elastic Gestione dei segreti AWS AWS Systems Manager Beanstalk con e Parameter Store
- Utilizzo di Elastic Beanstalk con Amazon VPC

Panoramica dell'architettura

Il diagramma seguente illustra un esempio di architettura di Elastic Beanstalk su più zone di disponibilità che funziona con altri prodotti AWS come Amazon CloudFront, Amazon Simple Storage Service (Amazon S3) e Amazon Relational Database Service (Amazon RDS).



Per pianificare la tolleranza agli errori, è consigliabile avere n istanze Amazon EC2 +1 e suddividere le istanze in più zone di disponibilità. Nell'improbabile caso in cui una zona di disponibilità sia inutilizzabile, avrai ancora altre istanze Amazon EC2 in esecuzione in un'altra zona di disponibilità. Puoi regolare Amazon EC2 Auto Scaling per consentire un numero minimo di istanze e più zone di disponibilità. Per istruzioni su come eseguire questa operazione, consultare Auto Scaling delle istanze dell'ambiente Elastic Beanstalk. Per ulteriori informazioni sulla creazione di applicazioni con tolleranza ai guasti, consulta Creazione di applicazioni con tolleranza ai guasti in AWS.

Nelle sezioni seguenti viene illustrata in modo più dettagliato l'integrazione con Amazon CloudFront, Amazon CloudWatch, Amazon DynamoDB Amazon, ElastiCache Amazon RDS, Amazon Route 53, Amazon Simple Storage Service, Amazon VPC e IAM.

Utilizzo di Elastic Beanstalk con Amazon CloudFront

Amazon CloudFront è un servizio web che velocizza la distribuzione di contenuti web statici e dinamici, ad esempio .html, .css, .php, immagini e file multimediali, agli utenti finali. CloudFront distribuisce i tuoi contenuti attraverso una rete mondiale di edge location. Quando un utente finale richiede i contenuti che utilizzi CloudFront, viene indirizzato all'edge location che offre la latenza più bassa, in modo che i contenuti vengano forniti con le migliori prestazioni possibili. Se i contenuti si trovano già in tale edge location, CloudFront li distribuisce immediatamente. Se il contenuto non si trova attualmente in quella edge location, lo CloudFront recupera da un bucket Amazon S3 o da un

server HTTP (ad esempio un server Web) che hai identificato come origine per la versione definitiva dei tuoi contenuti.

Dopo aver creato e distribuito l'applicazione Elastic Beanstalk, puoi registrarti e CloudFront iniziare a utilizzarla per distribuire CloudFront i tuoi contenuti. Scopri CloudFront di più nella Amazon CloudFront Developer Guide.

Registrazione delle chiamate API Elastic Beanstalk con AWS CloudTrail

Elastic Beanstalk è AWS CloudTrail integrato con, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o AWS servizio in Elastic Beanstalk. CloudTrailacquisisce tutte le chiamate API per Elastic Beanstalk come eventi, incluse le chiamate dalla console Elastic Beanstalk, dall'EB CLI e dal codice a Elastic Beanstalk. APIs Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Elastic Beanstalk. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella console nella cronologia degli eventi. CloudTrail Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a Elastic Beanstalk, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

Informazioni su Elastic Beanstalk in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in Elastic Beanstalk, tale attività viene registrata CloudTrail in un evento AWS insieme ad altri eventi di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo account. AWS Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Elastic Beanstalk, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail è valido per tutte le Regioni . Il percorso registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. Per ulteriori informazioni, consulta:

- Panoramica della creazione di un trail

- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di Elastic Beanstalk vengono CloudTrail registrate e documentate nell'API Reference.AWS Elastic Beanstalk Ad esempio, le chiamate alle `ListTagsForResource` azioni `DescribeApplicationsUpdateEnvironment`, e generano voci nei file di registro. CloudTrail

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Informazioni sulle voci dei file di log di Elastic Beanstalk

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra l'`UpdateEnvironment`azione eseguita da un utente IAM denominato`intern`, per l'`sample-env`ambiente dell'`sample-app`applicazione.

```
{
  "Records": [{
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
```

```
      "principalId": "AIXDAYQEXAMPLEUMLYNGL",
      "arn": "arn:aws:iam::123456789012:user/intern",
      "accountId": "123456789012",
      "accessKeyId": "ASXIAGXEXAMPLEQULKNXV",
      "userName": "intern",
      "sessionContext": {
        "attributes": {
          "mfaAuthenticated": "false",
          "creationDate": "2016-04-22T00:23:24Z"
        }
      },
      "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2016-04-22T00:24:14Z",
    "eventSource": "elasticbeanstalk.amazonaws.com",
    "eventName": "UpdateEnvironment",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "255.255.255.54",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
      "applicationName": "sample-app",
      "environmentName": "sample-env",
      "optionSettings": []
    },
    "responseElements": null,
    "requestID": "84ae9ecf-0280-17ce-8612-705c7b132321",
    "eventID": "e48b6a08-c6be-4a22-99e1-c53139cbfb18",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
  }]
}
```


Utilizzo di Elastic Beanstalk con Amazon CloudWatch

Amazon ti CloudWatch consente di monitorare, gestire e pubblicare diverse metriche, nonché di configurare azioni di allarme in base ai dati delle metriche. CloudWatch Il monitoraggio di Amazon ti consente di raccogliere, analizzare e visualizzare i parametri di sistema e applicazioni in modo da poter prendere decisioni operative e aziendali più rapidamente e con maggiore sicurezza.

Puoi usare Amazon CloudWatch per raccogliere parametri sulle tue risorse Amazon Web Services (AWS), come le prestazioni delle tue istanze Amazon. EC2 Puoi anche pubblicare le tue metriche direttamente su Amazon CloudWatch. Gli CloudWatch allarmi Amazon ti aiutano a implementare le

decisioni più facilmente consentendoti di inviare notifiche o apportare automaticamente modifiche alle risorse che stai monitorando, in base a regole che definisci. Ad esempio, puoi creare allarmi che avviano azioni di Amazon Auto EC2 Scaling e Amazon Simple Notification Service (Amazon SNS) per tuo conto.

Elastic Beanstalk CloudWatch utilizza automaticamente Amazon per aiutarti a monitorare lo stato dell'applicazione e dell'ambiente. Puoi accedere alla CloudWatch console Amazon per visualizzare la dashboard e ottenere una panoramica di tutte le tue risorse e dei tuoi allarmi. Puoi inoltre scegliere di visualizzare più parametri o aggiungere parametri personalizzati.

Per ulteriori informazioni su Amazon CloudWatch, consulta l'Amazon CloudWatch Developer Guide. Per un esempio di come usare Amazon CloudWatch con Elastic Beanstalk, consulta. the section called “Esempio: utilizzo di CloudWatch parametri Amazon personalizzati”

Utilizzo di Elastic CloudWatch Beanstalk con Amazon Logs

Questo argomento spiega le funzionalità di monitoraggio che il servizio Amazon CloudWatch Logs può fornire a Elastic Beanstalk. Inoltre, illustra la configurazione ed elenca le posizioni dei log per ogni piattaforma Elastic Beanstalk.

L'implementazione CloudWatch dei log può consentire di eseguire le seguenti attività di monitoraggio:

- Monitora e archivia l'applicazione Elastic Beanstalk, il sistema e i file di log personalizzati dalle istanze Amazon EC2 dei tuoi ambienti.
- Configura allarmi che ti consentano di reagire più facilmente a specifici eventi del flusso di log estratti dai filtri metrici.

L'agente CloudWatch Logs installato su ogni istanza Amazon EC2 nel tuo ambiente pubblica i punti dati metrici sul servizio per ogni gruppo di log CloudWatch che configuri. Ogni gruppo di log applica i propri modelli di filtro per determinare a quali eventi del flusso di log inviare CloudWatch come punti dati. I flussi del log appartenenti allo stesso gruppo di log condividono le stesse impostazioni di conservazione, monitoraggio e controllo degli accessi. Puoi configurare Elastic Beanstalk per trasmettere automaticamente i log CloudWatch al servizio, come descritto in. Trasmissione dei log delle istanze su Logs CloudWatch Per ulteriori informazioni sui CloudWatch log, inclusi la terminologia e i concetti, consulta la Amazon CloudWatch Logs User Guide.

Oltre ai log delle istanze, se abiliti l'integrità migliorata del tuo ambiente, puoi configurare l'ambiente in modo che trasmetta le informazioni sullo stato a Logs. CloudWatch Per informazioni, consulta

Streaming delle informazioni sullo stato dell'ambiente Elastic Beanstalk su Amazon Logs CloudWatch.

Argomenti

- Prerequisito, ad esempio lo streaming dei log su Logs CloudWatch
- Come Elastic Beanstalk configura i log CloudWatch
- Trasmissione dei log delle istanze su Logs CloudWatch
- Risoluzione dei problemi CloudWatch di integrazione dei log
- Streaming delle informazioni sullo stato dell'ambiente Elastic Beanstalk su Amazon Logs CloudWatch

Prerequisito, ad esempio lo streaming dei log su Logs CloudWatch

Se non disponi della policy gestita AWSElasticBeanstalkWebTiero di AWSElasticBeanstalkWorkerTierElastic Beanstalk nel profilo dell'istanza Elastic Beanstalk, devi aggiungere le seguenti autorizzazioni al tuo profilo per abilitare questa funzionalità.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents",
      "logs:CreateLogStream"
    ],
    "Resource": [
    "*"
    ]
  }
  ]
}
```

Come Elastic Beanstalk configura i log CloudWatch

Elastic Beanstalk CloudWatch installa un log agent con le impostazioni di configurazione predefinite su ogni istanza creata. Scopri di più nella documentazione dell'agente. CloudWatch

Quando abiliti lo streaming dei log delle istanze su CloudWatch Logs, Elastic Beanstalk invia i file di log dalle istanze dell'ambiente a Logs. CloudWatch Piattaforme diverse trasmettono log diversi. Nella tabella seguente sono elencati i log, suddivisi per piattaforma.

Piattaforma / Ramo della piattaforma	Log
Docker	• //eb-engine.log var/log • /var/log/eb-hooks.log • /var/log/docker • /var/log/docker-events.log • /var/log/eb- docker/containers /eb-corrente-.log app/stdouterr • var/log/nginx/access.log • /var/log/.log nginx/error
ECS su Docker	• /var/log/docker-events.log • /var/log/eb-ecs-mgr.log • /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/ecs/ecs-agent.log • /var/log/ecs/ecs-init.log
Go .NET Core su Linux Java	• /var/log/eb-engine.log • /var/log/eb-hooks.log • var/log/web.stdout.log • var/log/.log nginx/access • /var/log/.log nginx/error
Node.js	• //eb-engine.log var/log • /var/log/eb-hooks.log

Piattaforma / Ramo della piattaforma	Log
Python	• var/log/web.stdout.log • /var/log/httpd/access_log • /var/log/httpd/error_log • var/log/.log nginx/access • /var/log/.log nginx/error
Tomcat PHP	• //eb-engine.log var/log • /var/log/eb-hooks.log • /var/log/httpd/access_log • /var/log/httpd/error_log • var/log/nginx/access.log • /var/log/.log nginx/error
.NET on Windows Server	• C:\inetpub\logs\\ LogFiles W3SVC1\ u_ex*.log • C:\Program Files\ Amazon\ElasticBeanstalk\ logs\ AWSDeployment.log • C:\Program Files\ Amazon\ElasticBeanstalk\ logs\ Hooks.log
Ruby	• //eb-engine.log var/log • /var/log/eb-hooks.log • var/log/puma/puma.log • //web.stdout.log var/log • var/log/.log nginx/access • /var/log/.log nginx/error

File di log sulle piattaforme AMI Amazon Linux

> **Note**
>
> Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate sull'AMI Amazon Linux (AL1) come ritirato. Per ulteriori informazioni sulla migrazione

a una ramificazione della piattaforma Amazon Linux 2023 attuale e completamente supportata, consulta Migrazione dell'applicazione Elastic Beanstalk Linux ad Amazon Linux 2023 o Amazon Linux 2.

Nella tabella seguente sono elencati i file di log trasmessi dalle istanze sulle ramificazioni della piattaforma basate su AMI Amazon Linux (precedente ad Amazon Linux 2), per piattaforma.

Piattaforma / Ramo della piattaforma	Log
Docker / Ramo della piattaforma: Docker in esecuzione su Amazon Linux (a 64 bit)	• //eb-activity.log var/log • var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • //.log var/log nginx/access • //eb- var/log docker/containers /eb-corrente-.log app/stdouterr
Docker / Ramo piattaforma: Docker a più container che esegue Amazon Linux (a 64 bit).	• var/log//eb-activity.log • /var/log/ecs/ecs-init.log • /var/log/eb-ecs-mgr.log • /var/log/ecs/ecs-agent.log • /var/log/docker-events.log
Glassfish (Docker preconfig urato)	• /var/log/eb-activity.log • var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • //.log var/log nginx/access
Go	• //eb-activity.log var/log • var/log/nginx/error.log • /var/log/.log nginx/access
Java /	• //eb-activity.log var/log

Piattaforma / Ramo della piattaforma	Log
Ramo della piattaforma: Java 8 in esecuzione su Amazon Linux (a 64 bit) Ramo della piattaforma: Java 7 in esecuzione su Amazon Linux (a 64 bit)	• var/log/nginx/access.log • /var/log/.log nginx/error • //web-1.error.log var/log • var/log//web-1.log
Tomcat	• /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log • /var/log/nginx/error_log • /var/log/nginx/access_log
Node.js	• /var/log/eb-activity.log • var/log/nodejs/nodejs.log • /var/log/.log nginx/error • /var/log/.log nginx/access • /var/log/.log httpd/error • /var/log/.log httpd/access
PHP	• //eb-activity.log var/log • /var/log/httpd/error_log • /var/log/httpd/access_log
Python	• /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log • opt/python/log/supervisord.log

Piattaforma / Ramo della piattaforma	Log
Ruby / Ramo della piattaforma: Puma con Ruby in esecuzione su Amazon Linux (a 64 bit)	• //eb-activity.log var/log • var/log/nginx/error.log • /var/log/.log puma/puma • /var/log/.log nginx/access
Ruby / Ramo della piattaforma: Passenger con Ruby in esecuzione su Amazon Linux a 64 bit	• //eb-activity.log var/log • /var/app/support/logs/passenger.log • /var/app/support/logs/access.log • /var/app/support/logs/error.log

Elastic Beanstalk configura i CloudWatch gruppi di log in Logs per i vari file di log trasmessi. Per recuperare file di log specifici da CloudWatch Logs, devi conoscere il nome del gruppo di log corrispondente. Lo schema di denominazione dei gruppi di log dipende dal sistema operativo della piattaforma.

Per le piattaforme Linux, aggiungi alla posizione del file di log nell'istanza il prefisso `/aws/elasticbeanstalk/`*`environment_name`* per ottenere il nome del gruppo di log. Per recuperare, ad esempio, il file `/var/log/nginx/error.log`, specifica il gruppo di log `/aws/elasticbeanstalk/`*`environment_name`*`/var/log/nginx/error.log`.

Per le piattaforme Windows, consulta la tabella seguente per il gruppo di log corrispondente a ogni file di log.

On-instance file di registro	Gruppo di log
`C:\Program Files\Amazon\ElasticBeanstalk\logs\AWSDeployment.log`	`/aws/elasticbeanstalk/<environment-name>/EBDeploy-Log`
`C:\Program Files\Amazon\ElasticBeanstalk\logs\Hooks.log`	`/aws/elasticbeanstalk/<environment-name>/EBHooks-Log`

On-instance file di registro	Gruppo di log
`C:\inetpub\logs\LogFiles` (tutta la directory)	`/aws/elasticbeanstalk/<environment-name>/IIS-Log`

Trasmissione dei log delle istanze su Logs CloudWatch

È possibile abilitare lo streaming dei log delle istanze su CloudWatch Logs utilizzando la console Elastic Beanstalk, l'EB CLI o le opzioni di configurazione.

Prima di abilitarlo, configura le autorizzazioni IAM da utilizzare con l'agente Logs. CloudWatch Puoi collegare la seguente policy personalizzata al profilo dell'istanza assegnato al tuo ambiente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Streaming di log delle istanze utilizzando la console Elastic Beanstalk

Per trasmettere i log delle istanze a Logs CloudWatch

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. In Streaming dei log delle istanze su Logs: CloudWatch
   - Abilita l'opzione Log streaming (Streaming di log).
   - Imposta il valore di Retention (Conservazione) sul numero di giorni per cui salvare i log.
   - Seleziona l'impostazione Lifecycle (Ciclo di vita), che determina se i log vengono salvati dopo che l'ambiente viene terminato.
6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Dopo aver abilitato lo streaming dei log, Elastic Beanstalk visualizza i gruppi di log in streaming CloudWatch nella console di gestione dell'ambiente, nella pagina Logs. Scegli un gruppo di log per visualizzarne gli eventi di registro in un visualizzatore integrato: Elastic Beanstalk seleziona automaticamente il flusso di log attivo più di recente e puoi cambiare stream, cercare e filtrare gli eventi e caricare gli eventi precedenti.

Streaming di log delle istanze tramite l'interfaccia a riga di comando EB

Per abilitare lo streaming dei log delle istanze su CloudWatch Logs utilizzando la CLI EB, utilizzare eb logs il comando.

```
$ eb logs --cloudwatch-logs enable
```

Puoi anche usarlo eb logs per recuperare i log dai log. CloudWatch È possibile recuperare tutti i log delle istanze dell'ambiente oppure usare le numerose opzioni del comando per specificare sottoinsiemi di log da recuperare. Il comando seguente permette ad esempio di recuperare il set completo di log delle istanze per l'ambiente e di salvare i log in una directory in `.elasticbeanstalk/logs`.

```
$ eb logs --all
```

In particolare, l'opzione `--log-group` permette di recuperare i log delle istanze per un determinato gruppo di log, corrispondente a un file di log nell'istanza specifico. A tale scopo, è necessario

conoscere il nome del gruppo di log che corrisponde al file di log da recuperare. Puoi trovare queste informazioni in Come Elastic Beanstalk configura i log CloudWatch.

Streaming di log delle istanze tramite file di configurazione

Quando si crea o si aggiorna un ambiente, è possibile utilizzare un file di configurazione per impostare e configurare lo streaming dei log delle istanze su Logs. CloudWatch Il file di configurazione di esempio seguente permette di abilitare lo streaming di log delle istanze predefinito. Elastic Beanstalk trasmette in streaming il set predefinito di file di log per la piattaforma dell'ambiente. Per usare l'esempio, copia il testo in un file con l'estensione `.config` nella directory `.ebextensions` al primo livello del bundle di origine dell'applicazione.

```
option_settings:
  - namespace: aws:elasticbeanstalk:cloudwatch:logs
    option_name: StreamLogs
    value: true
```

Streaming di file di log personalizzati

L' CloudWatch integrazione di Elastic Beanstalk con Logs non supporta direttamente lo streaming di file di registro personalizzati generati dall'applicazione. Per lo streaming dei log personalizzati, utilizza un file di configurazione per installare direttamente l' CloudWatch agente e configurare i file da inviare. Per un esempio di file di configurazione, consulta `logs-streamtocloudwatch-linux.config`.

> **Note**
>
> L'esempio non funziona sulla piattaforma Windows.

Per ulteriori informazioni sulla configurazione dei CloudWatch log, consulta il riferimento al file di configurazione dell'CloudWatch agente nella Amazon CloudWatch User Guide.

Risoluzione dei problemi CloudWatch di integrazione dei log

> **Prova Amazon Q Developer CLI per la risoluzione dei problemi AI-assisted**
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli

eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Risoluzione dei problemi degli ambienti Elastic Beanstalk con Amazon Q Developer CLI nei blog. AWS

Impossibile individuare i registri delle istanze dell'ambiente

Se non riesci a trovare alcuni dei log delle istanze dell'ambiente che ti aspetti in CloudWatch Logs, esamina i seguenti problemi comuni:

- Il ruolo IAM non dispone delle autorizzazioni IAM richieste.
- Hai avviato il tuo ambiente in un ambiente Regione AWS che non supporta CloudWatch i log.
- Uno dei file di log personalizzati non esiste nel percorso specificato.

Log delle applicazioni mancanti o intermittenti

Se i log dell'applicazione Elastic Beanstalk`/var/log/web.stdout.log`, (), sembrano mancanti o intermittenti, ciò può essere dovuto alle impostazioni di limitazione della velocità predefinite in rsyslog e journald. La disattivazione completa della limitazione della velocità può risolvere questo problema, ma non è consigliata in quanto potrebbe comportare un utilizzo eccessivo del disco, una potenziale interruzione del servizio o un peggioramento delle prestazioni del sistema in caso di interruzioni impreviste dei log. È invece possibile modificare i limiti di velocità utilizzando quanto segue. `.ebextensions configuration` Questa configurazione aumenta l'intervallo limite di velocità a 600 secondi con limiti di burst più elevati, garantendo un equilibrio tra la corretta registrazione e la protezione del sistema.

Problemi di limitazione

Se un'operazione Elastic Beanstalk che avvia contemporaneamente un gran numero di istanze restituisce un `Error: fail to create log stream: ThrottlingException: Rate exceeded` messaggio del tipo, si tratta di una limitazione da troppe chiamate all'API. CloudWatch

Per risolvere il problema della limitazione, esegui una delle seguenti azioni:

- Utilizzate un batch di dimensioni inferiori con distribuzioni continue per ridurre gli aggiornamenti simultanei.
- Richiedi un aumento della quota di servizio limite Transaction Per Second (TPS) del tuo AWS account per. CreateLogStream Per ulteriori informazioni, consulta CloudWatch Logs quotas e Managing your CloudWatch Logs service quote nella Amazon CloudWatch Logs User Guide.

Streaming delle informazioni sullo stato dell'ambiente Elastic Beanstalk su Amazon Logs CloudWatch

Se abiliti la reportistica avanzata sullo stato del tuo ambiente, puoi configurare l'ambiente per trasmettere informazioni sullo stato di salute a Logs. CloudWatch Questo streaming è indipendente da quello dei log delle istanze Amazon EC2. In questo argomento viene descritto lo streaming delle informazioni sullo stato dell'ambiente. Per informazioni sullo streaming di log delle istanze, consulta Utilizzo di Elastic CloudWatch Beanstalk con Amazon Logs.

Quando configuri lo streaming dello stato dell'ambiente, Elastic Beanstalk CloudWatch crea un gruppo di log Logs per lo stato dell'ambiente. Il nome del gruppo di log è `/aws/elasticbeanstalk/`*`environment-name`*`/environment-health.log`. In questo gruppo di log, Elastic Beanstalk crea flussi di log denominati *`YYYY-MM-DD`*`#`*`<hash-suffix>`* (potrebbero essere presenti più flussi di log per ogni data).

Quando lo stato dell'ambiente cambia, Elastic Beanstalk aggiunge un record al flusso di log sullo stato. Il record rappresenta la transizione dello stato, ovvero il nuovo stato e una descrizione della causa del cambiamento. Lo stato di un ambiente può ad esempio cambiare in Severe (Grave) a causa di un errore del sistema di bilanciamento del carico. Per una descrizione degli stati avanzati, consulta Colori e stati indicanti l'integrità.

Prerequisiti per lo streaming dello stato dell'ambiente su Logs CloudWatch

Per abilitare lo streaming dello stato dell'ambiente su CloudWatch Logs, è necessario soddisfare le seguenti condizioni:

- Piattaforma: è necessario usare una versione della piattaforma che supporta i report sullo stato avanzati.
- Autorizzazioni: è necessario concedere a Elastic Beanstalk determinate autorizzazioni per la creazione di log in modo che possa eseguire lo streaming delle informazioni sullo stato dell'ambiente per tuo conto. Se l'ambiente non usa un ruolo di servizio creato da Elastic Beanstalk, `aws-elasticbeanstalk-service-role`, oppure il ruolo collegato ai servizi del tuo account, `AWSServiceRoleForElasticBeanstalk`, assicurati di aggiungere le autorizzazioni seguenti al tuo ruolo di servizio personalizzato.

```
{
      "Effect": "Allow",
      "Action": [
```

```
        "logs:DescribeLogStreams",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*:log-stream:*"
}
```

Streaming dei registri relativi allo stato dell'ambiente su Logs CloudWatch

Puoi abilitare lo streaming dello stato dell'ambiente su CloudWatch Logs utilizzando la console Elastic Beanstalk, l'EB CLI o le opzioni di configurazione.

Streaming di log sullo stato dell'ambiente utilizzando la console Elastic Beanstalk

Per trasmettere i log sullo stato dell'ambiente a Logs CloudWatch

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Monitoring (Monitoraggio), scegliere Edit (Modifica).
5. In Health reporting (Report sullo stato) assicurati che l'opzione System (Sistema) per la creazione di report sia impostata su Enhanced (Avanzato).
6. In Health, streaming degli eventi su CloudWatch Logs
   - Abilita l'opzione Log streaming (Streaming di log).
   - Imposta il valore di Retention (Conservazione) sul numero di giorni per cui salvare i log.
   - Seleziona l'impostazione Lifecycle (Ciclo di vita), che determina se i log vengono salvati dopo che l'ambiente viene terminato.
7. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Dopo aver abilitato lo streaming dei log, il gruppo di log dello stato dell'ambiente (**`/aws/elasticbeanstalk/`*`environment-name`***`/environment-health.log`**) viene visualizzato nella console Elastic Beanstalk insieme ai gruppi di log dell'istanza, nella pagina Logs dell'ambiente. Sceglilo per visualizzare i relativi eventi di registro in un visualizzatore integrato.

Streaming di log sullo stato dell'ambiente tramite l'interfaccia a riga di comando EB

Per abilitare lo streaming dei registri relativi allo stato dell'ambiente su CloudWatch Logs utilizzando la CLI EB, utilizzare eb logs il comando.

```
$ eb logs --cloudwatch-logs enable --cloudwatch-log-source environment-health
```

È inoltre possibile utilizzare eb logs per recuperare i log da Logs. CloudWatch Il comando seguente permette ad esempio di recuperare tutti i log sullo stato per l'ambiente e di salvarli in una directory in `.elasticbeanstalk/logs`.

```
$ eb logs --all --cloudwatch-log-source environment-health
```

Streaming di log sullo stato dell'ambiente tramite file di configurazione

Quando si crea o si aggiorna un ambiente, è possibile utilizzare un file di configurazione per impostare e configurare lo streaming dei registri dello stato dell'ambiente su Logs. CloudWatch Per usare l'esempio seguente, copia il testo in un file con l'estensione `.config` nella directory `.ebextensions` al livello principale del bundle di origine dell'applicazione. L'esempio configura Elastic Beanstalk per abilitare lo streaming di log sullo stato dell'ambiente, conservare i log dopo che l'ambiente è stato terminato e salvarli per 30 giorni.

Example File di configurazione dello streaming di informazioni sullo stato

```
###########################################################################
##  Sets up Elastic Beanstalk to stream environment health information
##  to Amazon CloudWatch Logs.
##  Works only for environments that have enhanced health reporting enabled.
###########################################################################

option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:health:
    HealthStreamingEnabled: true
    ### Settings below this line are optional.
    # DeleteOnTerminate: Delete the log group when the environment is
    # terminated. Default is false. If false, the health data is kept
    # RetentionInDays days.
    DeleteOnTerminate: false
    # RetentionInDays: The number of days to keep the archived health data
    # before it expires, if DeleteOnTerminate isn't set. Default is 7 days.
```

```
    RetentionInDays: 30
```

Per le opzioni predefinite e i valori validi, consulta `aws:elasticbeanstalk:cloudwatch:logs:health`.

Utilizzo di Elastic Beanstalk con Amazon EventBridge

Con Amazon EventBridge, puoi configurare regole basate sugli eventi che monitorano le tue risorse Elastic Beanstalk e avviano azioni mirate che utilizzano altri servizi. AWS Ad esempio, puoi impostare una regola per l'invio di notifiche e-mail segnalando un argomento Amazon SNS ogni volta che lo stato di integrità di un ambiente di produzione cambia in uno stato di Avviso. In alternativa, è possibile impostare una funzione Lambda per passare una notifica a Slack ogni volta che lo stato dell'ambiente cambia in uno stato Danneggiato o Grave .

Puoi creare regole in Amazon EventBridge per agire su uno qualsiasi dei seguenti eventi Elastic Beanstalk:

- Modifiche dello stato per le operazioni dell'ambiente (comprese le operazioni di creazione, aggiornamento e terminazione). L'evento specifica se la modifica dello stato è stata avviata, completata o non completata.
- Modifiche dello stato per altre risorse. Oltre agli ambienti, altre risorse monitorate includono load balancer, gruppi Auto Scaling e istanze.
- Transizione dello stato per ambienti. L'evento indica dove lo stato dell'ambiente è passato da un stato di integrità ad un altro.
- Modifica dello stato per gli aggiornamenti gestiti. L'evento specifica se la modifica dello stato è stata avviata, completata o non completata.

Per acquisire eventi Elastic Beanstalk specifici che ti interessano, definisci EventBridge modelli specifici dell'evento che puoi utilizzare per rilevare gli eventi. I modelli di eventi hanno la stessa struttura degli eventi a cui corrispondono. Il modello cita i campi che desideri abbinare e fornisce i valori che stai cercando. Gli eventi vengono emessi secondo il principio del massimo sforzo. Vengono distribuiti da Elastic EventBridge Beanstalk quasi in tempo reale in normali circostanze operative. Tuttavia, possono verificarsi situazioni che potrebbero ritardare o impedire la consegna di un evento.

Per un elenco dei campi inclusi negli eventi Elastic Beanstalk e dei loro possibili valori di stringa, consulta Mappatura dei campi evento Elastic Beanstalk. Per informazioni su come EventBridge le regole funzionano con i pattern di eventi, consulta Eventi e pattern di eventi in. EventBridge

Monitora una risorsa Elastic Beanstalk con EventBridge

Con EventBridge, puoi creare regole che definiscono le azioni da intraprendere quando Elastic Beanstalk emette eventi per le sue risorse. Ad esempio, puoi creare una regola per ricevere un messaggio di posta elettronica ogni volta che lo stato di un ambiente cambia.

La EventBridge console dispone di un'opzione di pattern predefinito per la creazione di pattern di eventi Elastic Beanstalk. Se selezioni questa opzione nella EventBridge console quando crei una regola, puoi creare rapidamente un pattern di eventi Elastic Beanstalk. È sufficiente selezionare i campi evento e i valori. Man mano che effettui le selezioni, la console crea e visualizza il modello di eventi. In alternativa, è possibile modificare manualmente il modello di eventi creato e salvarlo come modello personalizzato. La console fornisce l'opzione di visualizzare un evento di esempio dettagliato che è possibile copiare e incollare nel modello di evento che stai creando.

Se preferisci digitare o copiare e incollare un pattern di eventi nella EventBridge console, puoi scegliere di utilizzare l'opzione Modello personalizzato nella console. In questo modo, non è necessario eseguire le fasi di selezione dei campi e dei valori descritte in precedenza. Questo argomento offre esempi sia di modelli di corrispondenza degli eventi che di eventi Elastic Beanstalk che puoi utilizzare.

Per creare una regola per un evento risorsa

1. Accedi AWS utilizzando un account con autorizzazioni di utilizzo EventBridge ed Elastic Beanstalk.
2. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
3. Nel pannello di navigazione, scegli Regole.
4. Scegli Crea regola.
5. Immettere un Name (Nome) per la regola e, facoltativamente, una descrizione.
6. Per Event bus (Bus di eventi), scegli default. Quando un AWS servizio del tuo account emette un evento, questo passa sempre al bus eventi predefinito del tuo account.
7. Per Tipo di regola, scegli Regola con un modello di eventi.
8. Scegli Next (Successivo).
9. Per Event source, scegli AWS eventi o eventi EventBridge partner.
10. (Facoltativo) Per Evento di esempio, seleziona AWS eventi. Insersici Elastic Beanstalk nel campo di ricerca. Questo fornirà un elenco di esempi di eventi Elastic Beanstalk da cui è possibile scegliere la visualizzazione. Questo passaggio visualizza semplicemente un evento di

esempio a cui è possibile fare riferimento. Non influisce sul risultato della creazione della regola. La sezione Esempio di eventi Elastic Beanstalk successiva in questo argomento consente di visualizzare esempi dello stesso tipo di eventi.

11. Nella sezione Modello di eventi, scegli Modulo di modello di eventi.

> **Note**
>
> Se disponi già del testo per un pattern di eventi e non hai bisogno che la EventBridge console lo crei per te, seleziona Pattern personalizzato (editor JSON). È quindi possibile immettere manualmente o copiare e incollare il testo nella casella Modello di evento . Scegli Successivo e vai al passaggio sull'inserimento di un obiettivo.

12. Per Origine evento, scegli Servizi AWS .
13. Per il servizio AWS seleziona Elastic Beanstalk.
14. Per Tipo di evento seleziona Modifica stato.
15. In questa fase viene illustrato come lavorare con i campi evento Elastic Beanstalk relativi a tipo di dettaglio, stato e gravità. Mentre scegli questi campi e i valori che desideri abbinare, la console crea e visualizza il modello di evento.

    - Se selezioni solo un valore per Tipi di dettaglio specifici, puoi scegliere uno o più valori per il campo successivo della gerarchia.
    - Se scegliete più di un valore per i tipi di dettaglio specifici, non scegliete valori specifici per i campi successivi della gerarchia. Questo impedisce la logica di corrispondenza ambigua tra i campi nel modello di evento.

    Il campo evento ambiente non è interessato da questa gerarchia, quindi viene visualizzato come descritto nella fase successiva.

16. Per Ambiente, seleziona Qualsiasi ambiente o Ambienti specifici.

    - Se selezioni Ambienti specifici, puoi scegliere uno o più ambienti dall'elenco a discesa. EventBridge aggiunge tutti gli ambienti selezionati all'interno dell'elenco EnvironmentName[] nella sezione dei dettagli del modello di evento. Quindi, la regola filtra tutti gli eventi in modo da includere solo gli ambienti specifici che scegli.
    - Se selezioni Any environment (Qualsiasi ambiente), nessun ambiente verrà aggiunto al tuo modello di evento. Per questo motivo, la regola non filtra gli eventi Elastic Beanstalk in base all'ambiente.

17. Scegli Next (Successivo).
18. Per Tipi di destinazione, scegli servizio AWS .
19. In Seleziona destinazioni, scegliere l'operazione di destinazione da eseguire quando viene ricevuto un evento di modifica dello stato della risorsa da Elastic Beanstalk.

    Ad esempio, è possibile utilizzare un argomento Amazon Simple Notification Service (SNS) per inviare un'e-mail o un messaggio di testo quando si verifica un evento. Per fare ciò, è necessario creare un argomento Amazon SNS utilizzando la console di Amazon SNS. Per ulteriori informazioni, consulta Utilizzo di Amazon SNS per notifiche all'utente.

    > **Important**
    >
    > Alcune operazioni di destinazione potrebbero richiedere l'uso di altri servizi e comportare costi aggiuntivi, ad esempio il servizio Amazon SNS o Lambda. Per ulteriori informazioni sui AWS prezzi, consultahttps://aws.amazon.com/pricing/. Alcuni servizi fanno parte del piano di utilizzo AWS gratuito. Se sei un nuovo cliente, puoi provare questi servizi gratuitamente. Per ulteriori informazioni, consultare https://aws.amazon.com/free/.

20. (Facoltativo) Scegli Aggiungi destinazione per specificare un'operazione di destinazione aggiuntiva per la regola evento.
21. Scegli Next (Successivo).
22. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
23. Scegli Next (Successivo).
24. Rivedi i dettagli della regola e scegli Crea regola.

Esempio di modelli di eventi Elastic Beanstalk

I modelli di eventi hanno la stessa struttura degli eventi a cui corrispondono. Il modello cita i campi che desideri abbinare e fornisce i valori che stai cercando.

- Modifica dello stato di integrità per tutti gli ambienti

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
```

```
  "detail-type": [
    "Health status change"
    ]
}
```

- Modifica dello stato di integrità per i seguenti ambienti: `myEnvironment1` e `myEnvironment2`. Questo modello di eventi filtra per questi due ambienti specifici, mentre il precedente esempio di modifica dello stato di integrità che non filtra invia eventi per tutti gli ambienti.

```
{"source": [
    "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Health status change"
    ],
    "detail": {
        "EnvironmentName": [
            "myEnvironment1",
            "myEnvironment2"
        ]
    }
}
```

- Modifica dello stato delle risorse Elastic Beanstalk per tutti gli ambienti

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Elastic Beanstalk resource status change"
    ]
}
```

- Modifica dello stato delle risorse Elastic Beanstalk con aggiornamento dell'ambiente non riuscito `Status` ed ERROR `Severity` per i seguenti ambienti: `myEnvironment1` e `myEnvironment2`

```
{"source": [
    "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Elastic Beanstalk resource status change"
    ],
```

```
    "detail": {
        "Status": [
            "Environment update failed"
            ],
        "Severity": [
            "ERROR"
            ],
        "EnvironmentName": [
            "myEnvironment1",
            "myEnvironment2"
        ]
    }
}
```

- Modifica dello stato delle altre risorse per i load balancer, i gruppi Auto Scaling e le istanze

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Other resource status change"
    ]
}
```

- Modifica dello stato degli aggiornamenti gestiti per tutti gli ambienti

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Managed update status change"
    ]
}
```

- Per acquisire tutti gli eventi da Elastic Beanstalk (escludere la sezione `detail-type`)

```
{
  "source": [
    "aws.elasticbeanstalk"
  ]
}
```

Esempio di eventi Elastic Beanstalk

Di seguito è riportato un esempio di evento Elastic Beanstalk per una modifica dello stato delle risorse:

```
{
   "version":"0",
   "id":"1234a678-1b23-c123-12fd3f456e78",
   "detail-type":"Elastic Beanstalk resource status change",
   "source":"aws.elasticbeanstalk",
   "account":"111122223333",
   "time":"2020-11-03T00:31:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:was:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/
myEnvironment"
   ],
   "detail":{
      "Status":"Environment creation started",
      "EventDate":1604363513951,
      "ApplicationName":"myApplication",
      "Message":"createEnvironment is starting.",
      "EnvironmentName":"myEnvironment",
      "Severity":"INFO"
   }
}
```

Di seguito è riportato un esempio di evento Elastic Beanstalk per una modifica dello stato:

```
{
   "version":"0",
   "id":"1234a678-1b23-c123-12fd3f456e78",
   "detail-type":"Health status change",
   "source":"aws.elasticbeanstalk",
   "account":"111122223333",
   "time":"2020-11-03T00:34:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:was:elasticbeanstalk:us-east-1:111122223333:environment/myApplication/
myEnvironment"
   ],
   "detail":{
      "Status":"Environment health changed",
```

```
      "EventDate":1604363687870,
      "ApplicationName":"myApplication",
      "Message":"Environment health has transitioned from Pending to Ok. Initialization
 completed 1 second ago and took 2 minutes.",
      "EnvironmentName":"myEnvironment",
      "Severity":"INFO"
   }
}
```

Mappatura dei campi evento Elastic Beanstalk

La tabella seguente mappa i campi evento di Elastic Beanstalk e i relativi possibili valori di stringa sul campo. EventBridge `detail-type` Per ulteriori informazioni su come EventBridge funziona con i modelli di eventi per un servizio, consulta Eventi e modelli di eventi in. EventBridge

EventBridge tipo di dettaglio del campo	Campo Elastic Beanstalk Stato	campo Elastic Beanstalk Gravità	Campo Elastic Beanstalk Messaggio
Modifica dello stato delle risorse Elastic Beanstalk	Creazione dell'ambiente avviata	INFO	createEnvironment in fase di avvio.
	Creazione dell'ambiente riuscita	INFO	createEnvironment completato correttamente.
	Creazione dell'ambiente riuscita	INFO	Ambiente avviato: <Nome ambiente>. Tuttavia, si sono verificati problemi durante l'avvio. Per ulteriori informazioni, vedere il log di eventi.
	Creazione ambiente non riuscita	ERRORE	Impossibile avviare ambiente.

EventBridge tipo di dettaglio del campo	Campo Elastic Beanstalk Stato	campo Elastic Beanstalk Gravità	Campo Elastic Beanstalk Messaggio
	Aggiornamento dell'ambiente avviato	INFO	Aggiornamento dell'ambiente in corso.
	Aggiornamento dell'ambiente riuscito	INFO	Aggiornamento dell'ambiente completato correttamente.
	Aggiornamento dell'ambiente non riuscito	ERRORE	Impossibile distribuire la configurazione.
	Terminazione dell'ambiente avviata	INFO	terminateEnvironment in corso.
	Terminazione dell'ambiente riuscita	INFO	terminateEnvironment completato correttamente.

EventBridge tipo di dettaglio del campo	Campo Elastic Beanstalk Stato	campo Elastic Beanstalk Gravità	Campo Elastic Beanstalk Messaggio
	Terminazione dell'ambiente non riuscita	INFO	La fase di terminazione dell'ambiente non è riuscita perché almeno uno dei flussi di lavoro di terminazione dell'ambiente non è riuscito.
Modifica dello stato delle altre risorse	Gruppo Auto Scaling creato	INFO	createEnvironment in fase di avvio.
	Gruppo Auto Scaling eliminato	INFO	createEnvironment in fase di avvio.
	Istanza aggiunta	INFO	Aggiunta istanza [i-123456789a12b1234] all'ambiente.
	Istanza rimossa	INFO	Rimossa istanza [i-123456789a12b1234] dall'ambiente.
	Load balancer creato	INFO	Load balancer creato denominato: <Nome LB>
	Load balancer eliminato	INFO	Load balancer eliminato denominato: <Nome LB>

EventBridge tipo di dettaglio del campo	Campo Elastic Beanstalk Stato	campo Elastic Beanstalk Gravità	Campo Elastic Beanstalk Messaggio
Modifica dello stato	Modificato stato dell'ambiente	INFO/WARN	Lo stato dell'ambiente è passato a <healthStatus>.
	Modificato stato dell'ambiente	INFO/WARN	Lo stato dell'ambiente è passato da <healthStatus> a <healthStatus>.
Modifica dello stato degli aggiornamenti gestiti	Avviato aggiornato gestito	INFO	L'aggiornamento della piattaforma gestito è in corso.
	Aggiornamento gestito non riuscito	INFO	Aggiornamento gestito non riuscito. Riprovare tra %s minuti.

Ricerca e tracciamento delle risorse Elastic Beanstalk con AWS Config

AWS Configfornisce una visualizzazione dettagliata della configurazione delle risorse nel tuo account. AWS AWS Puoi vedere in che modo le risorse sono correlate, ottenere una cronologia delle modifiche alla configurazione ed esaminare come cambiano le relazioni e le configurazioni nel tempo. È possibile utilizzare AWS Config per definire regole che valutano le configurazioni delle risorse per la conformità dei dati.

Diversi tipi di risorse Elastic Beanstalk sono integrati con: AWS Config

- Applicazioni
- Versioni delle applicazioni

- Ambienti

La sezione seguente mostra come configurare la registrazione AWS Config di risorse di questi tipi.

Per ulteriori informazioni in merito AWS Config, consulta la Guida per AWS Config gli sviluppatori. Per informazioni sui prezzi, consulta la pagina Prezzi di AWS Config.

Configurazione AWS Config

Per la configurazione iniziale AWS Config, consulta i seguenti argomenti nella Guida per gli AWS Config sviluppatori.

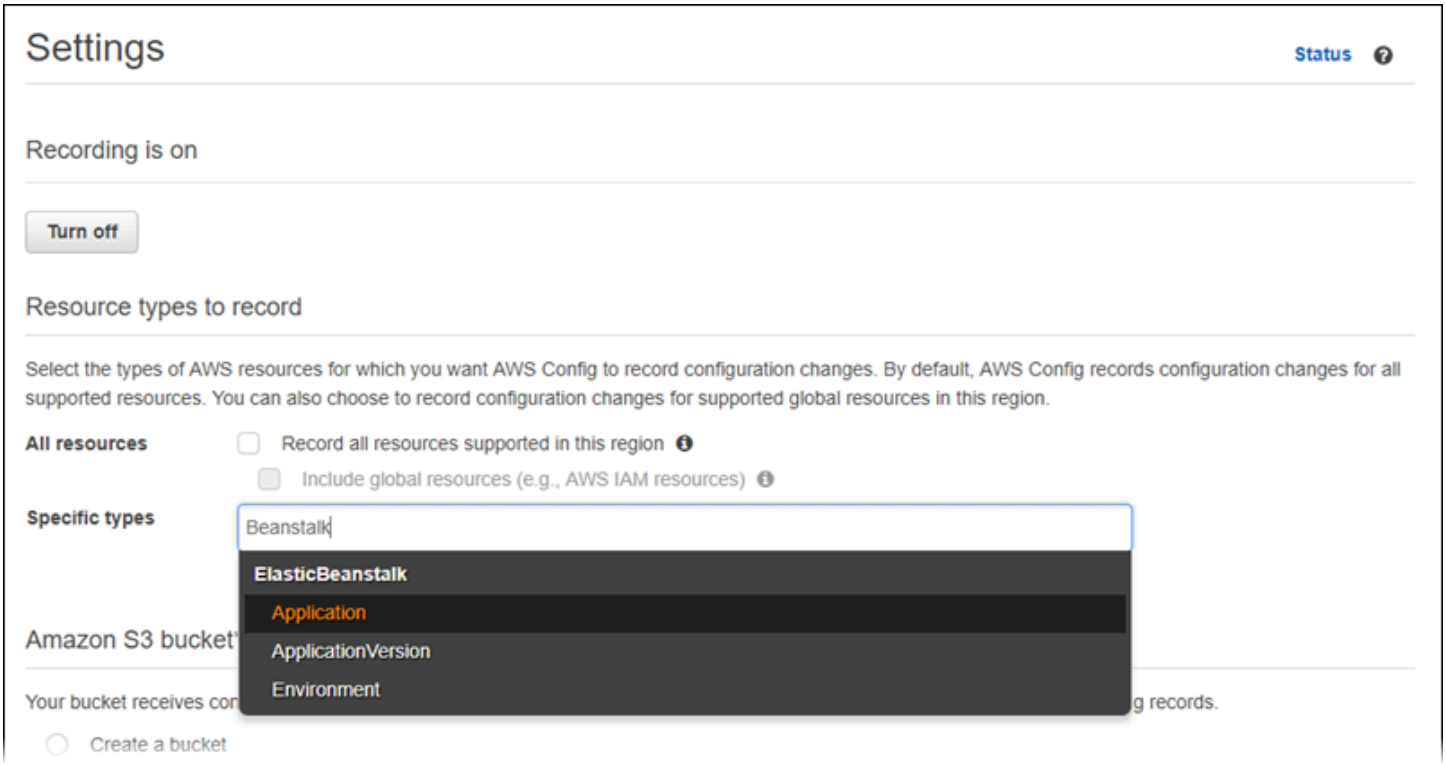
- Configurazione AWS Config con la console
- Configurazione AWS Config con AWS CLI

Configurazione AWS Config per registrare risorse Elastic Beanstalk

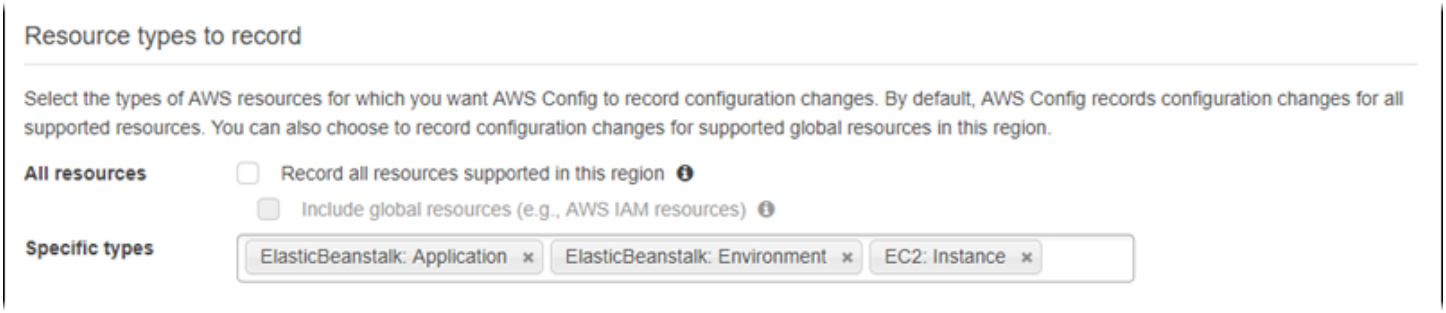
Per impostazione predefinita, AWS Config registra le modifiche alla configurazione per tutti i tipi di risorse regionali supportate che rileva nella regione in cui è in esecuzione l'ambiente. È possibile personalizzare AWS Config in modo da registrare le modifiche solo per tipi di risorse specifici o le modifiche alle risorse globali.

Ad esempio, puoi configurare AWS Config la registrazione delle modifiche per le risorse Elastic Beanstalk e un sottoinsieme di altre AWS risorse che Elastic Beanstalk avvia automaticamente. Utilizzando la AWS Config console, puoi selezionare Elastic Beanstalk come risorsa AWS Config nella pagina Impostazioni dal campo Tipi specifici. Da qui puoi scegliere di registrare qualsiasi tipo di risorsa Elastic Beanstalk: ApplicationVersionApplicazione e Ambiente.

La figura seguente mostra la pagina AWS Config Impostazioni, con i tipi di risorse Elastic Beanstalk che è possibile scegliere di registrare: ApplicationVersionApplicazione e Ambiente.



Dopo aver selezionato alcuni tipi di risorse, l'elenco Specific types (Tipi specifici) ha l'aspetto seguente.



Per ulteriori informazioni sulle risorse regionali e globali e per la procedura di personalizzazione completa, vedere Selezione dei record di risorse AWS Config.

Visualizzazione dei dettagli di configurazione di Elastic Beanstalk in AWS Config console

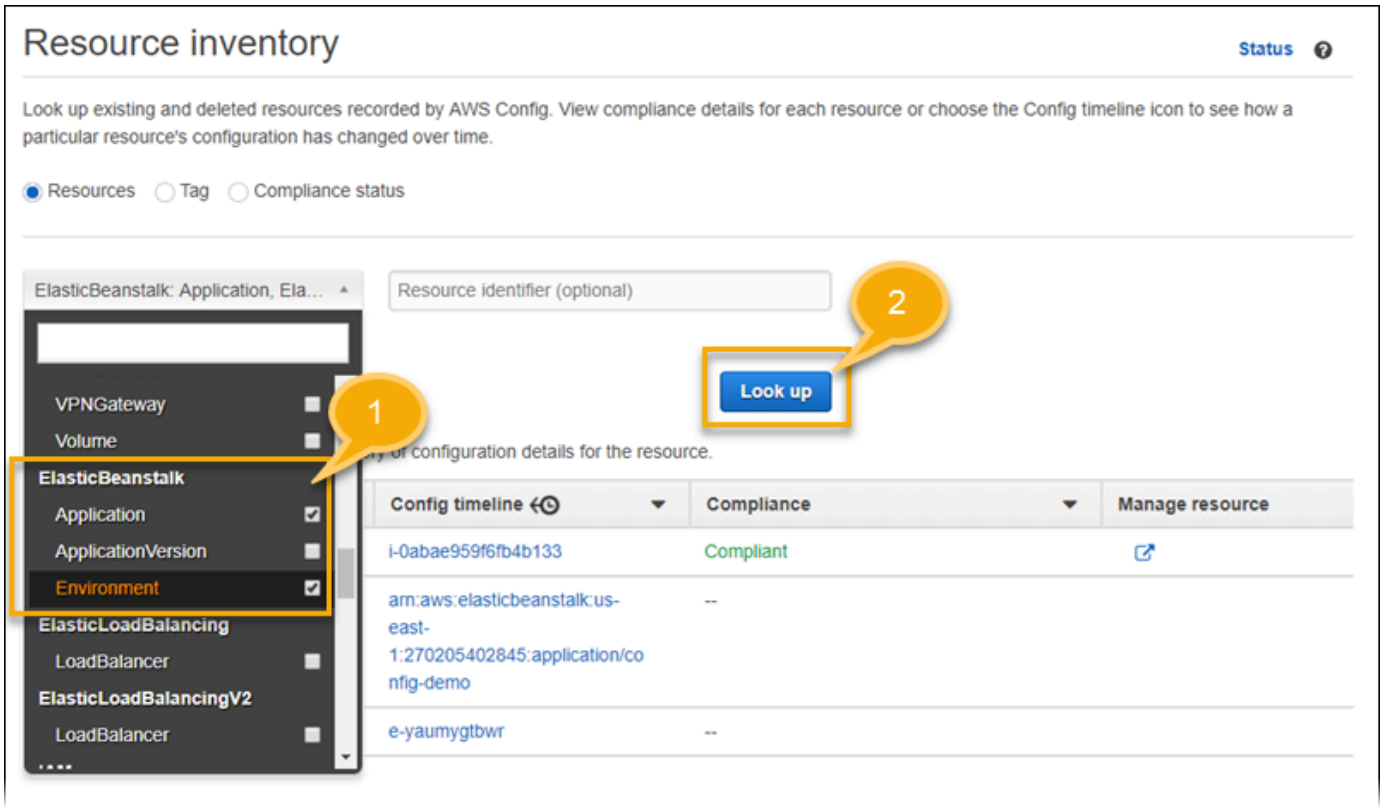
Puoi usare la AWS Config console per cercare risorse Elastic Beanstalk e ottenere dettagli attuali e storici sulle relative configurazioni. L'esempio seguente mostra come trovare informazioni su un ambiente Elastic Beanstalk.

Per trovare un ambiente Elastic Beanstalk nel AWS Config console

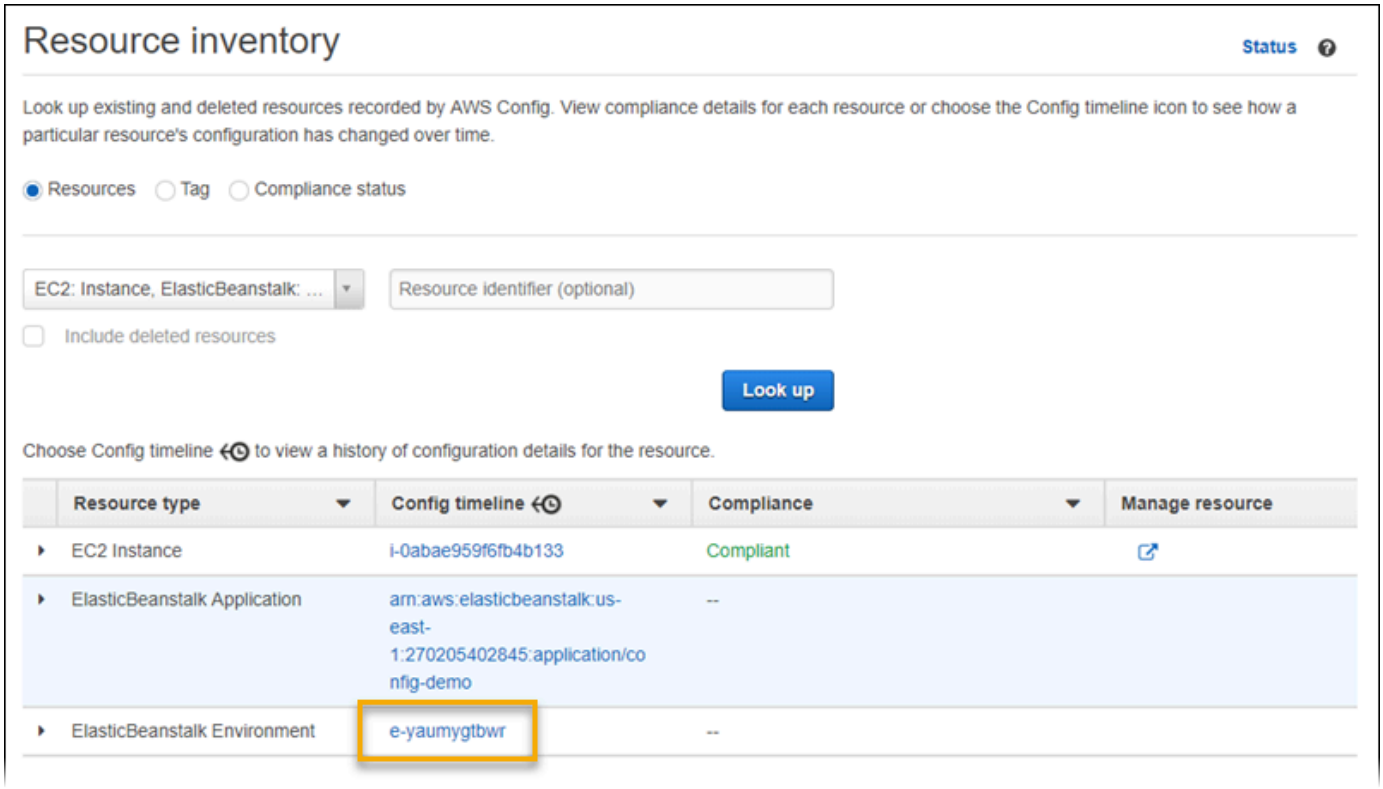
1. Aprire la console AWS Config.
2. Scegliere Resources (Risorse).
3. Nella pagina Resource inventory (Inventario risorse), scegliere Resources (Risorse).
4. Apri il menu Tipo di risorsa, scorri fino a ElasticBeanstalk, quindi scegli uno o più tipi di risorse Elastic Beanstalk.

> **Note**
>
> Per visualizzare i dettagli di configurazione per altre risorse create da Elastic Beanstalk per la tua applicazione, scegli altri tipi di risorse. Puoi ad esempio scegliere Instance (Istanza) in EC2.

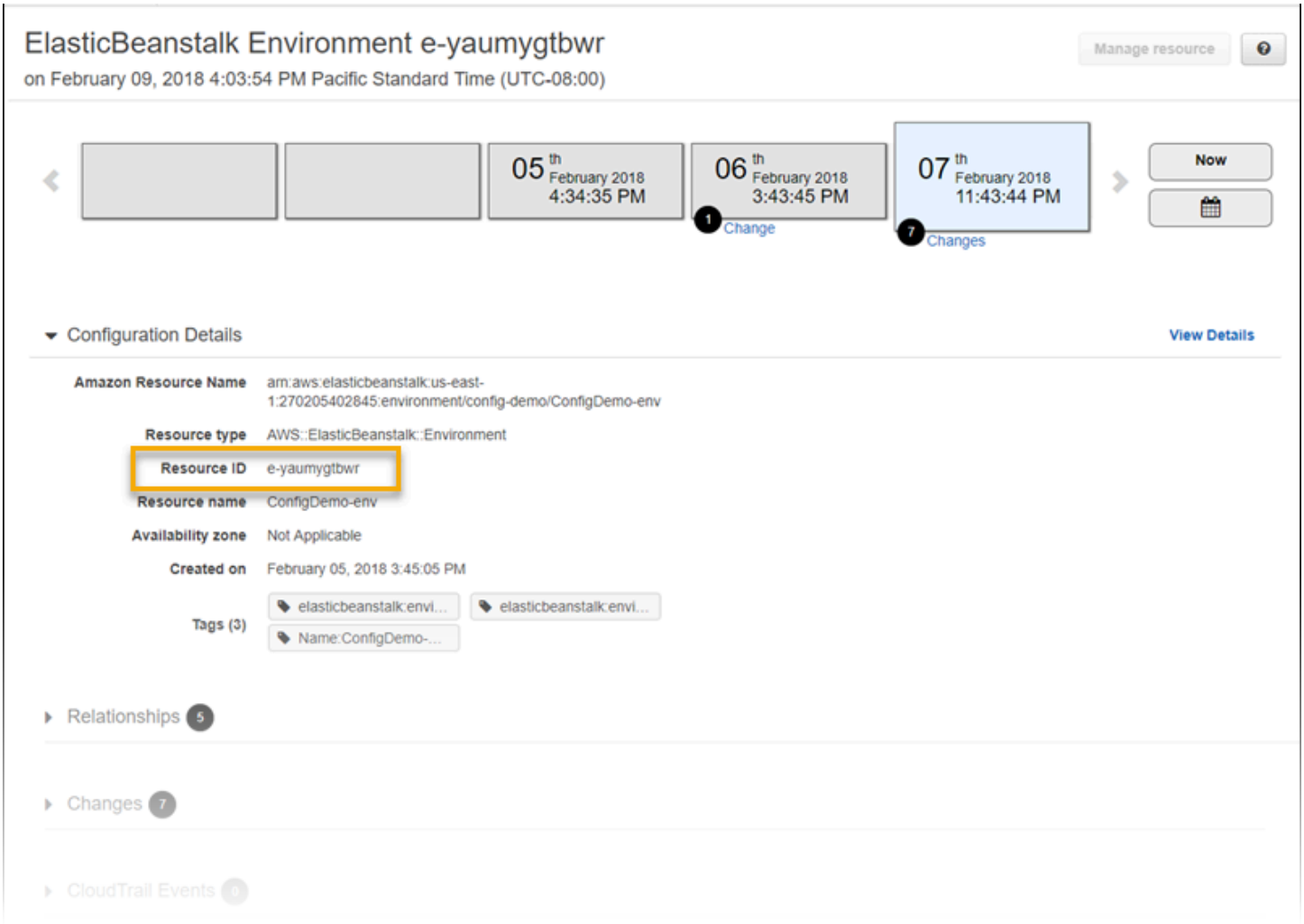
5. Scegliere Look up (Cerca). Consulta 2 nella figura seguente.



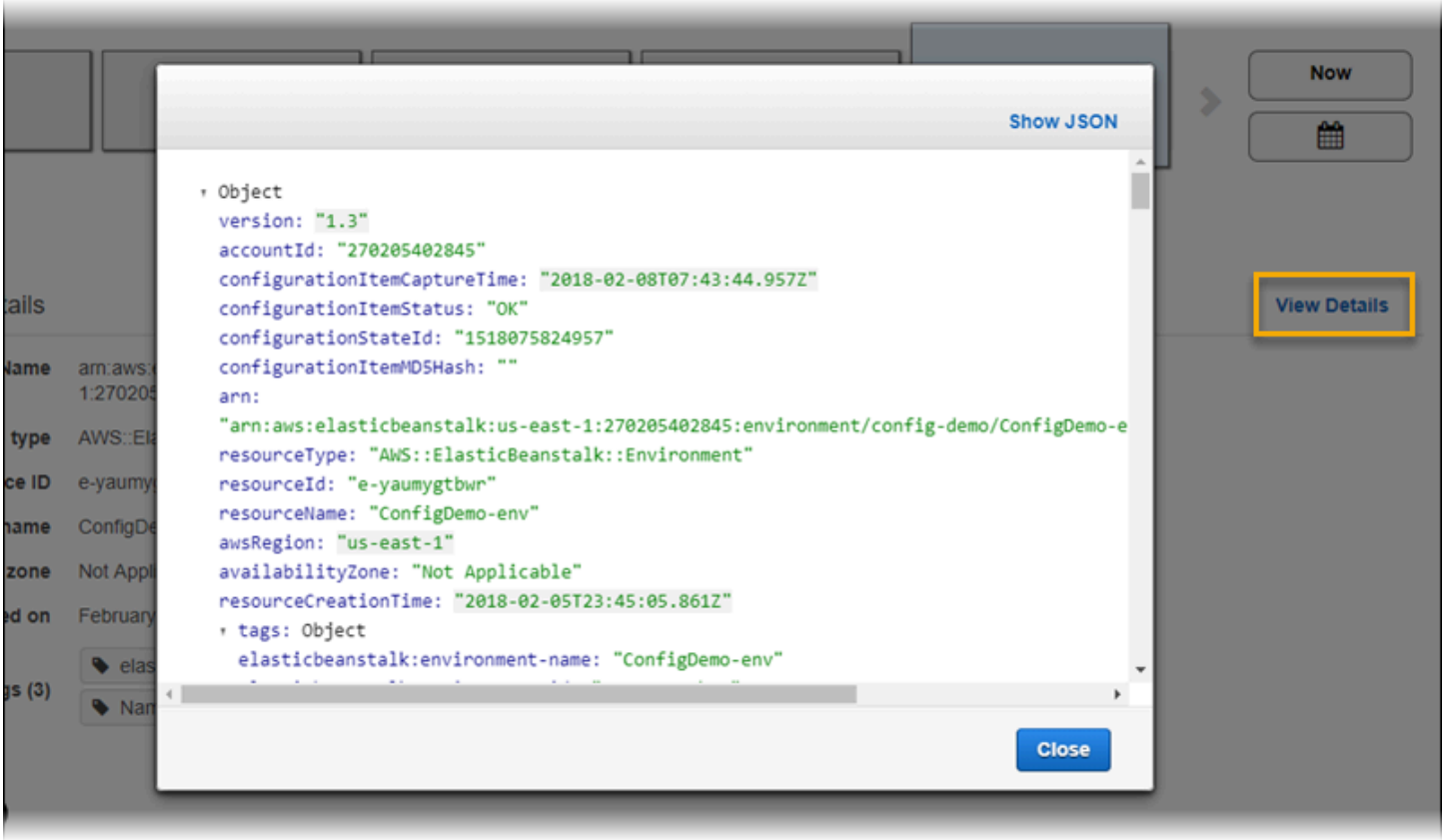
6. Scegli un ID di risorsa nell'elenco delle risorse AWS Config visualizzato.



AWS Config visualizza i dettagli di configurazione e altre informazioni sulla risorsa selezionata.



Per visualizzare i dettagli completi della configurazione registrata, scegliere View Details (Visualizza dettagli).



Per ulteriori informazioni su come trovare una risorsa e visualizzare le informazioni in questa pagina, consulta Visualizzazione AWS delle configurazioni e della cronologia delle risorse nella Guida per gli sviluppatori.AWS Config

Valutazione delle risorse Elastic Beanstalk utilizzando AWS Config rules

Puoi creare AWS Config regole che rappresentano le impostazioni di configurazione ideali per le tue risorse Elastic Beanstalk. È possibile utilizzare regole di configurazione AWS gestite predefinite o definire regole personalizzate. AWS Config monitora continuamente le modifiche alla configurazione delle risorse per determinare se tali modifiche violano alcune delle condizioni delle regole. La AWS Config console mostra lo stato di conformità delle regole e delle risorse.

Se una risorsa viola una regola e viene contrassegnata come non conforme, può AWS Config avvisarti utilizzando un argomento di Amazon Simple Notification Service (Amazon SNS). Per utilizzare in modo programmatico i dati contenuti in questi AWS Config avvisi, utilizza una coda Amazon Simple Queue Service (Amazon SQS) come endpoint di notifica per l'argomento Amazon SNS. Puoi ad esempio scrivere codice che avvia un flusso di lavoro quando qualcuno modifica la configurazione del gruppo Auto Scaling per il tuo ambiente.

Per ulteriori informazioni sulla configurazione e l'utilizzo delle regole, consulta Evaluating Resources with Rules nella Developer Guide. AWS Config AWS Config

Utilizzo di Elastic Beanstalk con Amazon DynamoDB

Amazon DynamoDB è un servizio di database NoSQL interamente gestito che combina prestazioni elevate e prevedibili con una scalabilità ottimale. Se sei uno sviluppatore, puoi utilizzare DynamoDB per creare una tabella di database in grado di archiviare e recuperare qualunque quantità di dati e di soddisfare qualsiasi livello di traffico di richieste. DynamoDB distribuisce automaticamente i dati e il traffico per la tabella su un numero di server sufficiente per gestire la capacità di richieste specificata dal cliente e la quantità di dati archiviati, garantendo al contempo prestazioni rapide e costanti. Tutti gli elementi di dati sono archiviati su unità a stato solido (SSDs) e vengono replicati automaticamente su più zone di disponibilità in una AWS regione per fornire elevata disponibilità e durabilità dei dati integrate.

Se utilizzi attività periodiche in un ambiente worker, Elastic Beanstalk crea una tabella DynamoDB e la usa per eseguire l'elezione del leader e per archiviare informazioni sull'attività. Ogni istanza dell'ambiente tenta di scrivere nella tabella ogni pochi secondi per diventare leader e di eseguire l'attività pianificata.

Puoi utilizzare i file di configurazione per creare una tabella DynamoDB per la tua applicazione. Di eb-node-express-sampleseguito è riportato un GitHub esempio di applicazione Node.js che crea una tabella con un file di configurazione e si connette ad esso tramite l' AWS SDK per Node.js JavaScript . Per una procedura guidata di esempio che utilizza DynamoDB con PHP, consulta Esempio: DynamoDB e CloudWatch SNS. Per un esempio che utilizza il AWS SDK per Java, consulta Gestire lo stato della sessione Tomcat con DynamoDB nella documentazione. AWS SDK per Java

Quando crei una tabella di DynamoDB usando i file di configurazione, la tabella non è collegata al ciclo di vita dell'ambiente e non viene eliminata quando termini l'ambiente. Per garantire che le informazioni personali non vengano conservate inutilmente, elimina i record non più necessari o elimina la tabella.

Per ulteriori informazioni su DynamoDB, consulta Guida per gli sviluppatori di DynamoDB.

Utilizzo di Elastic Beanstalk con Amazon ElastiCache

Amazon ElastiCache è un servizio web che consente di configurare, gestire e scalare ambienti di cache in memoria distribuiti nel cloud. Fornisce una cache in memoria ad alte prestazioni, scalabile

ed economica, eliminando al contempo la complessità associata all'implementazione e alla gestione di un ambiente di cache distribuito. ElastiCache è conforme al protocollo con Redis e Memcached, quindi il codice, le applicazioni e gli strumenti più diffusi che utilizzi oggi con gli ambienti Redis e Memcached esistenti funzioneranno perfettamente con il servizio. Per ulteriori informazioni ElastiCache, vai alla pagina ElastiCache del prodotto Amazon.

Per usare Elastic Beanstalk con Amazon ElastiCache

1. Crea un cluster. ElastiCache
   - Per istruzioni su come creare un ElastiCache cluster con Redis, consulta la sezione Getting Started with Amazon ElastiCache for Redis nella Guida ElastiCacheper l'utente di Redis.
   - Per istruzioni su come creare un ElastiCache cluster con Memcached, consulta la sezione Guida introduttiva ad Amazon ElastiCache for Memcached nella Guida ElastiCache per l'utente di Memcached.
2. Configura il tuo gruppo ElastiCache di sicurezza per consentire l'accesso dal gruppo EC2 di sicurezza Amazon utilizzato dalla tua applicazione Elastic Beanstalk. Per istruzioni su come trovare il nome del tuo gruppo di EC2 sicurezza utilizzando la console di AWS gestione, consulta la pagina del documento EC2 gruppi di sicurezza sulle EC2 istanze.
   - Per ulteriori informazioni su Redis, andare a Autorizza accesso nella Guida per l'utente di ElastiCache per Redis.
   - Per ulteriori informazioni su Memcached, vai su Autorizza accesso nella Guida per l'utente di ElastiCache per Memcached.

Puoi utilizzare i file di configurazione per personalizzare l'ambiente Elastic Beanstalk da utilizzare. ElastiCache Per esempi di file di configurazione che si integrano ElastiCache con Elastic Beanstalk, consulta. Esempio: ElastiCache

Utilizzo di Elastic Beanstalk con Amazon Elastic File System

Con Amazon Elastic File System (Amazon EFS) puoi creare file system di rete che possono essere montati dalle istanze in più zone di disponibilità. Un file system Amazon EFS è una AWS risorsa che utilizza gruppi di sicurezza per controllare l'accesso sulla rete che si trova nel tuo VPC predefinito o personalizzato.

In un ambiente Elastic Beanstalk è possibile utilizzare Amazon EFS per creare una directory condivisa che archivia i file caricati o modificati dagli utenti della tua applicazione. L'applicazione può

trattare un volume Amazon EFS montato come spazio di archiviazione locale. In questo modo, non è necessario modificare il codice dell'applicazione per dimensionare verso l'alto su più istanze.

Per ulteriori informazioni su Amazon EFS, consulta la Guida per l'utente di Amazon Elastic File System.

> **Note**
>
> Elastic Beanstalk crea un utente webapp che puoi configurare come proprietario per le directory delle applicazioni sulle istanze Amazon. EC2 Per ulteriori informazioni, consultare Archiviazione persistente nella sezione Considerazioni di natura progettuale di questa guida.

Sections

- File di configurazione
- File system crittografati
- Applicazioni di esempio
- Rimozione dei file system

File di configurazione

Elastic Beanstalk fornisce i file di configurazione che puoi utilizzare per creare e montare i file system EFS Amazon. Puoi creare un volume Amazon EFS come parte del tuo ambiente o montare un volume Amazon EFS creato indipendentemente da Elastic Beanstalk.

- storage-efs-createfilesystem.config: utilizza la `Resources` chiave per creare un nuovo file system e punti di montaggio in Amazon EFS. Tutte le istanze dell'ambiente possono connettersi allo stesso file system per lo storage scalabile condiviso. Utilizza `storage-efs-mountfilesystem.config` per montare il file system su ogni istanza.

> **Risorse interne**
>
> Tutte le risorse create con i file di configurazione sono associate al ciclo di vita dell'ambiente. Se si termina l'ambiente o si rimuove il file di configurazione, queste risorse andranno perse.

- storage-efs-mountfilesystem.config: monta un file system Amazon EFS su un percorso locale sulle istanze del tuo ambiente. È possibile creare il volume come parte dell'ambiente con `storage-efs-createfilesystem.config`. In alternativa, puoi montarlo nel tuo ambiente utilizzando la console Amazon EFS o AWS CLI l' AWS SDK.

Per utilizzare i file di configurazione, inizia a creare il file system Amazon EFS con `storage-efs-createfilesystem.config`. Segui le istruzioni nel file di configurazione e aggiungilo alla directory .ebextensions nel codice sorgente per creare il file system nel VPC.

Distribuire la versione aggiornata del codice sorgente nell'ambiente Elastic Beanstalk. Questo per verificare che il file system sia stato creato correttamente. Quindi, aggiungi il file `storage-efs-mountfilesystem.config` per montare il file system nelle istanze dell'ambiente. Se si esegue questa operazione in due implementazioni separate, si garantisce che, se l'operazione di montaggio non riesce, il file system viene lasciato intatto. Se si eseguono entrambe le operazioni nella stessa distribuzione, si verifica un problema con una delle due fasi che termina il file system quando la distribuzione non riesce.

File system crittografati

Amazon EFS supporta i file system crittografati. Il file di configurazione `storage-efs-createfilesystem.config` descritto in questo argomento definisce due opzioni personalizzate. È possibile utilizzare queste opzioni per creare un file system Amazon EFS crittografato. Per ulteriori informazioni, fare riferimento alle istruzioni nel file di configurazione.

Applicazioni di esempio

Elastic Beanstalk fornisce anche applicazioni di esempio che utilizzano Amazon EFS per lo storage condiviso. I due progetti dispongono di file di configurazione che è possibile utilizzare con un programma di installazione standard WordPress o Drupal per eseguire un blog o un altro sistema di gestione dei contenuti in un ambiente con carico bilanciato. Quando un utente carica una foto o altri file multimediali, questi vengono salvati in un file system Amazon EFS. Ciò evita di dover usare l'alternativa, ovvero utilizzare un plug-in per archiviare i file caricati in Amazon S3.

- Carico bilanciato WordPress: include i file di configurazione per l'installazione WordPress sicura e l'esecuzione in un ambiente Elastic Beanstalk con carico bilanciato.
- Drupal con sistema di bilanciamento del carico: include i file di configurazione e le istruzioni per l'installazione sicura di Drupal e la relativa esecuzione in un ambiente Elastic Beanstalk con sistema di bilanciamento del carico.

Rimozione dei file system

Se è stato creato un file system Amazon EFS utilizzando un file di configurazione come parte dell'ambiente Elastic Beanstalk, il file system viene rimosso da Elastic Beanstalk quando si termina l'ambiente. Per ridurre al minimo i costi di archiviazione di un'applicazione in esecuzione, eliminare regolarmente gli oggetti non più necessari per l'applicazione. In alternativa, assicurarsi che il codice dell'applicazione mantenga correttamente il ciclo di vita dei file.

> **Important**
>
> Se è stato creato un file system Amazon EFS all'esterno di un ambiente Elastic Beanstalk ed è stato montato nelle istanze dell'ambiente, Elastic Beanstalk non rimuove il file system quando si termina l'ambiente. Per evitare che le informazioni personali vengano conservate inutilmente, eliminare i file archiviati dall'applicazione se non servono più. In alternativa, è possibile rimuovere l'intero file system.

Utilizzo di Elastic Beanstalk con AWS Identity and Access Management

AWS Identity and Access Management (IAM) ti aiuta a controllare in modo sicuro l'accesso alle tue risorse. AWS Questa sezione include materiali di riferimento per l'utilizzo di ruoli del servizio, profili di istanza e policy IAM.

Per una panoramica delle autorizzazioni, consulta Ruoli, profili di istanza e policy utente di Elastic Beanstalk Service. Per la maggior parte degli ambienti, il ruolo di servizio e il profilo di istanza che la console Elastic Beanstalk chiede di creare all'avvio del primo ambiente dispongono di tutte le autorizzazioni necessarie. Analogamente, le policy gestite fornite da Elastic Beanstalk per l'accesso completo e l'accesso in sola lettura contengono tutte le autorizzazioni utente necessarie per l'uso quotidiano.

La IAM User Guide fornisce una copertura approfondita delle AWS autorizzazioni.

Argomenti

- Gestione dei profili dell'istanza Elastic Beanstalk
- Gestione dei ruoli del servizio Elastic Beanstalk
- Utilizzo dei ruoli collegati ai servizi per Elastic Beanstalk

- Gestione delle policy utente Elastic Beanstalk
- Formato Amazon Resource Name per Elastic Beanstalk
- Risorse e condizioni per le operazioni Elastic Beanstalk
- Utilizzo dei tag per controllare l'accesso alle risorse Elastic Beanstalk
- Policy di esempio basate su policy gestite
- Policy di esempio basate su autorizzazioni a livello di risorsa
- Impedire l'accesso ai bucket Amazon S3 tra ambienti

Gestione dei profili dell'istanza Elastic Beanstalk

Un profilo di istanza è un contenitore per un ruolo AWS Identity and Access Management (IAM) che puoi utilizzare per passare informazioni sul ruolo a un'istanza Amazon EC2 all'avvio dell'istanza.

Se il tuo AWS account non dispone di un profilo di istanza EC2, devi crearne uno utilizzando il servizio IAM. Potrai quindi assegnare il profilo dell'istanza EC2 ai nuovi ambienti che vengono creati. I passaggi di creazione dell'ambiente nella console Elastic Beanstalk forniscono l'accesso alla console IAM, in modo da poter creare un profilo di istanza EC2 con le autorizzazioni richieste.

> **Note**
>
> In precedenza Elastic Beanstalk creava un `aws-elasticbeanstalk-ec2-role` profilo di istanza EC2 predefinito denominato la prima volta AWS che un account creava un ambiente. Questo profilo dell'istanza includeva le policy gestite predefinite. Se il tuo account dispone già di questo profilo dell'istanza, rimarrà disponibile per poter essere assegnata ai tuoi ambienti. Tuttavia, le recenti linee guida AWS sulla sicurezza non consentono a un AWS servizio di creare automaticamente ruoli con policy di fiducia per altri AWS servizi, in questo caso EC2. A causa di queste linee guida sulla sicurezza, Elastic Beanstalk non crea più un profilo dell'istanza `aws-elasticbeanstalk-ec2-role` predefinito.

Policy gestite

Elastic Beanstalk fornisce diverse policy gestite per consentire all'ambiente di soddisfare diversi casi d'uso. Per soddisfare i casi d'uso predefiniti per un ambiente, queste policy devono essere associate al ruolo per il profilo dell'istanza EC2.

- AWSElasticBeanstalkWebTier— Concede all'applicazione le autorizzazioni per caricare i log su Amazon S3 e le informazioni di debug su. AWS X-RayPer visualizzare il contenuto della policy gestita, consulta la Managed Policy Reference Guide AWSElasticBeanstalkWebTier.AWS
- AWSElasticBeanstalkWorkerTier— Concede le autorizzazioni per il caricamento dei log, il debug, la pubblicazione delle metriche e le attività relative alle istanze di lavoro, tra cui la gestione delle code, l'elezione dei leader e le attività periodiche. Per visualizzare il contenuto delle policy gestite, consulta la Managed Policy Reference Guide. AWSElasticBeanstalkWorkerTierAWS
- AWSElasticBeanstalkMulticontainerDocker— Concede le autorizzazioni ad Amazon Elastic Container Service per coordinare le attività del cluster per gli ambienti Docker. Per visualizzare il contenuto delle policy gestite, consulta la AWS Managed Policy AWSElasticBeanstalkMulticontainerDockerReference Guide.

> **Important**
>
> Le policy gestite da Elastic Beanstalk non forniscono autorizzazioni granulari, ma concedono tutte le autorizzazioni potenzialmente necessarie per lavorare con le applicazioni Elastic Beanstalk. In alcuni casi potresti voler limitare ulteriormente le autorizzazioni delle nostre politiche gestite. Per un esempio di un caso d'uso, vediImpedire l'accesso ai bucket Amazon S3 tra ambienti.
> Inoltre, le nostre policy gestite non coprono le autorizzazioni per le risorse personalizzate che potresti aggiungere alla soluzione e che non sono gestite da Elastic Beanstalk. Per implementare autorizzazioni più granulari, autorizzazioni minime richieste o autorizzazioni a livello di risorsa personalizzate, utilizza le policy personalizzate.

Policy sulle relazioni di attendibilità per EC2

Per permettere alle istanze EC2 nell'ambiente di assumere il ruolo richiesto, il profilo dell'istanza deve specificare Amazon EC2 come entità attendibile nella policy di relazione di attendibilità, come riportato di seguito.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Per personalizzare le autorizzazioni, puoi aggiungere policy al ruolo collegato al profilo dell'istanza predefinito oppure creare un profilo dell'istanza personalizzato con un gruppo limitato di autorizzazioni.

Per utilizzare la funzionalità di analisi AI-powered dell'ambiente, aggiungi`bedrock:InvokeModel`, `bedrock:ListFoundationModelselasticbeanstalk:DescribeEvents`, e `elasticbeanstalk:DescribeEnvironmentHealth` le autorizzazioni al profilo dell'istanza. Queste autorizzazioni consentono a Elastic Beanstalk di utilizzare Amazon Bedrock e accedere ai dati dell'ambiente per analizzare log, eventi e stato delle istanze. Nelle aree commerciali, se è la prima volta che viene utilizzato un modello Anthropic Claude nel tuo AWS account, aggiungi anche le autorizzazioni, e. `aws-marketplace:Subscribe` `aws-marketplace:Unsubscribe` `aws-marketplace:ViewSubscriptions` Queste autorizzazioni AWS Marketplace abilitano l'abbonamento automatico al modello richiesto da Amazon Bedrock per il primo accesso a modelli di terze parti. Nelle GovCloud regioni, l'analisi AI utilizza il modello NVIDIA Nemotron e non richiede le autorizzazioni Marketplace AWS .

Sections

- Creazione di un profilo dell'istanza
- Aggiunta delle autorizzazioni al profilo dell'istanza predefinito
- Verifica delle autorizzazioni assegnate al profilo dell'istanza
- Aggiornamento di un profilo dell'istanza predefinito obsoleto

Creazione di un profilo dell'istanza

Un profilo dell'istanza è un wrapper intorno a un ruolo IAM standard che permette a un'istanza EC2 di assumere il ruolo. Puoi creare un profilo di istanza con le policy gestite di Elastic Beanstalk predefinite. Puoi anche creare profili di istanza aggiuntivi per personalizzare le autorizzazioni per

diverse applicazioni. Oppure puoi creare un profilo di istanza che non includa le due policy gestite che concedono le autorizzazioni per gli ambienti Docker gestiti da ECS o Worker, se non utilizzi queste funzionalità.

Per creare un profilo di istanza con le politiche gestite predefinite

1. Apri la pagina Roles (Ruoli) nella console IAM.
2. Scegli Crea ruolo.
3. Per Tipo di entità attendibile, scegli Servizio AWS .
4. Per Service o use case, scegli Elastic Beanstalk.
5. Per il caso d'uso, scegli Elastic Beanstalk — Compute.
6. Scegli Next (Successivo).
7. Immettere un nome di ruolo in Role name (Nome ruolo).

   Puoi inserire il nome del ruolo predefinito suggerito dalla console Elastic Beanstalk:. `aws-elasticbeanstalk-ec2-role`
8. Verifica che le politiche di autorizzazione includano quanto segue, quindi scegli Avanti:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
9. Scegli Crea ruolo.

Per creare un profilo di istanza con la tua scelta specifica di politiche gestite

1. Apri la pagina Roles (Ruoli) nella console IAM.
2. Scegli Crea ruolo.
3. In Tipo di entità attendibile, scegli Servizio AWS .
4. Per Caso d’uso, seleziona EC2.
5. Scegli Next (Successivo).
6. Collega le policy gestite appropriate di Elastic Beanstalk e tutte le altre policy che forniscono le autorizzazioni necessarie per la tua applicazione.
7. Scegli Next (Successivo).
8. Inserisci un nome per il ruolo.

9. (Facoltativo) Aggiungi i tag al ruolo.
10. Scegli Crea ruolo.

Aggiunta delle autorizzazioni al profilo dell'istanza predefinito

Se la tua applicazione accede ad AWS API o risorse per le quali le autorizzazioni non sono concesse nel profilo di istanza predefinito, aggiungi delle policy che concedono le autorizzazioni nella console IAM.

Per aggiungere policy al ruolo collegato al profilo dell'istanza predefinito

1. Aprire la pagina Roles (Ruoli) nella console IAM.
2. Scegli il ruolo assegnato come profilo dell'istanza EC2.
3. Nella scheda Autorizzazioni, scegli Collega policy.
4. Seleziona la policy gestita per i servizi aggiuntivi utilizzati dall'applicazione. Ad esempio `AmazonS3FullAccess` o `AmazonDynamoDBFullAccess`.
5. Scegli Collega policy.

Verifica delle autorizzazioni assegnate al profilo dell'istanza

Le autorizzazioni assegnate al profilo dell'istanza predefinito possono variare in base al momento in cui è stato creato, all'ultima volta in cui hai avviato un ambiente e al client che hai utilizzato. Puoi verificare le autorizzazioni per il profilo dell'istanza predefinito nella console IAM.

Per verificare le autorizzazioni del profilo dell'istanza predefinito

1. Apri la pagina Roles (Ruoli) nella console IAM.
2. Scegli il ruolo assegnato come profilo dell'istanza EC2.
3. Nella scheda Permissions (Autorizzazioni), esamina l'elenco delle policy collegate al ruolo.
4. Per visualizzare le autorizzazioni concesse da una policy, scegli la policy.

Aggiornamento di un profilo dell'istanza predefinito obsoleto

Se il profilo dell'istanza predefinito non dispone delle autorizzazioni necessarie, puoi aggiungere manualmente le policy gestite al ruolo assegnato come profilo dell'istanza EC2.

Per aggiungere policy gestite al ruolo collegato al profilo dell'istanza predefinito

1. Apri la pagina Roles (Ruoli) nella console IAM.
2. Scegli il ruolo assegnato come profilo dell'istanza EC2.
3. Nella scheda Autorizzazioni, scegli Collega policy.
4. Digita **`AWSElasticBeanstalk`** per filtrare le policy.
5. Seleziona le policy seguenti e scegli Collega policy:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`

Gestione dei ruoli del servizio Elastic Beanstalk

Per gestire e monitorare l'ambiente, AWS Elastic Beanstalk esegue azioni sulle risorse ambientali per conto dell'utente. Elastic Beanstalk necessita di determinate autorizzazioni per eseguire queste azioni e AWS Identity and Access Management presuppone ruoli di servizio (IAM) per ottenere queste autorizzazioni.

Elastic Beanstalk deve utilizzare delle credenziali di sicurezza provvisorie ogni volta che assume un ruolo di servizio. Per ottenere queste credenziali, Elastic Beanstalk invia una richiesta a AWS Security Token Service (AWS STS) su un endpoint specifico della regione. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee nella Guida per l'utente di IAM.

> **Note**
>
> Se l' AWS STS endpoint per la regione in cui si trova l'ambiente è disattivato, Elastic Beanstalk invia la richiesta su un endpoint alternativo che non può essere disattivato. Questo endpoint è associato a una regione diversa. Pertanto, la richiesta è una richiesta tra regioni. Per ulteriori informazioni, consulta Attivazione e disattivazione AWS STS in una regione nella Guida per l'utente IAM. AWS

Gestione dei ruoli del servizio utilizzando la console Elastic Beanstalk e CLI EB

La console Elastic Beanstalk e la CLI EB possono essere utilizzate per impostare i ruoli del servizio per l'ambiente con un set di autorizzazioni sufficiente. Creano un ruolo di servizio predefinito e utilizzano policy gestite.

Policy gestite dei ruoli dei servizi

Elastic Beanstalk fornisce una policy gestita per il monitoraggio avanzato dello stato e una con le autorizzazioni aggiuntive necessarie per gli aggiornamenti gestiti della piattaforma. La console e la CLI EB assegnano entrambi queste policy al ruolo di servizio predefinito creato per l'utente. Queste policy devono essere utilizzate solo per questo ruolo di servizio predefinito. Non devono essere utilizzate con altri utenti o ruoli nei tuoi account.

`AWSElasticBeanstalkEnhancedHealth`

Questa policy concede le autorizzazioni a Elastic Beanstalk per monitorare l'integrità dell'istanza e dell'ambiente. Inoltre include anche operazioni Amazon SQS per consentire a Elastic Beanstalk di monitorare le attività di coda per gli ambienti worker. Per visualizzare il contenuto di questa policy gestita, consulta la AWSElasticBeanstalkEnhancedHealthpagina della Managed Policy Reference AWS Guide.

`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Questa policy concede le autorizzazioni a Elastic Beanstalk per aggiornare gli ambienti per tuo conto ed eseguire gli aggiornamenti gestiti della piattaforma. Per visualizzare il contenuto di questa policy gestita, consulta la AWSElasticBeanstalkManagedUpdatesCustomerRolePolicypagina della AWS Managed Policy Reference Guide.

Raggruppamenti di autorizzazioni a livello di servizio

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- *`ElasticBeanstalkPermissions`*— Questo gruppo di autorizzazioni serve a richiamare le azioni del servizio Elastic Beanstalk (Elastic Beanstalk). APIs
- *`AllowPassRoleToElasticBeanstalkAndDownstreamServices`*: questo gruppo di autorizzazioni consente di passare qualsiasi ruolo a Elastic Beanstalk e ad altri servizi downstream come CloudFormation.
- *`ReadOnlyPermissions`*: questo gruppo di autorizzazioni serve a raccogliere informazioni sull'ambiente in esecuzione.

- `*OperationPermissions`: i gruppi con questo modello di denominazione servono a chiamare le operazioni necessarie per eseguire gli aggiornamenti della piattaforma.
- `*BroadOperationPermissions`: i gruppi con questo modello di denominazione servono a chiamare le operazioni necessarie per eseguire gli aggiornamenti della piattaforma. Includono inoltre autorizzazioni estese per il supporto degli ambienti legacy.
- `*TagResource`— I gruppi con questo modello di denominazione sono destinati alle chiamate che utilizzano per tag-on-create APIs allegare tag alle risorse che vengono create in un ambiente Elastic Beanstalk.

Per visualizzare il contenuto di una policy gestita, puoi utilizzare anche la pagina Policy nella console IAM.

> **Important**
>
> Le policy gestite da Elastic Beanstalk non forniscono autorizzazioni granulari, ma concedono tutte le autorizzazioni potenzialmente necessarie per lavorare con le applicazioni Elastic Beanstalk. In alcuni casi potresti voler limitare ulteriormente le autorizzazioni delle nostre politiche gestite. Per un esempio di un caso d'uso, vediImpedire l'accesso ai bucket Amazon S3 tra ambienti.
>
> Inoltre, le nostre policy gestite non coprono le autorizzazioni per le risorse personalizzate che potresti aggiungere alla soluzione e che non sono gestite da Elastic Beanstalk. Per implementare autorizzazioni più granulari, autorizzazioni minime richieste o autorizzazioni a livello di risorsa personalizzate, utilizza le policy personalizzate.

Policy gestite da obsolete

In passato, Elastic Beanstalk AWSElasticBeanstalkServicesupportava la politica dei ruoli dei servizi gestiti. Questa policy è stata sostituita da AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy. Potresti comunque essere in grado di visualizzare e utilizzare le policy precedenti nella console IAM.

Per visualizzare il contenuto della policy gestita, consulta la AWS Managed Policy AWSElasticBeanstalkServiceReference Guide.

Tuttavia, ti consigliamo di passare all'utilizzo della nuova policy gestita (AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy). Aggiungi policy personalizzate per concedere autorizzazioni alle risorse personalizzate, se presenti.

Utilizzo della console Elastic Beanstalk

Quando avvii un ambiente nella console Elastic Beanstalk, la console crea un ruolo di servizio predefinito denominato `aws-elasticbeanstalk-service-role` a cui collega le policy gestite con le autorizzazioni predefinite.

Per consentire a Elastic Beanstalk di assumere il ruolo `aws-elasticbeanstalk-service-role`, il ruolo di servizio specifica Elastic Beanstalk come entità attendibile nella policy delle relazioni di trust.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
          "Service": "elasticbeanstalk.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "sts:ExternalId": "elasticbeanstalk"
          }
        }
      }
    ]
}
```

Quando abiliti gli aggiornamenti gestiti della piattaforma per l'ambiente, Elastic Beanstalk assume un ruolo di servizio degli aggiornamenti gestiti separato per eseguire gli aggiornamenti gestiti. Per impostazione predefinita, la console Elastic Beanstalk utilizza lo stesso ruolo di servizio generato, `aws-elasticbeanstalk-service-role`, per il ruolo di servizio degli aggiornamenti gestiti. Se modifichi il ruolo di servizio predefinito, la console imposta il ruolo del servizio degli aggiornamenti gestiti in modo da utilizzare il ruolo collegato al servizio degli aggiornamenti gestiti, `AWSServiceRoleForElasticBeanstalkManagedUpdates`. Per ulteriori informazioni sui ruoli collegati al servizio, consulta the section called “Uso di ruoli collegati ai servizi”.

> **Note**
>
> A causa di problemi di autorizzazione, il servizio Elastic Beanstalk non sempre riesce a creare questo ruolo collegato ai servizi per tuo conto. Pertanto, la console tenta di crearlo in modo esplicito. Per garantire che l'account disponga di questo ruolo collegato ai servizi, crea un ambiente almeno una volta utilizzando la console e configura gli aggiornamenti gestiti da abilitare prima di creare l'ambiente.

Utilizzo della CLI EB

Quando avvii un ambiente utilizzando il comando the section called “eb create” dell'interfaccia a riga di comando di Elastic Beanstalk (CLI EB) e non specifichi un ruolo di servizio tramite l'opzione `--service-role`, Elastic Beanstalk crea il ruolo di servizio predefinito `aws-elasticbeanstalk-service-role`. Se il ruolo di servizio predefinito esiste già, Elastic Beanstalk lo utilizza per il nuovo ambiente. Anche la console Elastic Beanstalk, in queste situazioni, esegue azioni simili.

Al contrario della console, non puoi specificare un ruolo di servizio degli aggiornamenti gestiti quando si utilizza un'opzione di comando della CLI EB. Se attivi gli aggiornamenti gestiti per l'ambiente, imposta il ruolo di servizio degli aggiornamenti gestiti tramite le opzioni di configurazione. Nell'esempio seguente vengono attivati gli aggiornamenti gestiti e viene utilizzato il ruolo del servizio predefinito come ruolo del servizio degli aggiornamenti gestiti.

Example.ebextensions/ managed-platform-update .config

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
    ServiceRoleForManagedUpdates: "aws-elasticbeanstalk-service-role"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Gestione dei ruoli del servizio utilizzando l'API Elastic Beanstalk

Quando utilizzi l'operazione `CreateEnvironment` dell'API Elastic Beanstalk per creare un ambiente, specifica un ruolo di servizio utilizzando l'opzione di configurazione `ServiceRole` nello spazio dei nomi `aws:elasticbeanstalk:environment`. Per ulteriori informazioni sull'utilizzo del

monitoraggio avanzato dello stato con l'API Elastic Beanstalk, consulta Utilizzo di report dello stato avanzato con l'API Elastic Beanstalk.

Inoltre, se abiliti gli aggiornamenti gestiti della piattaforma per l'ambiente, puoi specificare un ruolo del servizio degli aggiornamenti gestiti utilizzando l'opzione `ServiceRoleForManagedUpdates` dello spazio dei nomi `aws:elasticbeanstalk:managedactions`.

Uso di ruoli collegati ai servizi

Un ruolo collegato al servizio è un tipo unico di ruolo di servizio predefinito da Elastic Beanstalk per includere tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS Il ruolo collegato al servizio è associato al tuo account. Elastic Beanstalk lo crea una volta, quindi lo riutilizza quando crea altri ambienti. Per informazioni dettagliate sull'utilizzo di ruoli collegati ai servizi con ambienti Elastic Beanstalk, consulta Utilizzo dei ruoli collegati ai servizi per Elastic Beanstalk.

Quando crei un ambiente utilizzando l'API Elastic Beanstalk senza specificare un ruolo di servizio, Elastic Beanstalk crea un ruolo collegato ai servizi di monitoraggio per il tuo account, se non esiste già. Il ruolo viene quindi utilizzato per il nuovo ambiente. Puoi anche utilizzare IAM per creare in anticipo il ruolo collegato ai servizi di monitoraggio del tuo account. Una volta che l'account ha questo ruolo, potrai utilizzarlo per creare un ambiente utilizzando l'API Elastic Beanstalk, la console Elastic Beanstalk o CLI EB.

Se si abilitano gli aggiornamenti della piattaforma gestita per l'ambiente e si specifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` come valore per l'opzione `ServiceRoleForManagedUpdates` dello spazio dei nomi `aws:elasticbeanstalk:managedactions`, Elastic Beanstalk crea un ruolo collegato ai servizi con aggiornamenti gestiti per l'account, se non esiste già. Elastic Beanstalk utilizza il ruolo per eseguire aggiornamenti gestiti per il nuovo ambiente.

> **Note**
>
> Quando Elastic Beanstalk cerca di creare i ruoli collegati ai servizi di monitoraggio e di aggiornamenti gestiti per il tuo account al momento della creazione di un ambiente, devi disporre dell'autorizzazione `iam:CreateServiceLinkedRole`. Se non disponi di questa autorizzazione, la creazione dell'ambiente non va a buon fine e viene visualizzato un messaggio che descrive il problema.

In alternativa, un altro utente con l'autorizzazione per creare ruoli collegati ai servizi può utilizzare IAM per creare in anticipo il ruolo collegato ai servizi. Utilizzando questo metodo, non è necessaria l'autorizzazione `iam:CreateServiceLinkedRole` per creare l'ambiente.

Verifica delle autorizzazioni del ruolo di servizio predefinito

Le autorizzazioni concesse dal ruolo di servizio predefinito possono variare in base al momento in cui è stato creato, all'ultima volta in cui hai avviato un ambiente e al client che hai utilizzato. Puoi verificare le autorizzazioni concesse dal ruolo predefinito del servizio nella console IAM.

Per verificare le autorizzazioni del ruolo di servizio predefinito

1. Apri la pagina Ruoli nella console IAM.
2. Scegli aws-elasticbeanstalk-service-role.
3. Nella scheda Permissions (Autorizzazioni), esamina l'elenco delle policy collegate al ruolo.
4. Per vedere le autorizzazioni concesse da una policy, scegli la policy.

Aggiornamento di un ruolo di servizio predefinito out-of-date

Se il ruolo predefinito del servizio non dispone delle autorizzazioni necessarie, puoi aggiornarlo mediante la creazione di un nuovo ambiente nella console di gestione dell'ambiente Elastic Beanstalk.

In alternativa, puoi aggiungere manualmente le policy gestite al ruolo di servizio predefinito.

Per aggiungere policy gestite al ruolo di servizio predefinito

1. Apri la pagina Ruoli nella console IAM.
2. Scegli aws-elasticbeanstalk-service-role.
3. Nella scheda Autorizzazioni, scegli Collega policy.
4. Digita **`AWSElasticBeanstalk`** per filtrare le policy.
5. Seleziona le policy seguenti e scegli Collega policy:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Aggiunta di autorizzazioni al ruolo di servizio predefinito

Se l'applicazione include file di configurazione che fanno riferimento a AWS risorse per le quali le autorizzazioni non sono incluse nel ruolo di servizio predefinito, Elastic Beanstalk potrebbe aver bisogno di autorizzazioni aggiuntive. Queste autorizzazioni aggiuntive sono necessarie per risolvere questi riferimenti quando elabora i file di configurazione durante un aggiornamento gestito. Se le autorizzazioni mancano, l'aggiornamento non riesce ed Elastic Beanstalk restituisce un messaggio che indica l'autorizzazione richiesta. Segui le istruzioni riportate per aggiungere le autorizzazioni per i servizi aggiuntivi al ruolo di servizio predefinito nella console IAM.

Per aggiungere policy aggiuntive al ruolo di servizio predefinito

1. Apri la pagina Ruoli nella console IAM.
2. Scegli aws-elasticbeanstalk-service-role.
3. Nella scheda Autorizzazioni, scegli Collega policy.
4. Seleziona la policy gestita per i servizi aggiuntivi utilizzati dall'applicazione. Ad esempio `AmazonAPIGatewayAdministrator` o `AmazonElasticFileSystemFullAccess`.
5. Scegli Collega policy.

Creazione di un ruolo del servizio

Se non riesci a utilizzare il ruolo di servizio predefinito, creane uno.

Per creare un ruolo di servizio

1. Apri la pagina Ruoli nella console IAM.
2. Scegli Crea ruolo.
3. In AWS service (Servizio AWS), scegli AWS Elastic Beanstalk e seleziona il caso d'uso.
4. Scegli Successivo: autorizzazioni.
5. Collega le policy gestite `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` e `AWSElasticBeanstalkEnhancedHealth` ed eventuali policy aggiuntive che forniscono le autorizzazioni necessarie per l'applicazione.
6. Scegli Successivo: Tag.
7. (Facoltativo) Aggiungi i tag al ruolo.
8. Scegli Prossimo: Rivedi.
9. Inserisci un nome per il ruolo.

10. Scegli Crea ruolo.

Puoi applicare il ruolo di servizio personalizzato quando crei un ambiente nella procedura guidata per la creazione di un ambiente o con l'opzione `--service-role` nel comando `eb create`.

Utilizzo dei ruoli collegati ai servizi per Elastic Beanstalk

AWS Elastic Beanstalk utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a Elastic Beanstalk. I ruoli collegati ai servizi sono predefiniti da Elastic Beanstalk e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Elastic Beanstalk definisce alcuni tipi di ruoli collegati ai servizi:

- Monitoraggio del ruolo collegato ai servizi: consente a Elastic Beanstalk di monitorare lo stato degli ambienti in esecuzione e di pubblicare le notifiche degli eventi di integrità.
- Ruolo collegato ai servizi di manutenzione: consente a Elastic Beanstalk di eseguire regolarmente le attività di manutenzione per gli ambienti in esecuzione.
- Managed-updates service-linked role (Ruolo collegato ai servizi di aggiornamenti gestiti): consente a Elastic Beanstalk di eseguire gli aggiornamenti pianificati della piattaforma per gli ambienti in esecuzione.

Argomenti

- Ruolo collegato ai servizi di monitoraggio
- Ruolo collegato ai servizi di manutenzione
- Ruolo collegato ai servizi di aggiornamenti gestiti.

Ruolo collegato ai servizi di monitoraggio

AWS Elastic Beanstalk utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a Elastic Beanstalk. I ruoli collegati ai servizi sono predefiniti da Elastic Beanstalk e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Elastic Beanstalk perché non devi aggiungere manualmente le autorizzazioni necessarie. Elastic Beanstalk definisce le autorizzazioni dei propri ruoli associati ai servizi e, salvo diversamente definito, solo Elastic Beanstalk può

assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Elastic Beanstalk perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare Servizi AWS che funzionano con IAM e cercare i servizi che riportano Yes (Sì) nella colonna Service-Linked Role (Ruolo collegato ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni dei ruoli collegati ai servizi per Elastic Beanstalk

Elastic Beanstalk utilizza il ruolo collegato al servizio AWSServiceRoleForElasticBeanstalkdenominato — Consente a Elastic Beanstalk di monitorare lo stato degli ambienti in esecuzione e di pubblicare notifiche sugli eventi sanitari.

Il ruolo AWSService RoleForElasticBeanstalk collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `elasticbeanstalk.amazonaws.com`

La politica delle autorizzazioni del ruolo AWSService RoleForElasticBeanstalk collegato al servizio contiene tutte le autorizzazioni di cui Elastic Beanstalk ha bisogno per completare le azioni per tuo conto:

AllowCloudformationReadOperationsOnElasticBeanstalkStacks

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCloudformationReadOperationsOnElasticBeanstalkStacks",
            "Effect": "Allow",
            "Action": [
                "cloudformation:DescribeStackResource",
                "cloudformation:DescribeStackResources",
```

```
                "cloudformation:DescribeStacks"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/awseb-*",
                "arn:aws:cloudformation:*:*:stack/eb-*"
            ]
        },
        {
            "Sid": "AllowOperations",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeNotificationConfigurations",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:PutNotificationConfiguration",
                "ec2:DescribeInstanceStatus",
                "ec2:AssociateAddress",
                "ec2:DescribeAddresses",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "elasticloadbalancing:DescribeInstanceHealth",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTargetHealth",
                "elasticloadbalancing:DescribeTargetGroups",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sns:Publish"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio è necessario configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

In alternativa, puoi utilizzare una policy AWS gestita per fornire l'accesso completo a Elastic Beanstalk.

Creazione di un ruolo collegato ai servizi per Elastic Beanstalk

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un ambiente Elastic Beanstalk utilizzando l'API Elastic Beanstalk e non specifichi un ruolo di servizio, Elastic Beanstalk crea automaticamente il ruolo collegato ai servizi.

> **Important**
>
> Se utilizzavi il servizio Elastic Beanstalk prima del 27 settembre 2017, quando ha iniziato a AWSService RoleForElasticBeanstalk supportare il ruolo collegato al servizio, e il tuo account ne aveva bisogno, Elastic Beanstalk ha creato il ruolo nel tuo account. AWSService RoleForElasticBeanstalk Per ulteriori informazioni, consulta Un nuovo ruolo è apparso nel mio account IAM.

Quando Elastic Beanstalk tenta di AWSService RoleForElasticBeanstalk creare il ruolo collegato al servizio per il tuo account quando crei un ambiente, devi disporre dell'autorizzazione. `iam:CreateServiceLinkedRole` Se non disponi di questa autorizzazione, la creazione dell'ambiente non va a buon fine e viene visualizzato un messaggio che descrive il problema.

In alternativa, un altro utente che dispone dell'autorizzazione per creare ruoli collegati ai servizi può usare IAM per creare preventivamente in anticipo il ruolo collegato ai servizi. Puoi quindi creare l'ambiente anche senza disporre dell'autorizzazione `iam:CreateServiceLinkedRole`.

Puoi utilizzare la console IAM per creare un ruolo collegato ai servizi con il caso d'uso Elastic Beanstalk. Nella CLI IAM o nell'API IAM, crea un ruolo collegato ai servizi con il nome servizio `elasticbeanstalk.amazonaws.com`. Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell’account. Quando crei un ambiente Elastic Beanstalk utilizzando l'API Elastic Beanstalk e non specifichi un ruolo di servizio, Elastic Beanstalk crea di nuovo automaticamente il ruolo collegato ai servizi.

Modifica di un ruolo collegato ai servizi per Elastic Beanstalk

Elastic Beanstalk non consente di modificare il ruolo collegato al servizio. AWSService RoleForElasticBeanstalk Dopo avere creato un ruolo collegato al servizio, non sarà possibile

modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Elastic Beanstalk

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Prima di poter utilizzare IAM per eliminare un ruolo collegato ai servizi, devi innanzitutto assicurarti che tutti gli ambienti Elastic Beanstalk utilizzino un ruolo di servizio diverso o siano terminati.

> **Note**
>
> Se il servizio Elastic Beanstalk utilizza il ruolo collegato ai servizi quando tenti di terminare gli ambienti, è possibile che la terminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per terminare un ambiente Elastic Beanstalk che utilizza (console) AWSService RoleForElasticBeanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Per informazioni dettagliate sulla terminazione di un ambiente Elastic Beanstalk utilizzando CLI EB, consultare eb terminate.

TerminateEnvironmentPer ulteriori informazioni sulla chiusura di un ambiente Elastic Beanstalk tramite l'API, consulta.

Eliminazione manuale del ruolo collegato ai servizi

Utilizza la console IAM, la CLI IAM o l'API IAM per eliminare il ruolo collegato al AWSService RoleForElasticBeanstalk servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi Elastic Beanstalk

Elastic Beanstalk supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Endpoint e quote per AWS Elastic Beanstalk.

Ruolo collegato ai servizi di manutenzione

AWS Elastic Beanstalk utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a Elastic Beanstalk. I ruoli collegati ai servizi sono predefiniti da Elastic Beanstalk e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Elastic Beanstalk perché non devi aggiungere manualmente le autorizzazioni necessarie. Elastic Beanstalk definisce le autorizzazioni dei propri ruoli associati ai servizi e, salvo diversamente definito, solo Elastic Beanstalk può assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Elastic Beanstalk perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare Servizi AWS che funzionano con IAM e cercare i servizi che riportano Yes (Sì) nella colonna Service-Linked Role (Ruolo collegato ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni dei ruoli collegati ai servizi per Elastic Beanstalk

Elastic Beanstalk utilizza il ruolo collegato al servizio AWSServiceRoleForElasticBeanstalkMaintenancedenominato: consente a Elastic Beanstalk di eseguire attività di manutenzione regolari per gli ambienti in esecuzione.

Il ruolo AWSService RoleForElasticBeanstalkMaintenance collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `maintenance.elasticbeanstalk.amazonaws.com`

La politica delle autorizzazioni del ruolo AWSService RoleForElasticBeanstalkMaintenance collegato al servizio contiene tutte le autorizzazioni di cui Elastic Beanstalk ha bisogno per completare le azioni per tuo conto:

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
        {
            "Sid":
 "AllowCloudformationChangeSetOperationsOnElasticBeanstalkStacks",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateChangeSet",
                "cloudformation:DescribeChangeSet",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:ListChangeSets",
                "cloudformation:DescribeStacks"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/awseb-*",
                "arn:aws:cloudformation:*:*:stack/eb-*"
            ]
        }
}
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio è necessario configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

In alternativa, puoi utilizzare una policy AWS gestita per fornire l'accesso completo a Elastic Beanstalk.

Creazione di un ruolo collegato ai servizi per Elastic Beanstalk

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un ambiente Elastic Beanstalk utilizzando l'API Elastic Beanstalk e non specifichi un profilo dell'istanza, Elastic Beanstalk crea automaticamente il ruolo collegato ai servizi.

> **Important**
>
> Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo. Se stavi utilizzando il servizio Elastic Beanstalk prima del 18 aprile 2019, quando ha iniziato a AWSService RoleForElasticBeanstalkMaintenance supportare il ruolo collegato al servizio, e il tuo account ne aveva bisogno, Elastic Beanstalk ha creato il ruolo nel tuo account. AWSService RoleForElasticBeanstalkMaintenance Per ulteriori informazioni, consulta Un nuovo ruolo è apparso nel mio account IAM.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell’account. Quando crei un ambiente Elastic Beanstalk utilizzando l'API Elastic Beanstalk e non specifichi un profilo dell'istanza, Elastic Beanstalk crea di nuovo automaticamente il ruolo collegato ai servizi.

Modifica di un ruolo collegato ai servizi per Elastic Beanstalk

Elastic Beanstalk non consente di modificare il ruolo collegato al servizio. AWSService RoleForElasticBeanstalkMaintenance Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Elastic Beanstalk

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Per poter utilizzare IAM per eliminare un ruolo collegato ai servizi, è necessario prima terminare qualsiasi ambiente Elastic Beanstalk che usa il ruolo.

> **Note**
>
> Se il servizio Elastic Beanstalk utilizza il ruolo collegato ai servizi quando tenti di terminare gli ambienti, è possibile che la terminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per terminare un ambiente Elastic Beanstalk che utilizza (console) AWSService RoleForElasticBeanstalkMaintenance

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Per informazioni dettagliate sulla terminazione di un ambiente Elastic Beanstalk utilizzando CLI EB, consultare eb terminate.

TerminateEnvironmentPer ulteriori informazioni sulla chiusura di un ambiente Elastic Beanstalk tramite l'API, consulta.

Eliminazione manuale del ruolo collegato ai servizi

Utilizza la console IAM, la CLI IAM o l'API IAM per eliminare il ruolo collegato al AWSService RoleForElasticBeanstalkMaintenance servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi Elastic Beanstalk

Elastic Beanstalk supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Endpoint e quote per AWS Elastic Beanstalk.

Ruolo collegato ai servizi di aggiornamenti gestiti.

AWS Elastic Beanstalk utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a Elastic Beanstalk. I ruoli collegati ai servizi sono predefiniti da Elastic Beanstalk e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Elastic Beanstalk perché non devi aggiungere manualmente le autorizzazioni necessarie. Elastic Beanstalk definisce le autorizzazioni dei propri ruoli associati ai servizi e, salvo diversamente definito, solo Elastic Beanstalk può assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Elastic Beanstalk perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare Servizi AWS che funzionano con IAM e cercare i servizi che riportano Yes (Sì) nella colonna Service-Linked Role (Ruolo collegato ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni dei ruoli collegati ai servizi per Elastic Beanstalk

Elastic Beanstalk utilizza il ruolo collegato al servizio AWSServiceRoleForElasticBeanstalkManagedUpdatesdenominato: consente a Elastic Beanstalk di eseguire aggiornamenti pianificati della piattaforma degli ambienti in esecuzione.

Il ruolo AWSService RoleForElasticBeanstalkManagedUpdates collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `managedupdates.elasticbeanstalk.amazonaws.com`

La policy gestita AWSElasticBeanstalkManagedUpdatesServiceRolePolicyconcede al ruolo AWSService RoleForElasticBeanstalkManagedUpdates collegato al servizio tutte le autorizzazioni necessarie a Elastic Beanstalk per completare le azioni di aggiornamento gestito per tuo conto. Per visualizzare il contenuto della policy gestita, consulta la AWSElasticBeanstalkManagedUpdatesServiceRolePolicypagina nella Managed Policy Reference Guide.AWS

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio è necessario configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

In alternativa, puoi utilizzare una policy AWS gestita per fornire l'accesso completo a Elastic Beanstalk.

Creazione di un ruolo collegato ai servizi per Elastic Beanstalk

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un ambiente Elastic Beanstalk utilizzando l'API Elastic Beanstalk, abiliti gli aggiornamenti gestiti e specifichi `AWSServiceRoleForElasticBeanstalkManagedUpdates` come valore per l'opzione `ServiceRoleForManagedUpdates` dello spazio dei nomi `aws:elasticbeanstalk:managedactions`, Elastic Beanstalk crea automaticamente il ruolo collegato ai servizi.

Quando Elastic Beanstalk tenta di AWSService RoleForElasticBeanstalkManagedUpdates creare il ruolo collegato al servizio per il tuo account quando crei un ambiente, devi disporre dell'autorizzazione. `iam:CreateServiceLinkedRole` Se non disponi di questa autorizzazione, la creazione dell'ambiente non va a buon fine e viene visualizzato un messaggio che descrive il problema.

In alternativa, un altro utente che dispone dell'autorizzazione per creare ruoli collegati ai servizi può usare IAM per creare preventivamente in anticipo il ruolo collegato ai servizi. Puoi quindi creare l'ambiente anche senza disporre dell'autorizzazione `iam:CreateServiceLinkedRole`.

Puoi utilizzare la console IAM per creare un ruolo collegato ai servizi con il caso d'uso Aggiornamenti gestiti di Elastic Beanstalk. Nella CLI IAM o nell’API IAM, crea un ruolo collegato al servizio con il nome servizio `managedupdates.elasticbeanstalk.amazonaws.com`. Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell’account. Quando crei un ambiente Elastic Beanstalk utilizzando l'API Elastic Beanstalk, abiliti gli aggiornamenti gestiti e specifichi `AWSServiceRoleForElasticBeanstalkManagedUpdates` come valore per l'opzione `ServiceRoleForManagedUpdates` dello spazio dei nomi `aws:elasticbeanstalk:managedactions`, Elastic Beanstalk crea di nuovo automaticamente il ruolo collegato ai servizi.

Modifica di un ruolo collegato ai servizi per Elastic Beanstalk

Elastic Beanstalk non consente di modificare il ruolo collegato al servizio. AWSService RoleForElasticBeanstalkManagedUpdates Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Elastic Beanstalk

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Prima di poter utilizzare IAM per eliminare un ruolo collegato ai servizi, devi innanzitutto assicurarti che tutti gli ambienti Elastic Beanstalk con aggiornamenti gestiti abilitati utilizzino un ruolo di servizio diverso o siano terminati.

> **Note**
>
> Se il servizio Elastic Beanstalk utilizza il ruolo collegato ai servizi quando tenti di terminare gli ambienti, è possibile che la terminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l’operazione.

Per terminare un ambiente Elastic Beanstalk che utilizza (console) AWSService RoleForElasticBeanstalkManagedUpdates

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Terminate Environment (Termina ambiente).
4. Utilizza la finestra di dialogo su schermo per confermare la terminazione dell'ambiente.

Per informazioni dettagliate sulla terminazione di un ambiente Elastic Beanstalk utilizzando CLI EB, consultare eb terminate.

TerminateEnvironmentPer ulteriori informazioni sulla chiusura di un ambiente Elastic Beanstalk tramite l'API, consulta.

Eliminazione manuale del ruolo collegato ai servizi

Utilizza la console IAM, la CLI IAM o l'API IAM per eliminare il ruolo collegato al AWSService RoleForElasticBeanstalkManagedUpdates servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi Elastic Beanstalk

Elastic Beanstalk supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Endpoint e quote per AWS Elastic Beanstalk.

Gestione delle policy utente Elastic Beanstalk

AWS Elastic Beanstalk fornisce due policy gestite che consentono di assegnare l'accesso completo o l'accesso di sola lettura a tutte le risorse gestite da Elastic Beanstalk. Puoi collegare le policy a utenti o gruppi AWS Identity and Access Management (IAM) o ai ruoli assunti dagli utenti.

Policy utente gestite

- AdministratorAccess- AWSElastic Beanstalk: fornisce all'utente le autorizzazioni amministrative complete per creare, modificare ed eliminare applicazioni Elastic Beanstalk, versioni delle applicazioni, impostazioni di configurazione, ambienti e le relative risorse sottostanti. Per visualizzare il contenuto della policy gestita, consulta la pagina AdministratorAccess- AWSElastic Beanstalk nella Managed Policy Reference Guide.AWS
- AWSElasticBeanstalkReadOnly— Consente all'utente di visualizzare applicazioni e ambienti, ma non di eseguire operazioni che li modifichino. Fornisce accesso in sola lettura a tutte le risorse Elastic Beanstalk e ad altre risorse AWS recuperate dalla console Elastic Beanstalk. Tieni presente che l'accesso in sola lettura non abilita operazioni quali ad esempio il download dei log di Elastic Beanstalk per leggerli. Questo perché i log vengono caricati nel bucket Amazon S3 per cui Elastic Beanstalk necessita dell'autorizzazione in scrittura. Vedi l'esempio al termine di questo argomento per informazioni su come abilitare l'accesso ai log Elastic Beanstalk. Per visualizzare il contenuto delle policy gestite, consulta la pagina nella Managed Policy Reference Guide. AWSElasticBeanstalkReadOnlyAWS

> **Important**
>
> Le policy gestite da Elastic Beanstalk non forniscono autorizzazioni granulari, ma concedono tutte le autorizzazioni potenzialmente necessarie per lavorare con le applicazioni Elastic Beanstalk. In alcuni casi potresti voler limitare ulteriormente le autorizzazioni delle nostre politiche gestite. Per un esempio di un caso d'uso, vediImpedire l'accesso ai bucket Amazon S3 tra ambienti.
>
> Inoltre, le nostre policy gestite non coprono le autorizzazioni per le risorse personalizzate che potresti aggiungere alla soluzione e che non sono gestite da Elastic Beanstalk. Per implementare autorizzazioni più granulari, autorizzazioni minime richieste o autorizzazioni a livello di risorsa personalizzate, utilizza le policy personalizzate.

Policy gestite da obsolete

In precedenza, Elastic Beanstalk supportava altre due policy gestite per gli utenti, e. AWSElasticBeanstalkFullAccessAWSElasticBeanstalkReadOnlyAccess Abbiamo in programma di ritirare queste policy precedenti. Potresti comunque essere in grado di visualizzarle e utilizzarle nella console IAM. Tuttavia, ti consigliamo di passare all'utilizzo delle nuove policy utente gestite e di aggiungere policy personalizzati per concedere autorizzazioni alle risorse personalizzate, se presenti.

Policy per l'integrazione con altri servizi

Se si preferisce, forniamo anche policy più dettagliate che consentono di integrare il proprio ambiente con altri servizi.

- AWSElasticBeanstalkRoleCWL: consente a un ambiente di gestire i gruppi di log di Amazon CloudWatch Logs.
- AWSElasticBeanstalkRoleRDS: consente a un ambiente di integrare un'istanza Amazon RDS.
- AWSElasticBeanstalkRoleWorkerTier— Consente a un livello di ambiente di lavoro di creare una tabella Amazon DynamoDB e una coda Amazon SQS.
- AWSElasticBeanstalkRoleECS: consente a un ambiente Docker multicontainer di gestire i cluster Amazon ECS.
- AWSElasticBeanstalkRoleCore— Consente le operazioni principali di un ambiente di servizi Web.
- AWSElasticBeanstalkRoleSNS: consente a un ambiente di abilitare l'integrazione degli argomenti di Amazon SNS.

Per visualizzare il codice sorgente JSON per una politica gestita specifica, consulta la Managed Policy Reference AWS Guide.

Controllo degli accessi con policy gestite

Puoi utilizzare le policy gestite per assegnare l'accesso completo o di sola lettura a Elastic Beanstalk. Elastic Beanstalk aggiorna queste policy automaticamente quando sono richieste autorizzazioni aggiuntive per accedere alle nuove caratteristiche.

Per applicare una policy gestita a utenti o gruppi IAM

1. Apri la pagina Policies (Policy) nella console IAM.
2. Nella casella di ricerca, digita **`AWSElasticBeanstalk`** per filtrare le policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a AWSElasticBeanstalkReadOnlyo AdministratorAccess- AWSElastic Beanstalk.
4. Scegli Operazioni di policy, quindi Collega.
5. Seleziona uno o più utenti o gruppi per collegare alla policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali.
6. Scegli Collega policy.

Creazione di una policy utente personalizzata

È possibile creare policy IAM personalizzate per consentire o negare operazioni specifiche dell'API Elastic Beanstalk su determinate risorse Elastic Beanstalk e per controllare l'accesso a risorse personalizzate non gestite da Elastic Beanstalk. Per ulteriori informazioni su come allegare una policy a un utente o a un gruppo, consultare la sezione Utilizzo delle policy nella Guida per l'utente di IAM. Per maggiori dettagli sulla creazione di una policy personalizzata, consultare la sezione Creazione di policy IAM nella Guida per l'utente di IAM.

> **Note**
>
> Mentre è possibile limitare il modo in cui un utente interagisce con le API Elastic Beanstalk, al momento non c'è un modo efficace per impedire agli utenti che hanno l'autorizzazione per creare le risorse sottostanti necessarie di creare altre risorse in Amazon EC2 e in altri servizi. Queste policy possono essere considerate come un metodo efficace per distribuire le responsabilità in Elastic Beanstalk, ma non come un metodo per proteggere tutte le risorse sottostanti.

> **Important**
>
> Se hai policy personalizzate assegnate a un ruolo del servizio Elastic Beanstalk, è importante assegnargli le autorizzazioni appropriate per i modelli di avvio. Altrimenti potresti non disporre delle autorizzazioni necessarie per aggiornare un ambiente o avviarne uno nuovo. Per ulteriori informazioni, consulta Autorizzazioni richieste per i modelli di avvio.

Una policy IAM contiene istruzioni che descrivono le autorizzazioni che desideri concedere. Quando crei un'istruzione di policy per Elastic Beanstalk, è necessario comprendere come utilizzare le seguenti quattro parti di un'istruzione:

- Effetto specifica se consentire o negare le operazioni nell'istruzione.
- Operazione specifica le operazioni dell'API che desideri controllare. Ad esempio, è possibile usare `elasticbeanstalk:CreateEnvironment` per specificare l'operazione `CreateEnvironment`. Determinate operazioni, ad esempio la creazione di un ambiente, richiedono autorizzazioni aggiuntive per l'esecuzione di tali operazioni. Per ulteriori informazioni, consulta Risorse e condizioni per le operazioni Elastic Beanstalk.

  > **Note**
  >
  > Per usare l'operazione dell'API `UpdateTagsForResource`, specifica una delle seguenti due operazioni virtuali (o entrambe) anziché il nome dell'operazione dell'API:
  >
  > `elasticbeanstalk:AddTags`
  >
  > Controlla l'autorizzazione di chiamare `UpdateTagsForResource` e passare un elenco di tag da aggiungere al parametro `TagsToAdd`.
  >
  > `elasticbeanstalk:RemoveTags`
  >
  > Controlla l'autorizzazione per chiamare `UpdateTagsForResource` e passare un elenco di chiavi di tag da eliminare nel parametro `TagsToRemove`.

- Risorsa specifica le risorse di cui desideri controllare l'accesso. Per specificare le risorse Elastic Beanstalk, elenca l'Amazon Resource Name (ARN) di ciascuna risorsa.
- (facoltativo) Condizione specifica limitazioni all'autorizzazione concessa nell'istruzione. Per ulteriori informazioni, consulta Risorse e condizioni per le operazioni Elastic Beanstalk.

Le sezioni seguenti illustrano alcuni casi in cui puoi prendere in considerazione l'uso di una policy utente personalizzata.

Abilitazione della creazione di un ambiente Elastic Beanstalk con limitazioni

La policy nell'esempio seguente permette a un utente di chiamare l'operazione `CreateEnvironment` per creare un ambiente il cui nome inizia con **`Test`** con l'applicazione e la versione dell'applicazione specificate.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"CreateEnvironmentPerm",
      "Action": [
        "elasticbeanstalk:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My First
 Elastic Beanstalk Application/Test*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My First Elastic Beanstalk Application"],
          "elasticbeanstalk:FromApplicationVersion":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My First
 Elastic Beanstalk Application/First Release"]
        }
      }
    },
    {
      "Sid":"AllNonResourceCalls",
      "Action":[
        "elasticbeanstalk:CheckDNSAvailability",
        "elasticbeanstalk:CreateStorageLocation"
      ],
      "Effect":"Allow",
      "Resource":[
```

```
        "*"
      ]
    }
  ]
}
```

La policy qui sopra mostra come concedere un accesso limitato alle operazioni Elastic Beanstalk. Per avviare effettivamente un ambiente, l'utente deve disporre dell'autorizzazione a creare anche le AWS risorse che alimentano l'ambiente. Ad esempio, la policy seguente consente l'accesso a un insieme predefinito di risorse per un ambiente server Web:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "ecs:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "cloudformation:*",
        "sqs:*"
        ],
      "Resource": "*"
    }
  ]
}
```

Abilitazione dell'accesso ai log Elastic Beanstalk archiviati in Amazon S3

La policy riportata nell'esempio seguente consente a un utente di estrarre i log Elastic Beanstalk, inserirli per fasi in Amazon S3 e recuperarli.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:DeleteObject",
        "s3:GetObjectAcl",
        "s3:PutObjectAcl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::elasticbeanstalk-*"
    }
  ]
}
```

> **Note**
>
> Per limitare le autorizzazioni solo al percorso dei log, utilizza il seguente formato di risorse.
>
> ```
> "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/resources/environments/logs/*"
> ```

Abilitazione della gestione di un'applicazione Elastic Beanstalk specifica

La policy nell'esempio seguente permette a un utente di gestire gli ambienti e altre risorse all'interno di un'applicazione Elastic Beanstalk specifica. La policy impedisce le operazioni di Elastic Beanstalk sulle risorse di altre applicazioni, nonché la creazione e l'eliminazione delle applicazioni Elastic Beanstalk.

> **Note**
>
> La policy non nega l'accesso alle risorse tramite altri servizi. Rappresenta un metodo efficace per distribuire le responsabilità per la gestione di applicazioni Elastic Beanstalk tra diversi utenti, ma non un metodo per proteggere le risorse sottostanti.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:DeleteApplication"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateApplicationVersion",
        "elasticbeanstalk:CreateConfigurationTemplate",
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:DeleteApplicationVersion",
        "elasticbeanstalk:DeleteConfigurationTemplate",
        "elasticbeanstalk:DeleteEnvironmentConfiguration",
        "elasticbeanstalk:DescribeApplicationVersions",
        "elasticbeanstalk:DescribeConfigurationOptions",
        "elasticbeanstalk:DescribeConfigurationSettings",
        "elasticbeanstalk:DescribeEnvironmentResources",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticbeanstalk:DescribeEvents",
        "elasticbeanstalk:DeleteEnvironmentConfiguration",
        "elasticbeanstalk:RebuildEnvironment",
        "elasticbeanstalk:RequestEnvironmentInfo",
        "elasticbeanstalk:RestartAppServer",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "elasticbeanstalk:SwapEnvironmentCNAMEs",
        "elasticbeanstalk:TerminateEnvironment",
        "elasticbeanstalk:UpdateApplicationVersion",
        "elasticbeanstalk:UpdateConfigurationTemplate",
        "elasticbeanstalk:UpdateEnvironment",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "elasticbeanstalk:ValidateConfigurationSettings"
      ],
```

```
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringNotEquals": {
          "elasticbeanstalk:InApplication": [
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
myapplication"
          ]
        }
      }
    }
  ]
}
```

Formato Amazon Resource Name per Elastic Beanstalk

Per specificare una risorsa per una policy IAM devi utilizzare l'Amazon Resource Name (ARN) di quella risorsa. Per Elastic Beanstalk, l'ARN ha il seguente formato.

```
arn:aws:elasticbeanstalk:region:account-id:resource-type/resource-path
```

Dove:

- *`region`* è la regione in cui risiede la risorsa (ad esempio, **`us-west-2`**).
- *`account-id`*è l'ID AWS dell'account, senza trattini (ad esempio,) **`123456789012`**
- *`resource-type`* identifica il tipo di risorsa Elastic Beanstalk, ad esempio `environment`. Consulta la tabella di seguito per l'elenco di tutti i tipi di risorse Elastic Beanstalk.
- *`resource-path`* è la porzione che identifica la risorsa specifica. Il percorso identifica in modo univoco una risorsa Elastic Beanstalk. Consulta la tabella di seguito per il formato del percorso della risorsa per ogni tipo di risorsa. Ad esempio, un ambiente è sempre associato a un'applicazione. Il percorso della risorsa per l'ambiente **`myEnvironment`** nell'applicazione **`myApp`** avrà il formato seguente:

  ```
  myApp/myEnvironment
  ```

Elastic Beanstalk offre diversi tipi di risorse che puoi specificare in una policy. La tabella riportata di seguito mostra il formato ARN per ogni tipo di risorsa e un esempio.

Tipo di risorsa	Formato per ARN
application	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:application/` *application-name* Esempio: **`arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App`**
applicationversion	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:applicationversion/` *application-name* `/`*version-label* Esempio: **`arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version`**
configurationtemplate	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:configurationtemplate/` *application-name* `/`*template-name* Esempio: **`arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template`**
environment	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:environment/` *application-name* `/`*environment-name* Esempio: **`arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/MyEnvironment`**
platform	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:platform/` *platform-name* `/`*platform-version* Esempio: **`arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/MyPlatform/1.0`**
solutionstack	`arn:aws:elasticbeanstalk:` *region*`::solutionstack/` *solutionstack-name*

Tipo di risorsa	Formato per ARN
	Ad esempio: **`arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7`**

Un ambiente, la versione dell'applicazione e il modello di configurazione sono sempre contenuti all'interno di un'applicazione specifica. Noterai che queste risorse dispongono tutte di un nome dell'applicazione nel percorso delle risorse in modo che siano identificate in maniera univoca dal nome della risorsa e dall'applicazione che le contiene. Sebbene gli stack di soluzioni siano utilizzati dai modelli e dagli ambienti di configurazione, gli stack di soluzioni non sono specifici di un'applicazione o di un AWS account e non contengono l'applicazione o AWS l'account. ARNs

Risorse e condizioni per le operazioni Elastic Beanstalk

Questa sezione descrive le risorse e le condizioni che puoi utilizzare nelle istruzioni di policy per concedere autorizzazioni che consentano l'esecuzione di operazioni Elastic Beanstalk specifiche su determinate risorse Elastic Beanstalk.

Le condizioni consentono di specificare le autorizzazioni per le risorse di cui l'operazione ha bisogno per il completamento dell'attività. Ad esempio, quando chiami l'operazione `CreateEnvironment`, è necessario specificare la versione dell'applicazione da distribuire e l'applicazione che contiene tale nome dell'applicazione. Quando imposti le autorizzazioni per l'operazione `CreateEnvironment`, è necessario specificare l'applicazione e la versione dell'applicazione su cui desideri agisca l'operazione utilizzando le condizioni `InApplication` e `FromApplicationVersion`.

Inoltre, è possibile specificare la configurazione dell'ambiente con uno stack di soluzioni (`FromSolutionStack`) o un modello di configurazione (`FromConfigurationTemplate`). La seguente istruzione di policy consente l'operazione `CreateEnvironment` di creare un ambiente con il nome **`myenv`** (specificato da `Resource`) nell'applicazione **`My App`** (specificata dalla condizione `InApplication`) utilizzando la versione dell'applicazione **`My Version`** (`FromApplicationVersion`) con una configurazione **`32bit Amazon Linux running Tomcat 7`** (`FromSolutionStack`):

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromApplicationVersion":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My
 Version"],
          "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
        }
      }
    }
  ]
}
```

> **Note**
>
> La maggior parte delle chiavi di condizione menzionate in questo argomento è specifica di Elastic Beanstalk, pertanto i loro nomi contengono il prefisso `elasticbeanstalk:`. Per brevità, abbiamo omesso tale prefisso dai nomi delle chiavi delle condizioni quando abbiamo menzionarle nelle seguenti sezioni. Ad esempio, abbiamo menzionato `InApplication` invece del nome completo `elasticbeanstalk:InApplication`.
>
> Al contrario, citiamo alcune chiavi di condizione utilizzate tra AWS i servizi e includiamo il loro `aws:` prefisso per evidenziare l'eccezione.
>
> Gli esempi di policy mostrano sempre i nomi delle chiavi di condizione completi, incluso il prefisso.

Sections

- Informazioni sulle policy per le operazioni Elastic Beanstalk

- Chiavi di condizione per operazioni Elastic Beanstalk

Informazioni sulle policy per le operazioni Elastic Beanstalk

La tabella seguente elenca tutte le operazioni Elastic Beanstalk, la risorsa che agisce su ciascuna operazione e le altre informazioni contestuali che possono essere fornite utilizzando le condizioni.

Le informazioni sulle policy per le operazioni Elastic Beanstalk includono risorse, condizioni, esempi e dipendenze

Risorsa	Condizioni	Istruzione di esempio
Operazione: `AbortEnvironmentUpdate`		

`application`
 	
`environment`
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo)
		La policy seguente consente all'utente di interrompere le operazioni di aggiornamento di ambiente per gli ambienti in un'applicazione denominata `My App`.
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:AbortEnvironmentUpdate" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`

Operazione: `CheckDNSAvailability`		
`"*"`	N/D	`{ "Version":"2012-10-17",`

Risorsa	Condizioni	Istruzione di esempio
		`"Statement": [ { "Action": [ "elasticbeanstalk:CheckDNSAvailability" ], "Effect": "Allow", "Resource": "*" } ] }`
Operazione: `ComposeEnvironments`		
`application`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	La policy seguente consente all'utente di comporre ambienti che appartengono a un'applicazione denominata `My App`. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:ComposeEnvironments" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/*" ] } ] }`
Operazione: `CreateApplication`		

Risorsa	Condizioni	Istruzione di esempio
`application`	`aws:RequestTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	Questo esempio consente l'operazione `CreateApplication` per creare applicazioni i cui nomi iniziano con **`DivA`**: `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreateApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/DivA*" ] } ] }`
Operazione: `CreateApplicationVersion`		

Risorsa	Condizioni	Istruzione di esempio
`applicationversion`	`InApplication` `aws:RequestTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	Questo esempio consente l'operazione `CreateApplicationVersion` per creare versioni dell'applicazione con qualsiasi nome (*****) nell'applicazione **My App**: `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `CreateConfigurationTemplate`		

Risorsa	Condizioni	Istruzione di esempio

`configurationtemplate`	`InApplication`
	
	`FromApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`FromEnvironment`
	
	`FromSolutionStack`
	
	`aws:RequestTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo)
		La policy seguente consente l'operazione `CreateConfigurationTemplate` per creare modelli di configurazione il cui nome inizia con **`My Template`** (`My Template*`) nell'applicazione **`My App`**:
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateConfigurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template*"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"], "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"] } } }] }`

Operazione: `CreateEnvironment`

Risorsa	Condizioni	Istruzione di esempio

`environment`	`InApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`FromSolutionStack`
	
	`aws:RequestTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo)
		La policy seguente consente l'operazione `CreateEnvironment` per creare un ambiente il cui nome è **`myenv`** nell'applicazione **`My App`** e utilizza lo stack di soluzioni **`32bit Amazon Linux running Tomcat 7`**:
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:CreateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"], "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7"] } } }]`

Risorsa	Condizioni	Istruzione di esempio
		`}`
Operazione: `CreatePlatformVersion`		
`platform`	`aws:RequestTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	Questo esempio consente l'operazione `CreatePlatformVersion` per creare versioni della piattaforma per la regione `us-east-2`, i cui nomi iniziano con **`us-east-2_`** : `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreatePlatformVersion" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*" ] } ] }`
Operazione: `CreateStorageLocation`		

Risorsa	Condizioni	Istruzione di esempio
`"*"`	N/D	`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreateStorageLocation" ], "Effect": "Allow", "Resource": "*" } ] }`
Operazione: `DeleteApplication`		
`application`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	La policy seguente consente l'operazione `DeleteApplication` per eliminare l'applicazione **`My App`**: `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeleteApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Operazione: `DeleteApplicationVersion`		

Risorsa	Condizioni	Istruzione di esempio
`applicationversion`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	La policy seguente consente l'operazione `DeleteApplicationVersion` per eliminare una versione dell'applicazione il cui nome è **`My Version`** nell'applicazione **`My App`**: `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:DeleteApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `DeleteConfigurationTemplate`		

Risorsa	Condizioni	Istruzione di esempio
`configurationtemplate`	`InApplication` (facoltativo) `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	La policy seguente consente l'operazione `DeleteConfigurationTemplate` per eliminare un modello di configurazione il cui nome è **`My Template`** nell'applicazione **`My App`**. Specificare il nome dell'applicazione come una condizione è facoltativo. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeleteConfigurationTemplate" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template" ] } ] }`
Operazione: `DeleteEnvironmentConfiguration`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication` (facoltativo)	La policy seguente consente l'operazione `DeleteEnvironmentConfiguration` per eliminare una bozza di configurazione per l'ambiente **myenv** nell'applicazione **My App**. Specificare il nome dell'applicazione come una condizione è facoltativo.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:DeleteEnvironmentConfiguration"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"
      ]
    }
  ]
}
```

Operazione: `DeletePlatformVersion`

Risorsa	Condizioni	Istruzione di esempio
`platform`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `DeletePlatformVersion` per eliminare le versioni della piattaforma per la regione `us-east-2`, i cui nomi iniziano con **`us-east-2_`** : `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeletePlatformVersion" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*" ] } ] }`
Operazione: `DescribeApplications`		

Risorsa	Condizioni	Istruzione di esempio
`application`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `DescribeApplications` per descrivere l'applicazione Mia App. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DescribeApplications" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Operazione: `DescribeApplicationVersions`		

Risorsa	Condizioni	Istruzione di esempio
`applicationversion`	`InApplication` (facoltativo) `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `DescribeApplicationVersions` per descrivere la versione dell'applicazione **`My Version`** nell'applicazione **`My App`**. Specificare il nome dell'applicazione come una condizione è facoltativo. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DescribeApplicationVersions" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version" ] } ] }`

Operazione: `DescribeConfigurationOptions`

Risorsa	Condizioni	Istruzione di esempio

`environment`
 	
`configurationtemplate`
 	
`solutionstack`
	`InApplication` (facoltativo)
	
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `DescribeConfigurationOptions` per descrivere le opzioni di configurazione per l'ambiente **`myenv`** nell'applicazione **`My App`**. Specificare il nome dell'applicazione come una condizione è facoltativo.
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeConfigurationOptions", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`

Operazione: `DescribeConfigurationSettings`		

Risorsa	Condizioni	Istruzione di esempio

`environment`
 	
`configurationtemplate`
	`InApplication` (facoltativo)
	
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `DescribeConfigurationSettings` per descrivere le impostazioni di configurazione per l'ambiente **`myenv`** nell'applicazione **`My App`**. Specificare il nome dell'applicazione come una condizione è facoltativo.
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeConfigurationSettings", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`

Operazione: `DescribeEnvironmentHealth`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'uso di `DescribeEnvironmentHealth` per recuperare l'utilizzo di informazioni sullo stato per un ambiente denominato **`myenv`**. `{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeEnvironmentHealth", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Operazione: `DescribeEnvironmentResources`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication` (facoltativo) `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La seguente politica consente all'`DescribeEnvironmentResources` azione di restituire e un elenco di AWS risorse per l'ambiente dell'**`myenv`**applicazione**`My App`**. Specificare il nome dell'applicazione come una condizione è facoltativo. `{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeEnvironmentResources", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Operazione: `DescribeEnvironments`		

Risorsa	Condizioni	Istruzione di esempio

`environment`	`InApplication` (facoltativo)
	
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `DescribeEnvironments` per descrivere la versione gli ambienti **`myenv`** e **`myotherenv`** nell'applicazione **`My App`**. Specificare il nome dell'applicazione come una condizione è facoltativo.
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": "elasticbeanstalk:DescribeEnvironments",`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv",`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App2/myotherenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Operazione: `DescribeEvents`		

Risorsa	Condizioni	Istruzione di esempio

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
	`InApplication`
	
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `DescribeEvents` per elencare le descrizioni degli eventi per l'ambiente **`myenv`** e la versione dell'applicazione **`My Version`** nell'applicazione **`My App`**.
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": "elasticbeanstalk:DescribeEvents", "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv", "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`

Operazione: `DescribeInstancesHealth`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	N/D	La policy seguente consente l'uso di `DescribeInstancesHealth` per recuperare l'utilizzo di informazioni sullo stato per le istanze in un ambiente denominato **`myenv`**. `{ "Version":"2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeInstancesHealth", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Operazione: `DescribePlatformVersion`		

Risorsa	Condizioni	Istruzione di esempio
`platform`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `DescribePlatformVersion` per eliminare le versioni della piattaforma per la regione `us-east-2`, i cui nomi iniziano con **`us-east-2_`** : `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DescribePlatformVersion" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*" ] } ] }`
Operazione: `ListAvailableSolutionStacks`		

Risorsa	Condizioni	Istruzione di esempio
`solutionstack`	N/D	La policy seguente consente l'operazione `ListAvailableSolutionStacks` per restituire solo lo stack di soluzioni **`32bit Amazon Linux running Tomcat 7`**. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:ListAvailableSolutionStacks" ], "Effect": "Allow", "Resource": "arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7" } ] }`

Operazione: `ListPlatformVersions`

Risorsa	Condizioni	Istruzione di esempio
`platform`	`aws:RequestTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo)	Questo esempio consente l'operazione `CreatePlatformVersion` per creare versioni della piattaforma per la regione `us-east-2`, i cui nomi iniziano con **`us-east-2_`** : `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:ListPlatformVersions" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/us-east-2_*" ] } ] }`

Operazione: `ListTagsForResource`

Risorsa	Condizioni	Istruzione di esempio

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `ListTagsForResource` per elencare i tag di risorse esistenti solo se dispongono di un tag denominato `stage` con il valore `test`:
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:ListTagsForResource" ], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag/stage": ["test"] } } } ] }`

Operazione: `RebuildEnvironment`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `RebuildEnvironment` per ricreare l'ambiente **`myenv`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RebuildEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `RequestEnvironmentInfo`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `RequestEnvironmentInfo` per compilare le informazioni sull'ambiente **`myenv`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RequestEnvironmentInfo"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `RestartAppServer`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication`	La policy seguente consente l'operazione `RestartAppServer` per riavviare il server del container dell'applicazione per l'ambiente **`myenv`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RestartAppServer"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `RetrieveEnvironmentInfo`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `RetrieveEnvironmentInfo` per recuperare le informazioni compilate sull'ambiente **`myenv`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:RetrieveEnvironmentInfo"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `SwapEnvironmentCNAMEs`		

Risorsa	Condizioni	Istruzione di esempio

`environment`	`InApplication` (facoltativo)
	
	`FromEnvironment` (facoltativo).
		La seguente politica consente di sostituire l'`SwapEnvironmentCNAMEs` azione con CNAMEs gli ambienti **`mysrcenv`** e**`mydestenv`** .
		
		`{`
		`  "Version":"2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:SwapEnvironmentCNAMEs"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mysrcenv",`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/mydestenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Operazione: `TerminateEnvironment`		

Risorsa	Condizioni	Istruzione di esempio
`environment`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `TerminateEnvironment` per terminare l'ambiente **`myenv`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:TerminateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: UpdateApplication		

Risorsa	Condizioni	Istruzione di esempio
`application`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `UpdateApplication` per aggiornare le proprietà dell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:UpdateApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Operazione: UpdateApplicationResourceLifecycle		

Risorsa	Condizioni	Istruzione di esempio
`application`	`aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `UpdateApplicationResourceLifecycle` per aggiornare le impostazioni del ciclo di vita dell'applicazione **My App**. `{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:UpdateApplicationResourceLifecycle" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Operazione: `UpdateApplicationVersion`		

Risorsa	Condizioni	Istruzione di esempio
`applicationversion`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `UpdateApplicationVersion` per aggiornare le proprietà della versione dell'applicazione **`My Version`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateApplicationVersion"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `UpdateConfigurationTemplate`		

Risorsa	Condizioni	Istruzione di esempio
`configurationtemplate`	`InApplication` `aws:ResourceTag/` *`key-name`* (facoltativo) `aws:TagKeys` (facoltativo).	La policy seguente consente l'operazione `UpdateConfigurationTemplate` per aggiornare le proprietà o le opzioni del modello di configurazione **`My Template`** nell'applicazione **`My App`**. `{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateConfigurationTemplate"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`
Operazione: `UpdateEnvironment`		

Risorsa	Condizioni	Istruzione di esempio

`environment`	`InApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `UpdateEnvironment` per aggiornare l'ambiente **`myenv`** nell'applicazione **`My App`** distribuendo la versione dell'applicazione **`My Version`**.
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:UpdateEnvironment"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"], "elasticbeanstalk:FromApplicationVersion": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"] } } }] }`

Azione: **`UpdateTagsForResource`** — `AddTags`

Risorsa	Condizioni	Istruzione di esempio

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (facoltativo).
	
	`aws:RequestTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		L'operazione `AddTags` è una delle due operazioni virtuali associate all'API `UpdateTagsForResource`.
		
		La policy seguente consente l'operazione `AddTags` per modificare i tag di risorse esistenti solo se dispongono di un tag denominato `stage` con il valore `test`:
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:AddTags" ], "Effect": "Allow", "Resource": "*", "Condition": { "StringEquals": { "aws:ResourceTag/stage": ["test"] } } } ] }`

Azione: **`UpdateTagsForResource`** — `RemoveTags`		

Risorsa	Condizioni	Istruzione di esempio

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (facoltativo).
	
	`aws:TagKeys` (facoltativo).
		L'operazione `RemoveTags` è una delle due operazioni virtuali associate all'API `UpdateTagsForResource`.
		
		La policy seguente nega l'operazione `RemoveTags` per richiedere l'eliminazione di un tag denominato `stage` da risorse esistenti:
		
		`{ "Version":"2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:RemoveTags" ], "Effect": "Deny", "Resource": "*", "Condition": { "ForAnyValue:StringEquals": { "aws:TagKeys": ["stage"] } } } ] }`

Operazione: `ValidateConfigurationSettings`		

Risorsa	Condizioni	Istruzione di esempio

`template`
 	
`environment`
	`InApplication`
	
	`aws:ResourceTag/` *`key-name`* (facoltativo)
	
	`aws:TagKeys` (facoltativo).
		La policy seguente consente l'operazione `ValidateConfigurationSettings` per convalidare le impostazioni di configurazione rispetto all'ambiente **`myenv`** nell'applicazione **`My App`**.
		
		`{ "Version":"2012-10-17", "Statement": [{ "Action": ["elasticbeanstalk:ValidateConfigurationSettings"], "Effect": "Allow", "Resource": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"], "Condition": { "StringEquals": { "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App"] } } }] }`

Chiavi di condizione per operazioni Elastic Beanstalk

Le chiavi consentono di specificare le condizioni che esprimono dipendenze, limitano le autorizzazioni o specificano vincoli per i parametri di input per un'operazione. Elastic Beanstalk supporta le seguenti chiavi.

`InApplication`

Specifica l'applicazione che contiene la risorsa su cui si attiva 'operazione.

Il seguente esempio consente l'operazione `UpdateApplicationVersion` per aggiornare le proprietà della versione dell'applicazione **`My Version`**. La condizione `InApplication` specifica **`My App`** come container per **`My Version`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:UpdateApplicationVersion"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My
 App/My Version"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"]
        }
      }
    }
  ]
}
```

`FromApplicationVersion`

Specifica una versione dell'applicazione come dipendenza o vincolo in un parametro di input.

Il seguente esempio consente l'operazione `UpdateEnvironment` per aggiornare l'ambiente **`myenv`** nell'applicazione **`My App`**. La condizione `FromApplicationVersion` vincola il parametro `VersionLabel` per consentire solo alla versione dell'applicazione **`My Version`** di aggiornare l'ambiente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:UpdateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromApplicationVersion":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/
My Version"]
        }
      }
    }
  ]
}
```

`FromConfigurationTemplate`

Specifica un modello di configurazione come dipendenza o vincolo in un parametro di input.

Il seguente esempio consente l'operazione `UpdateEnvironment` per aggiornare l'ambiente **`myenv`** nell'applicazione **`My App`**. La condizione `FromConfigurationTemplate` vincola il parametro `TemplateName` per consentire solo al modello di configurazione **`My Template`** di aggiornare l'ambiente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
```

```
        "elasticbeanstalk:UpdateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
myenv"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
          "elasticbeanstalk:FromConfigurationTemplate":
 ["arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My
 App/My Template"]
        }
      }
    }
  ]
}
```

`FromEnvironment`

Specifica un ambiente come dipendenza o vincolo in un parametro di input.

L'esempio seguente consente all'`SwapEnvironmentCNAMEs`azione di sostituire l' CNAMEs accesso **`My App`** per tutti gli ambienti i cui nomi iniziano con **`mysrcenv`** e, ad **`mydestenv`** eccezione di quelli i cui nomi iniziano con **`mysrcenvPROD*`** e**`mydestenvPROD*`**.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:SwapEnvironmentCNAMEs"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mysrcenv*",
        "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/
mydestenv*"
```

```
      ],
      "Condition": {
        "ArnNotLike": {
          "elasticbeanstalk:FromEnvironment": [
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My
 App/mysrcenvPROD*",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My
 App/mydestenvPROD*"
          ]
        }
      }
    }
  ]
}
```

`FromSolutionStack`

Specifica uno stack di soluzioni come dipendenza o vincolo in un parametro di input.

La policy seguente consente l'operazione `CreateConfigurationTemplate` per creare modelli di configurazione il cui nome inizia con **`My Template`** (`My Template*`) nell'applicazione **`My App`**. La condizione `FromSolutionStack` vincola il parametro `solutionstack` per consentire solo lo stack di soluzioni **`32bit Amazon Linux running Tomcat 7`** come valore di input per quel parametro.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticbeanstalk:CreateConfigurationTemplate"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:us-
east-2:123456789012:configurationtemplate/My App/My Template*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticbeanstalk:InApplication": ["arn:aws:elasticbeanstalk:us-
east-2:123456789012:application/My App"],
```

```
          "elasticbeanstalk:FromSolutionStack": ["arn:aws:elasticbeanstalk:us-
east-2::solutionstack/32bit Amazon Linux running Tomcat 7"]
        }
      }
    }
  ]
}
```

`aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, `aws:TagKeys`

Specifica condizioni basate sui tag. Per informazioni dettagliate, vedi Utilizzo dei tag per controllare l'accesso alle risorse Elastic Beanstalk.

Utilizzo dei tag per controllare l'accesso alle risorse Elastic Beanstalk

Questo argomento spiega come il controllo degli accessi basato su tag può aiutarti a creare e gestire le policy IAM.

Possiamo utilizzare le condizioni nelle dichiarazioni relative alle policy degli utenti di IAM per configurare le autorizzazioni per l'accesso di Elastic Beanstalk alle risorse. Per ulteriori informazioni sulle condizioni delle dichiarazioni politiche, consulta. Risorse e condizioni per le operazioni Elastic Beanstalk L'utilizzo di tag nelle condizioni è un modo per controllare l'accesso alle risorse e alle richieste. Per informazioni sull'uso dei tag per le risorse Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Durante la progettazione di policy IAM, potrebbe essere necessario impostare autorizzazioni granulari concedendo l'accesso a risorse specifiche. Poiché il numero di risorse che puoi gestire cresce, questa operazione diventa più difficile. Il tagging delle risorse e l'utilizzo di tag in condizioni di dichiarazione di policy possono semplificare questa attività. Puoi concedere l'accesso in blocco a qualsiasi risorsa con un determinato tag. Quindi applicare ripetutamente questo tag a risorse pertinenti, durante la creazione o in seguito.

I tag possono essere collegati alla risorsa o trasferiti nella richiesta verso servizi che supportano il tagging. In Elastic Beanstalk le risorse possono avere tag e le operazioni possono includere tag. Quando crei una policy IAM, puoi utilizzare le chiavi delle condizioni dei tag per controllare le seguenti condizioni:

- Quali gli utenti possono eseguire operazioni su un ambiente in base ai tag di cui dispone già;
- Quali tag possono essere passati in una richiesta di operazione;

- Se delle chiavi di tag specifiche possono essere utilizzate in una richiesta.

Per la sintassi completa e la semantica delle chiavi di condizione di tag, consulta Controllo degli accessi tramite tag nella Guida per l'utente di IAM.

Esempi di condizioni relative ai tag nelle politiche

I seguenti esempi mostrano come specificare le condizioni dei tag nelle policy per gli utenti Elastic Beanstalk.

Example 1: Limitare le operazioni in base ai tag nella richiesta

La policy utente gestita da AdministratorAccessElastic Beanstalk - AWSElastic Beanstalk offre agli utenti autorizzazioni illimitate per eseguire qualsiasi azione di Elastic Beanstalk su qualsiasi risorsa gestita da Elastic Beanstalk.

La policy seguente limita questa capacità e nega agli utenti non autorizzati l'autorizzazione a creare ambienti di produzione Elastic Beanstalk. A tale scopo, nega l'operazione `CreateEnvironment` se nella richiesta è specificato un tag denominato `stage` con uno dei valori `gamma` o `prod`. Inoltre, la policy impedisce a questi utenti non autorizzati di eseguire manomissioni nella fase di ambienti di produzione non consentendo operazioni di modifica volte a includere questi stessi valori di tag o a rimuovere completamente il tag `stage`. L'amministratore di un cliente deve collegare questa policy IAM a utenti IAM non autorizzati oltre alla policy gestita dall'utente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stage": ["gamma", "prod"]
        }
```

```
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:RemoveTags"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Example 2: Limitare le operazioni in base ai tag delle risorse

La policy utente gestita da AdministratorAccessElastic Beanstalk - AWSElastic Beanstalk offre agli utenti autorizzazioni illimitate per eseguire qualsiasi azione di Elastic Beanstalk su qualsiasi risorsa gestita da Elastic Beanstalk.

La policy seguente limita questa capacità e nega agli utenti non autorizzati l'autorizzazione a eseguire operazioni sugli ambienti di produzione Elastic Beanstalk. A tale scopo, nega operazioni specifiche se l'ambiente ha un tag denominato `stage` con uno dei valori `gamma` o `prod`. L'amministratore di un cliente deve collegare questa policy IAM a utenti IAM non autorizzati oltre alla policy gestita dall'utente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "elasticbeanstalk:AddTags",
        "elasticbeanstalk:RemoveTags",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticbeanstalk:TerminateEnvironment",
```

```
        "elasticbeanstalk:UpdateEnvironment",
        "elasticbeanstalk:ListTagsForResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": ["gamma", "prod"]
        }
      }
    }
  ]
}
```

Example 3: Consentire operazioni in base ai tag nella richiesta

La policy seguente concede agli utenti l'autorizzazione per creare applicazioni di sviluppo Elastic Beanstalk.

A questo scopo, consente le operazioni `CreateApplicatione` `AddTags` se la richiesta specifica un tag denominato `stage` con il valore `development`. La condizione `aws:TagKeys` garantisce che l'utente non possa aggiungere altre chiavi di tag. In particolare, garantisce la distinzione tra maiuscole e minuscole della chiave di tag `stage`. Nota che questa policy è utile per gli utenti IAM a cui non è allegata la policy Elastic Beanstalk AdministratorAccess - AWSElasticBeanstalk managed user. La policy gestita fornisce agli utenti l'autorizzazione illimitata per eseguire qualsiasi operazione Elastic Beanstalk su qualsiasi risorsa gestita da Elastic Beanstalk.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/stage": "development"
```

```
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["stage"]
        }
      }
    }
  ]
}
```

Example 4: Consentire operazioni in base ai tag delle risorse

La policy seguente concede agli utenti l'autorizzazione per eseguire operazioni, e ottenere informazioni, su applicazioni di sviluppo Elastic Beanstalk.

A questo scopo, consente operazioni specifiche se l'applicazione dispone di un tag denominato `stage` con il valore `development`. La condizione `aws:TagKeys` garantisce che l'utente non possa aggiungere altre chiavi di tag. In particolare, garantisce la distinzione tra maiuscole e minuscole della chiave di tag `stage`. Nota che questa policy è utile per gli utenti IAM a cui non è allegata la policy Elastic Beanstalk AdministratorAccess - AWSElasticBeanstalk managed user. La policy gestita fornisce agli utenti l'autorizzazione illimitata per eseguire qualsiasi operazione Elastic Beanstalk su qualsiasi risorsa gestita da Elastic Beanstalk.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticbeanstalk:UpdateApplication",
        "elasticbeanstalk:DeleteApplication",
        "elasticbeanstalk:DescribeApplications"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/stage": "development"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["stage"]
```

```
        }
      }
    }
  ]
}
```

Policy di esempio basate su policy gestite

Questa sezione illustra come controllare l'accesso degli utenti AWS Elastic Beanstalk e include policy di esempio che forniscono l'accesso richiesto per scenari comuni. Queste policy derivano dalle policy gestite Elastic Beanstalk. Per informazioni su come collegare policy gestite a utenti e gruppi, vedi Gestione delle policy utente Elastic Beanstalk.

In questo scenario, Example Corp. è un'azienda di software con tre team responsabili del sito Web dell'azienda: amministratori che gestiscono l'infrastruttura, sviluppatori che creano il software per il sito Web e un team di QA che verifica il sito Web. Per aiutare a gestire le autorizzazioni per le risorse Elastic Beanstalk, Example Corp. crea tre gruppi a cui appartengono i membri di ogni team: amministratori, sviluppatori e collaudatori. Example Corp. vuole che il gruppo amministratori abbia l'accesso completo a tutte le applicazioni, a tutti gli ambienti e alle risorse sottostanti in modo che possano creare, risolvere i problemi ed eliminare tutti gli asset Elastic Beanstalk. Gli sviluppatori necessitano delle autorizzazioni per visualizzare tutti gli asset Elastic Beanstalk e per creare e distribuire le versioni delle applicazioni. Gli sviluppatori non devono essere in grado di creare nuove applicazioni o ambienti o di terminare ambienti in esecuzione. I collaudatori devono visualizzare tutte le risorse Elastic Beanstalk per monitorare e testare le applicazioni. I collaudatori non devono essere in grado di apportare modifiche alle risorse Elastic Beanstalk.

Le seguenti policy di esempio forniscono le autorizzazioni necessarie per ciascun gruppo.

Esempio 1: gruppo di amministratori: tutti gli Elastic Beanstalk e il servizio correlato APIs

La policy seguente fornisce agli utenti le autorizzazioni per tutte le operazioni necessarie per utilizzare Elastic Beanstalk. Questa policy consente inoltre a Elastic Beanstalk di effettuare il provisioning e di gestire le risorse a tuo nome nei seguenti servizi. Elastic Beanstalk si basa su questi servizi aggiuntivi per effettuare il provisioning delle risorse sottostanti durante la creazione di un ambiente.

- Amazon Elastic Compute Cloud

- Elastic Load Balancing
- Auto Scaling
- Amazon CloudWatch
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Relational Database Service
- CloudFormation

Importante: questa policy è un esempio. Offre un'ampia gamma di autorizzazioni per i servizi AWS utilizzati da Elastic Beanstalk per gestire applicazioni e ambienti. Ad esempio, `ec2:*` consente a un utente AWS Identity and Access Management (IAM) di eseguire qualsiasi azione su qualsiasi risorsa Amazon EC2 nell' AWS account. Queste autorizzazioni non sono limitate alle risorse che utilizzi con Elastic Beanstalk. Come best practice, dovresti concedere agli individui solo le autorizzazioni necessarie per eseguire le proprie funzioni.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticbeanstalk:*",
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"
      ],
      "Resource" : "*"
    }
  ]
}
```

Esempio 2: gruppo Sviluppatori - Tutte le operazioni tranne quelle con privilegi elevati

La policy seguente nega il permesso di creare applicazioni e ambienti, ma consente tutte le altre operazioni Elastic Beanstalk.

Importante: questa policy è un esempio. Offre un'ampia gamma di autorizzazioni per i prodotti AWS che Elastic Beanstalk impiega per gestire applicazioni e ambienti. Ad esempio, `ec2:*` consente a un utente IAM di eseguire qualsiasi operazione su una risorsa Amazon EC2 nell'account AWS . Queste autorizzazioni non sono limitate alle risorse che utilizzi con Elastic Beanstalk. Come best practice, dovresti concedere agli individui solo le autorizzazioni necessarie per eseguire le proprie funzioni.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Action" : [
        "elasticbeanstalk:CreateApplication",
        "elasticbeanstalk:CreateEnvironment",
        "elasticbeanstalk:DeleteApplication",
        "elasticbeanstalk:RebuildEnvironment",
        "elasticbeanstalk:SwapEnvironmentCNAMEs",
        "elasticbeanstalk:TerminateEnvironment"],
      "Effect" : "Deny",
      "Resource" : "*"
    },
    {
      "Action" : [
        "elasticbeanstalk:*",
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"],
      "Effect" : "Allow",
      "Resource" : "*"
    }
  ]
```

```
}
```

Esempio 3: collaudatori - Sola visualizzazione

La policy seguente consente l'accesso in sola lettura a tutte le applicazioni, alle versioni delle applicazioni, agli eventi e agli ambienti. Non consente di eseguire alcuna operazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticbeanstalk:Check*",
        "elasticbeanstalk:Describe*",
        "elasticbeanstalk:List*",
        "elasticbeanstalk:RequestEnvironmentInfo",
        "elasticbeanstalk:RetrieveEnvironmentInfo",
        "ec2:Describe*",
        "elasticloadbalancing:Describe*",
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:List*",
        "cloudwatch:Get*",
        "s3:Get*",
        "s3:List*",
        "sns:Get*",
        "sns:List*",
        "rds:Describe*",
        "cloudformation:Describe*",
        "cloudformation:Get*",
        "cloudformation:List*",
        "cloudformation:Validate*",
        "cloudformation:Estimate*"
      ],
      "Resource" : "*"
    }
  ]
}
```

Policy di esempio basate su autorizzazioni a livello di risorsa

Questa sezione esamina un caso d'uso per il controllo delle autorizzazioni utente per le operazioni Elastic Beanstalk che accedono a specifiche risorse Elastic Beanstalk. Esamineremo le policy di esempio che supportano il caso d'uso. Per ulteriori informazioni sulle policy per le risorse Elastic Beanstalk, consulta Creazione di una policy utente personalizzata. Per informazioni su come collegare le policy a utenti e gruppi, consultare Gestione di policy IAM in Utilizzo di AWS Identity and Access Management.

In questo caso d'uso Example Corp. è una piccola società di consulenza che sviluppa applicazioni per due clienti diversi. John è il responsabile di sviluppo che supervisiona lo sviluppo delle due applicazioni Elastic Beanstalk, app1 e app2. John si occupa dello sviluppo e di alcuni test sulle due applicazioni ed è l'unico che può aggiornare l'ambiente di produzione per entrambe le applicazioni. Queste sono le autorizzazioni necessarie per app1 e app2:

- Visualizzazione di applicazioni, versioni di applicazioni, ambienti e modelli di configurazione
- Creazione di versioni di applicazioni e relativa distribuzione nell'ambiente di gestione temporanea
- Aggiornamento dell'ambiente di produzione
- Creazione e terminazione di ambienti

Jill è un tester che deve accedere per visualizzare le risorse seguenti per monitorare e testare le due applicazioni: applicazioni, versioni di applicazioni, ambienti e modelli di configurazione. Tuttavia, non deve essere in grado di apportare modifiche alle risorse Elastic Beanstalk.

Jack è lo sviluppatore per app1 che deve accedere per visualizzare tutte le risorse per app1 e deve creare versioni di applicazione per app1 e distribuirle nell'ambiente di gestione temporanea.

Judy è l'amministratore dell' AWS account di Example Corp. Ha creato utenti IAM per John, Jill e Jack e attribuisce le seguenti politiche a tali utenti per concedere le autorizzazioni appropriate alle applicazioni app1 e app2.

Esempio 1: John - manager di sviluppo per app1, app2

Abbiamo suddiviso la policy di John in tre policy separate, in modo che siano più facili da leggere e gestire. Insieme, offrono a John le autorizzazioni necessarie per eseguire le operazioni di sviluppo, testing e distribuzione sulle due applicazioni.

La prima policy specifica le azioni per Auto Scaling, Amazon S3, Amazon EC2, Amazon SNS CloudWatch, Elastic Load Balancing, Amazon RDS e. CloudFormation Elastic Beanstalk si basa su questi servizi aggiuntivi per effettuare il provisioning delle risorse sottostanti durante la creazione di un ambiente.

Importante: questa policy è un esempio. Fornisce un ampio set di autorizzazioni ai AWS prodotti utilizzati da Elastic Beanstalk per gestire applicazioni e ambienti. Ad esempio, `ec2:*` consente a un utente IAM di eseguire qualsiasi azione su qualsiasi risorsa Amazon EC2 nell' AWS account. Queste autorizzazioni non sono limitate alle risorse che utilizzi con Elastic Beanstalk. Come best practice, dovresti concedere agli individui solo le autorizzazioni necessarie per eseguire le proprie funzioni.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Action": [
   "ec2:*",
   "ecs:*",
   "ecr:*",
   "elasticloadbalancing:*",
   "autoscaling:*",
   "cloudwatch:*",
   "s3:*",
   "sns:*",
   "cloudformation:*",
   "dynamodb:*",
   "rds:*",
   "sqs:*",
   "logs:*",
   "iam:GetPolicyVersion",
   "iam:GetRole",
   "iam:ListRolePolicies",
   "iam:ListAttachedRolePolicies",
   "iam:ListInstanceProfiles",
   "iam:ListRoles",
   "iam:ListServerCertificates",
   "acm:DescribeCertificate",
   "acm:ListCertificates",
   "codebuild:CreateProject",
```

```
   "codebuild:DeleteProject",
   "codebuild:BatchGetBuilds",
   "codebuild:StartBuild"
   ],
   "Resource": "*"
   },
   {
   "Effect": "Allow",
   "Action": [
   "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::111122223333:role/MyRole"
   }
   ]
   }
```

La seconda policy specifica le operazioni Elastic Beanstalk che John può eseguire sulle risorse app1 e app2. L'istruzione `AllCallsInApplications` consente tutte le operazioni Elastic Beanstalk (`"elasticbeanstalk:*"`) eseguite su tutte le risorse all'interno di app1 e app2 (ad esempio, `elasticbeanstalk:CreateEnvironment`). L'istruzione `AllCallsOnApplications` consente tutte le operazioni Elastic Beanstalk (`"elasticbeanstalk:*"`) sulle risorse delle applicazioni app1 e app2 (ad esempio `elasticbeanstalk:DescribeApplications`, `elasticbeanstalk:UpdateApplication` e così via). L'istruzione `AllCallsOnSolutionStacks` consente tutte le operazioni Elastic Beanstalk (`"elasticbeanstalk:*"`) per le risorse dello stack di soluzioni (ad esempio `elasticbeanstalk:ListAvailableSolutionStacks`).

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllCallsInApplications",
         "Action":[
            "elasticbeanstalk:*"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
```

```
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1",
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app2"
               ]
            }
         }
      },
      {
         "Sid":"AllCallsOnApplications",
         "Action":[
            "elasticbeanstalk:*"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
         ]
      },
      {
         "Sid":"AllCallsOnSolutionStacks",
         "Action":[
            "elasticbeanstalk:*"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
         ]
      }
   ]
}
```

La terza policy specifica le operazioni Elastic Beanstalk per cui la seconda policy richiede le autorizzazioni per essere completate. L'istruzione `AllNonResourceCalls` consente l'operazione `elasticbeanstalk:CheckDNSAvailability`, necessaria per chiamare `elasticbeanstalk:CreateEnvironment` e altre operazioni. Consente anche l'operazione `elasticbeanstalk:CreateStorageLocation`, necessaria per

`elasticbeanstalk:CreateApplication`, `elasticbeanstalk:CreateEnvironment` e altre operazioni.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability",
            "elasticbeanstalk:CreateStorageLocation"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Esempio 2: Jill - tester per app1, app2

Abbiamo suddiviso la policy di Jill in tre policy separate, in modo che siano più facili da leggere e gestire. Insieme, offrono a Jill le autorizzazioni necessarie per eseguire le operazioni di testing e monitoraggio sulle due applicazioni.

La prima policy specifica e specifica `Get*` le azioni su Auto Scaling `Describe*List*`, Amazon S3, Amazon EC2, Amazon SNS, Elastic Load Balancing CloudWatch, CloudFormation Amazon RDS e (per i tipi di container non legacy) in modo che le azioni Elastic Beanstalk siano in grado di recuperare le informazioni pertinenti sulle risorse sottostanti delle applicazioni app1 e app2.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
         "Effect":"Allow",
         "Action":[
            "ec2:Describe*",
            "elasticloadbalancing:Describe*",
            "autoscaling:Describe*",
            "cloudwatch:Describe*",
            "cloudwatch:List*",
            "cloudwatch:Get*",
            "s3:Get*",
            "s3:List*",
            "sns:Get*",
            "sns:List*",
            "rds:Describe*",
            "cloudformation:Describe*",
         "cloudformation:Get*",
         "cloudformation:List*",
         "cloudformation:Validate*",
         "cloudformation:Estimate*"
         ],
         "Resource":"*"
      }
   ]
}
```

La seconda policy specifica le operazioni Elastic Beanstalk che Jill può eseguire sulle risorse app1 e app2. L'istruzione `AllReadCallsInApplications` consente a Jill di chiamare le operazioni `Describe*` e le operazioni informative sull'ambiente. L'istruzione `AllReadCallsOnApplications` le consente di chiamare le operazioni `DescribeApplications` e `DescribeEvents` sulle risorse delle applicazioni app1 e app2. L'istruzione `AllReadCallsOnSolutionStacks` le consente di visualizzare operazioni che coinvolgono le risorse dello stack di soluzioni (`ListAvailableSolutionStacks`, `DescribeConfigurationOptions` e `ValidateConfigurationSettings`).

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllReadCallsInApplications",
```

```
         "Action":[
            "elasticbeanstalk:Describe*",
            "elasticbeanstalk:RequestEnvironmentInfo",
            "elasticbeanstalk:RetrieveEnvironmentInfo"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1",
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app2"
               ]
            }
         }
      },
      {
         "Sid":"AllReadCallsOnApplications",
         "Action":[
            "elasticbeanstalk:DescribeApplications",
            "elasticbeanstalk:DescribeEvents"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1",
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app2"
         ]
      },
      {
         "Sid":"AllReadCallsOnSolutionStacks",
         "Action":[
            "elasticbeanstalk:ListAvailableSolutionStacks",
            "elasticbeanstalk:DescribeConfigurationOptions",
            "elasticbeanstalk:ValidateConfigurationSettings"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
         ]
      }
```

```
   ]
}
```

La terza policy specifica le operazioni Elastic Beanstalk per cui la seconda policy richiede le autorizzazioni per essere completate. L'istruzione `AllNonResourceCalls` le consente l'operazione `elasticbeanstalk:CheckDNSAvailability`, che è necessaria per alcune azioni di visualizzazione.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Esempio 3: Jack - sviluppatore per app1

Abbiamo suddiviso la policy di Jack in tre policy separate, in modo che siano più facili da leggere e gestire. Insieme, danno a Jack le autorizzazioni necessarie per eseguire operazioni di testing, monitoraggio e distribuzione sulla risorsa app1.

La prima policy specifica le azioni su Auto Scaling, Amazon S3, Amazon EC2, Amazon SNS, Elastic Load Balancing CloudWatch, CloudFormation Amazon RDS e (per i tipi di container non legacy) in modo che le azioni Elastic Beanstalk siano in grado di visualizzare e utilizzare le risorse sottostanti di app1. Per un elenco di tipi di container non legacy supportati, consulta the section called “Perché alcune versioni di piattaforma sono contrassegnate come legacy?”.

Importante: questa policy è un esempio. Fornisce un ampio set di autorizzazioni ai AWS prodotti utilizzati da Elastic Beanstalk per gestire applicazioni e ambienti. Ad esempio, `ec2:*` consente a un utente IAM di eseguire qualsiasi azione su qualsiasi risorsa Amazon EC2 nell' AWS account. Queste autorizzazioni non sono limitate alle risorse che utilizzi con Elastic Beanstalk. Come best practice, dovresti concedere agli individui solo le autorizzazioni necessarie per eseguire le proprie funzioni.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "ec2:*",
            "elasticloadbalancing:*",
            "autoscaling:*",
            "cloudwatch:*",
            "s3:*",
            "sns:*",
            "rds:*",
            "cloudformation:*"
         ],
         "Resource":"*"
      }
   ]
}
```

La seconda policy specifica le operazioni Elastic Beanstalk che Jack può eseguire sulla risorsa app1.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllReadCallsAndAllVersionCallsInApplications",
         "Action":[
            "elasticbeanstalk:Describe*",
            "elasticbeanstalk:RequestEnvironmentInfo",
```

```
            "elasticbeanstalk:RetrieveEnvironmentInfo",
            "elasticbeanstalk:CreateApplicationVersion",
            "elasticbeanstalk:DeleteApplicationVersion",
            "elasticbeanstalk:UpdateApplicationVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1"
               ]
            }
         }
      },
      {
         "Sid":"AllReadCallsOnApplications",
         "Action":[
            "elasticbeanstalk:DescribeApplications",
            "elasticbeanstalk:DescribeEvents"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/app1"
         ]
      },
      {
         "Sid":"UpdateEnvironmentInApplications",
         "Action":[
            "elasticbeanstalk:UpdateEnvironment"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/app1/
app1-staging*"
         ],
         "Condition":{
            "StringEquals":{
               "elasticbeanstalk:InApplication":[
                  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/
app1"
```

```
                ]
            },
            "ArnLike":{
                "elasticbeanstalk:FromApplicationVersion":[
                    "arn:aws:elasticbeanstalk:us-
east-2:123456789012:applicationversion/app1/*"
                ]
            }
        }
    },
    {
        "Sid":"AllReadCallsOnSolutionStacks",
        "Action":[
            "elasticbeanstalk:ListAvailableSolutionStacks",
            "elasticbeanstalk:DescribeConfigurationOptions",
            "elasticbeanstalk:ValidateConfigurationSettings"
        ],
        "Effect":"Allow",
        "Resource":[
            "arn:aws:elasticbeanstalk:us-east-2::solutionstack/*"
        ]
    }
  ]
}
```

La terza policy specifica le operazioni Elastic Beanstalk per cui la seconda policy richiede le autorizzazioni per essere completate. L'istruzione `AllNonResourceCalls` consente l'operazione `elasticbeanstalk:CheckDNSAvailability`, necessaria per chiamare `elasticbeanstalk:CreateEnvironment` e altre operazioni. Consente anche l'operazione `elasticbeanstalk:CreateStorageLocation`, necessaria per `elasticbeanstalk:CreateEnvironment` e altre operazioni.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
        "Sid":"AllNonResourceCalls",
        "Action":[
            "elasticbeanstalk:CheckDNSAvailability",
```

```
            "elasticbeanstalk:CreateStorageLocation"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Impedire l'accesso ai bucket Amazon S3 tra ambienti

Questo argomento spiega come le policy gestite possono consentire l'accesso ai bucket S3 tra ambienti e come è possibile creare policy personalizzate per gestire questo tipo di accesso.

Elastic Beanstalk fornisce policy gestite per AWS gestire le risorse richieste dagli ambienti Elastic Beanstalk del tuo account. AWS Le autorizzazioni fornite di default a un'applicazione del tuo AWS account hanno accesso alle risorse S3 che appartengono ad altre applicazioni dello stesso account. AWS

Se il tuo AWS account esegue più applicazioni Beanstalk, puoi limitare la sicurezza delle tue policy creando una policy personalizzata da allegare al tuo ruolo di servizio o profilo di istanza per ogni ambiente. Puoi quindi limitare le autorizzazioni S3 nella tua policy personalizzata a un ambiente specifico.

> **Note**
>
> Tieni presente che sei responsabile del mantenimento della tua politica personalizzata. Se una policy gestita da Elastic Beanstalk su cui si basa la policy personalizzata cambia, dovrai modificare la policy personalizzata con le rispettive modifiche alla policy di base. Per una cronologia delle modifiche delle policy gestite da Elastic Beanstalk, consulta. Elastic AWS Beanstalk: aggiornamenti alle policy gestite

Esempio di autorizzazioni limitate

L'esempio seguente si basa sulla politica AWSElasticBeanstalkWebTiergestita.

La politica predefinita include le seguenti righe per le autorizzazioni ai bucket S3. Questa politica predefinita non limita le azioni del bucket S3 a ambienti o applicazioni specifici.

```
{
   "Sid" : "BucketAccess",
   "Action" : [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
     ],
   "Effect" : "Allow",
   "Resource" : [
      "arn:aws:s3:::elasticbeanstalk-*",
      "arn:aws:s3:::elasticbeanstalk-*/*"
     ]
}
```

È possibile limitare l'accesso qualificando risorse specifiche per un ruolo di servizio specificato come a. `Principal` L'esempio seguente fornisce le `aws-elasticbeanstalk-ec2-role-my-example-env` autorizzazioni personalizzate per i ruoli di servizio ai bucket S3 nell'ambiente con id. `my-example-env-ID`

Example Concedi le autorizzazioni solo ai bucket S3 di un ambiente specifico

```
{
   "Sid": "BucketAccess",
   "Action": [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
    ],
   "Effect": "Allow",
   "Principal": {
      "AWS": "arn:aws:iam::...:role/aws-elasticbeanstalk-ec2-role-my-example-env"
     },
   "Resource": [
      "arn:aws:s3:::elasticbeanstalk-my-region-account-id-12345",
      "arn:aws:s3:::elasticbeanstalk-my-region-account-id-12345/resources/environments/
my-example-env-ID/*"
    ]
}
```

> **Note**
>
> L'ARN della risorsa deve includere l'ID dell'ambiente Elastic Beanstalk (non il nome dell'ambiente). Puoi ottenere l'id dell'ambiente dalla console Elastic Beanstalk nella pagina di panoramica dell'ambiente. Puoi anche usare il comando AWS CLI describe-environments per ottenere queste informazioni.

Per ulteriori informazioni su come aggiornare le autorizzazioni dei bucket S3 per gli ambienti Elastic Beanstalk, consulta le seguenti risorse:

- Utilizzo di Elastic Beanstalk con Amazon S3 in questa guida
- Tipi di risorse definiti da Amazon S3 nella guida di riferimento per l'autorizzazione del servizio
- Formato ARN nella Guida per l'utente IAM

Utilizzo di Elastic Beanstalk con Amazon RDS

Questa sezione spiega come utilizzare Elastic Beanstalk con Amazon Relational Database Service (Amazon RDS) per configurare, gestire e scalare un database relazionale. Spieghiamo alcuni concetti sulla configurazione e forniamo consigli. Quindi ti guideremo attraverso il processo per creare e connetterti a un Amazon RDS.

Esistono due opzioni per iniziare:

- Creare un nuovo database Amazon RDS.
- Iniziare con un database precedentemente creato da Elastic Beanstalk e successivamente disaccoppiato da un ambiente Beanstalk. Per ulteriori informazioni, consulta the section called “Database”.

Seleziona l'approccio

È possibile utilizzare entrambi gli approcci per eseguire un'istanza database in Amazon RDS e configurare l'applicazione in modo da connettersi all'avvio. È possibile connettere più ambienti a un database.

> **Note**
>
> Se è la prima volta che utilizzi un'istanza database con la tua applicazione, consigliamo di aggiungere un database a un ambiente di test con la console Elastic Beanstalk. In questo modo, è possibile verificare se l'applicazione può leggere le proprietà dell'ambiente, costruire una stringa di connessione e connettersi a un'istanza database, senza la configurazione aggiuntiva necessaria per un database autonomo. Per ulteriori informazioni, consulta Aggiunta di un database all'ambiente Elastic Beanstalk.

Configurare un gruppo di sicurezza

Per consentire alle EC2 istanze Amazon del tuo ambiente di connettersi a un database esterno, configura un gruppo di sicurezza aggiuntivo per il gruppo Auto Scaling associato al tuo ambiente. È possibile collegare lo stesso gruppo di sicurezza associato all'istanza del database. In alternativa, è possibile utilizzare un gruppo di sicurezza separato. Se si collega un gruppo di sicurezza diverso, è necessario configurare il gruppo di sicurezza collegato al database per consentire l'accesso in entrata da questo gruppo di sicurezza.

> **Note**
>
> È possibile connettere l'ambiente a un database aggiungendo una regola al gruppo di sicurezza collegato al database. Questa regola deve consentire l'accesso in entrata dal gruppo di sicurezza generato automaticamente che Elastic Beanstalk collega al gruppo Auto Scaling per il proprio ambiente. Tuttavia, creando questa regola, si crea anche una dipendenza tra i due gruppi di sicurezza. Successivamente, quando tenti di terminare l'ambiente, Elastic Beanstalk non sarà in grado di eliminare il gruppo di sicurezza dell'ambiente, poiché il gruppo di sicurezza del database dipende da esso.

Configura la connessione al database

Dopo l'avvio dell'istanza di database e la configurazione dei gruppi di sicurezza, è possibile trasferire le informazioni di connessione, come endpoint password, all'applicazione impiegando le proprietà dell'ambiente. Questo è lo stesso meccanismo usato in background da Elastic Beanstalk quando esegui un'istanza database nell'ambiente.

Per un livello di sicurezza più elevato, puoi archiviare le informazioni di connessione in Amazon S3 e configurare Elastic Beanstalk per recuperarle durante la distribuzione. Con i file di configurazione

(`.ebextensions`), puoi configurare le istanze nell'ambiente per recuperare in modo sicuro i file da Amazon S3 quando distribuisci l'applicazione.

Argomenti

- Avvio e connessione a un'istanza Amazon RDS esterna in un VPC predefinito
- Archiviazione delle credenziali Amazon RDS in Gestione dei segreti AWS
- Eliminazione di un'istanza Amazon RDS esterna

Avvio e connessione a un'istanza Amazon RDS esterna in un VPC predefinito

Le seguenti procedure descrivono il processo di connessione a un'istanza Amazon RDS esterna a un VPC predefinito. Il processo è lo stesso di un VPC personalizzato. Gli unici requisiti aggiuntivi sono che l'ambiente e l'istanza di database si trovano nella stessa sottorete, o in sottoreti autorizzate a comunicare tra loro. Per ulteriori informazioni sulla configurazione di un VPC personalizzato da utilizzare con Elastic Beanstalk, consultare Utilizzo di Elastic Beanstalk con Amazon VPC.

> **Note**
>
> - Un'alternativa al lancio di una nuova istanza DB consiste nell'iniziare con un database precedentemente creato da Elastic Beanstalk e successivamente disaccoppiato da un ambiente Beanstalk. Per ulteriori informazioni, consulta the section called “Database”. Con questa opzione, non è necessario completare la procedura per avviare un nuovo database. Tuttavia, è necessario completare le procedure successive descritte in questo argomento.
> - Se si inizia con un database creato da Elastic Beanstalk e successivamente disaccoppiato da un ambiente Beanstalk, è possibile saltare il primo gruppo di passaggi e continuare con i passaggi raggruppati in Modificare le regole in entrata del gruppo di sicurezza dell'istanza RDS.
> - Se si prevede di utilizzare il database disaccoppiato per un ambiente di produzione, verificare che il tipo di archiviazione utilizzato dal database sia adatto al carico di lavoro. Per ulteriori informazioni, consultare Archiviazione dell'istanza database e Modifica di un'istanza database nella Guida per l'utente di Amazon RDS.

Per avviare un'istanza database RDS in un VPC predefinito

1. Aprire la console di RDS.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere Create database (Crea database).
4. Scegliere Standard Create (Creazione standard).

> **Important**
>
> Non selezionare Easy Create (Creazione rapida). Se viene selezionata questa opzione, non è possibile configurare le impostazioni necessarie per avviare questo database RDS.

5. In Additional configuration (Configurazione aggiuntiva), digitare **ebdb** per Initial database name (Nome database iniziale).
6. Controllare le impostazioni di default e regolare queste impostazioni in base alle proprie esigenze specifiche. Prestare particolare attenzione alle seguenti opzioni:
   - DB instance class (Classe di istanze database): scegliere una dimensione di istanza con capacità di memoria e potenza di CPU sufficienti per il tuo carico di lavoro.
   - Implementazione Multi-AZ: per un'elevata disponibilità, impostala su Crea un nodo Replica/Reader Aurora in una AZ diversa.
   - Master username (Nome utente master) e Master password (Password master): nome utente e password del database. Prendi nota di queste impostazioni perché ti serviranno in seguito.
7. Verificare le impostazioni predefinite per le opzioni rimanenti e scegliere Create database (Crea database).

Quindi, modifica il gruppo di sicurezza associato all'istanza database per consentire il traffico in entrata sulla porta appropriata. Questo è lo stesso gruppo di sicurezza che verrà collegato successivamente all'ambiente Elastic Beanstalk. Di conseguenza, la regola aggiunta concederà l'autorizzazione di accesso in entrata ad altre risorse nello stesso gruppo di sicurezza.

Per modificare le regole in entrata del gruppo di sicurezza collegato all'istanza RDS

1. Aprire la console Amazon RDS.
2. Scegli Databases (Database).
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.

4. Nella sezione Connectivity (Connettività), prendere nota dei valori relativi a Subnets (Sottoreti), Security groups (Gruppi di sicurezza) ed Endpoint mostrati in questa pagina. In questo modo è possibile utilizzare queste informazioni in un secondo momento.
5. In Security (Sicurezza) viene visualizzato il gruppo di sicurezza associato all'istanza database. Apri il link per visualizzare il gruppo di sicurezza nella EC2 console Amazon.
6. Nei dettagli del gruppo di sicurezza, scegliere Inbound (In entrata).
7. Scegli Modifica.
8. Selezionare Add Rule (Aggiungi regola).
9. In Type (Tipo), scegliere il motore di database utilizzato dall'applicazione.
10. In Source (Origine), digitare **`sg-`** per visualizzare un elenco dei gruppi di sicurezza disponibili. Scegliere il gruppo di sicurezza associato al gruppo Auto Scaling utilizzato con l'ambiente Elastic Beanstalk. In questo modo EC2 le istanze Amazon nell'ambiente possono avere accesso al database.

11. Seleziona Save (Salva).

Successivamente, aggiungi il gruppo di sicurezza per l'istanza database all'ambiente in esecuzione. In questa procedura Elastic Beanstalk rieffettua il provisioning di tutte le istanze dell'ambiente con il gruppo di sicurezza aggiuntivo collegato.

Per aggiungere un gruppo di sicurezza al tuo ambiente

- Esegui una delle seguenti operazioni:
  - Per aggiungere un gruppo di sicurezza tramite la console Elastic Beanstalk
    a. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS

b. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.

c. Nel pannello di navigazione, seleziona Configuration (Configurazione).

d. Nella categoria di configurazione Instances (Istanze), scegliere Edit (Modifica).

e. In Gruppi EC2 di sicurezza, scegli il gruppo di sicurezza da collegare alle istanze, oltre al gruppo di sicurezza dell'istanza creato da Elastic Beanstalk.

f. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

g. Leggere l'avviso, quindi selezionare Confirm (Conferma).

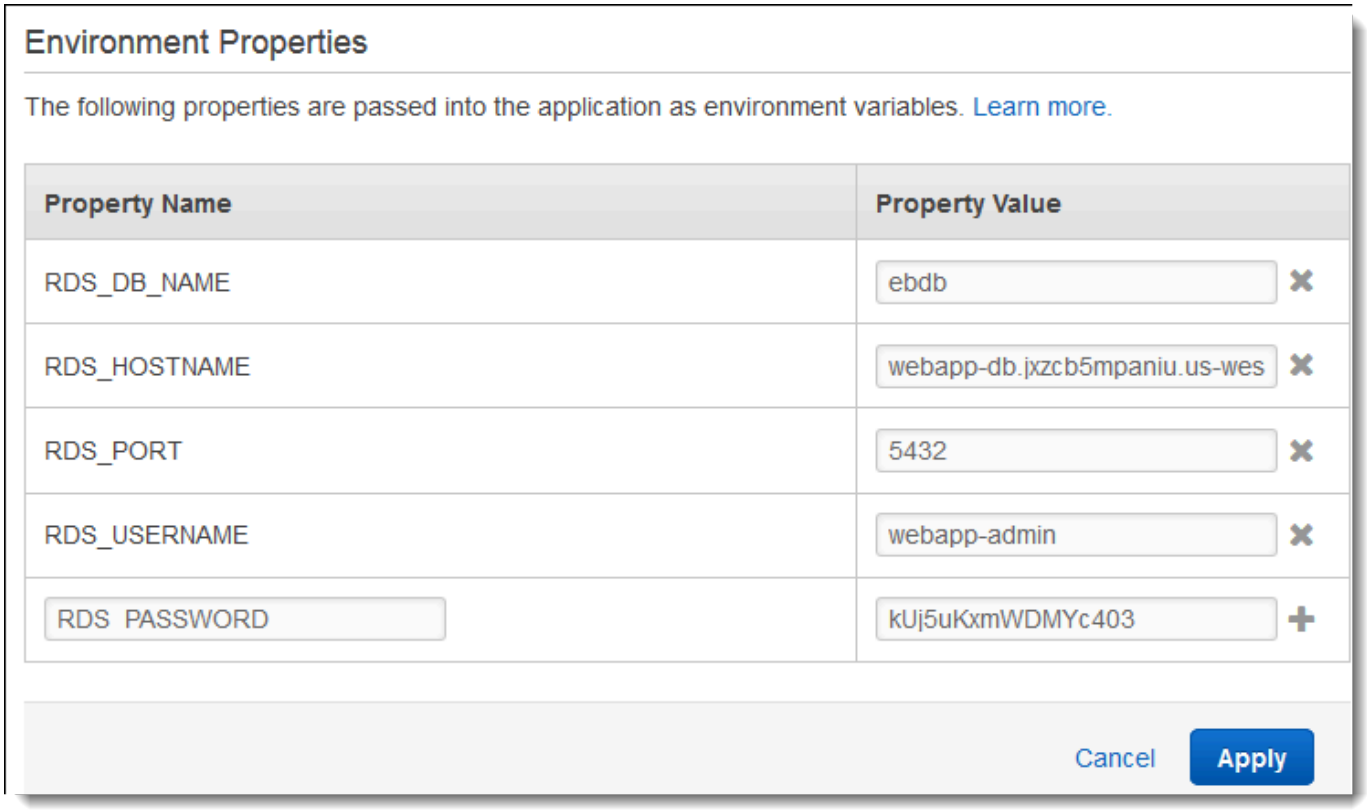
- Per aggiungere un gruppo di sicurezza tramite un file di configurazione, utilizzare il file di esempio `securitygroup-addexisting.config`.

Quindi, passa le informazioni di connessione al tuo ambiente utilizzando le proprietà dell'ambiente. Quando aggiungi un'istanza database al tuo ambiente con la console Elastic Beanstalk, Elastic Beanstalk utilizza proprietà dell'ambiente, come RDS_HOSTNAME, per passare le informazioni di connessione all'applicazione. È possibile utilizzare le stesse proprietà. È possibile utilizzare le stesse proprietà per usare lo stesso codice dell'applicazione con le istanze database integrate e le istanze database esterne. Oppure, in alternativa, è possibile scegliere i nomi delle proprie proprietà.

Per configurare le proprietà dell'ambiente per un'istanza database Amazon RDS

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Nella sezione Environment properties (Proprietà ambiente), definire le variabili che l'applicazione deve leggere per formare una stringa di collegamento. Per la compatibilità con gli ambienti che dispongono di un'istanza database RDS integrata, utilizza i seguenti nomi e valori. Tutti i valori, ad eccezione della password, sono disponibili nella console RDS.

Nome proprietà	Descrizione	Valore proprietà
`RDS_HOSTNAME`	Il nome host dell'istanza DB.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Endpoint.
`RDS_PORT`	La porta su cui l'istanza database accetta le connessioni. Il valore predefinito varia tra i motori di database.	Nella scheda Connectivity & security (Connettività e sicurezza) della console Amazon RDS: Port (Porta).
`RDS_DB_NAME`	Il nome del database, **`ebdb`**.	Nella scheda Configuration (Configurazione) della console Amazon RDS: DB Name (Nome DB).
`RDS_USERNAME`	Il nome utente configurato per il database.	Nella scheda Configuration (Configurazione) della console Amazon RDS: Master username (Nome utente master).
`RDS_PASSWORD`	La password configurata per il database.	Non disponibile per riferimento nella console Amazon RDS.



6. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Se non hai programmato la tua applicazione per leggere le proprietà dell'ambiente e creare una stringa di connessione, consultare i seguenti argomenti specifici per linguaggio per istruzioni:

- Java SE: Connessione a un database (piattaforme Java SE)
- Java con Tomcat: Connessione a un database (piattaforme Tomcat)
- Node.js: Connessione a un database
- .NET: Connessione a un database
- PHP: Connessione a un database con PDO o My SQLi
- Python: Connessione a un database
- Ruby: Connessione a un database

Infine, a seconda di quando l'applicazione legge le variabili di ambiente, potrebbe essere necessario riavviare il server delle applicazioni sulle istanze nell'ambiente.

Per riavviare i server delle app dell'ambiente

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Seleziona Actions (Operazioni), quindi Restart app server(s) (Riavvia server dell'applicazione).

Archiviazione delle credenziali Amazon RDS in Gestione dei segreti AWS

Questo argomento spiega come Gestione dei segreti AWS migliorare il livello di sicurezza per il recupero delle credenziali con Elastic Beanstalk. Fornisce inoltre riferimenti a risorse specifiche che possono aiutarti a configurare le credenziali per l'applicazione Elastic Beanstalk.

Gestione dei segreti AWS ti aiuta a migliorare il tuo livello di sicurezza, offrendo la possibilità di archiviare e recuperare credenziali crittografate. L'archiviazione delle credenziali in Secrets Manager aiuta a evitare possibili compromissioni da parte di chiunque possa esaminare la tua applicazione o componenti a essa correlati. Il codice può effettuare una chiamata runtime al servizio Secrets Manager per recuperare le credenziali in modo dinamico. Secrets Manager offre anche funzionalità come componenti di cache segreta lato client per i linguaggi di runtime, che includono Python, Go e Java.

Per ulteriori informazioni, consulta gli argomenti seguenti nella Guida per l'utente Gestione dei segreti AWS .

- Come Amazon RDS utilizza Gestione dei segreti AWS
- Crea un database segreto Gestione dei segreti AWS
- Recupera segreti da Gestione dei segreti AWS

Eliminazione di un'istanza Amazon RDS esterna

Quando connetti un'istanza Amazon RDS esterna all'ambiente Elastic Beanstalk, l'istanza database non dipende dal ciclo di vita dell'ambiente e, quindi, non viene eliminata quando termini l'ambiente. Per evitare che le informazioni personali che potresti aver archiviato nell'istanza database vengano conservate inutilmente, elimina tutti i record che non ti servono più. In alternativa, eliminare l'istanza database.

Utilizzo di Elastic Beanstalk con Amazon S3

Questo argomento spiega come Elastic Beanstalk utilizza Amazon Simple Storage Service (Amazon S3) e i tipi di oggetti che archivia nei bucket S3. Indica inoltre quali oggetti è necessario eliminare manualmente dopo aver chiuso l'ambiente Elastic Beanstalk e fornisce istruzioni per farlo.

Il bucket di account cliente Elastic Beanstalk Amazon S3

Elastic Beanstalk crea un `elasticbeanstalk-`*`region`*`-`*`account-id` bucket Amazon S3 crittografato denominato per ogni regione in cui crei gli ambienti. Il tuo account possiede AWS questo bucket. Elastic Beanstalk archivia i file di configurazione temporanei e altri oggetti per il corretto funzionamento dell'applicazione in questo bucket. Elastic Beanstalk ACLs richiede l'abilitazione per i bucket gestiti dal servizio e quindi abilita l'Access Control List (ACL) di questo bucket.

Tieni presente che Amazon S3 disabilita per impostazione predefinita le liste di controllo degli accessi dei bucket ()ACLs. Inoltre, l'argomento di panoramica ACL nella Guida per l'utente di Amazon S3 consiglia di ACLs mantenerlo disabilitato, ad eccezione di casi d'uso specifici. I bucket gestiti dal servizio Elastic Beanstalk rientrano in un caso d'uso che richiede l'attivazione. ACLs Per mantenere la sicurezza, le implementazioni di Elastic Beanstalk impongono che questo bucket sia di proprietà dell'account che esegue l'applicazione.

Elastic Beanstalk mantiene la crittografia predefinita fornita dai bucket Amazon S3. Per ulteriori informazioni sulla crittografia dei bucket, consulta la crittografia predefinita di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Contenuto del bucket di account cliente Elastic Beanstalk Amazon S3

La tabella seguente elenca alcuni oggetti che Elastic Beanstalk memorizza nel bucket dell'account cliente. La tabella indica inoltre che gli oggetti devono essere eliminati manualmente. Per evitare costi di storage superflui e la conservazione di informazioni personali, assicurati di eliminare manualmente questi oggetti quando non ti servono più.

Oggetto	Quando viene archiviato?	Quando viene eliminato?
Versioni delle applicazioni	Quando crei un ambiente o distribuisci il codice dell'applicazione in un ambiente esistente, Elastic Beanstalk archivia una	Durante l'eliminazione dell'appl icazione e in base a Ciclo di vita delle versioni.

Oggetto	Quando viene archiviato?	Quando viene eliminato?
	versione dell'applicazione in Amazon S3 e la associa all'ambiente.	
Bundle di origine	Quando carichi una nuova versione dell'applicazione utilizzando la console Elastic Beanstalk o CLI EB, Elastic Beanstalk ne archivia una copia in Amazon S3 e la imposta come bundle di origine dell'ambiente.	Manualmente. Quando elimini una versione dell'applicazione, puoi scegliere Delete versions from Amazon S3 (Elimina versioni da Amazon S3) per eliminare anche il bundle di origine correlato . Per informazioni dettagliate, vedi Gestione delle versioni delle applicazioni.
Piattaforme personalizzate	Quando crei una piattaforma personalizzata, Elastic Beanstalk archivia temporaneamente i dati correlati in Amazon S3.	Al termine della creazione della piattaforma personalizzata.
File di log	Puoi richiedere a Elastic Beanstalk di recuperare i file di log delle istanze (log di coda o bundle) e archiviarli in Amazon S3. Puoi anche abilitare la rotazione dei log e configurare l'ambiente per la pubblicazione automatica dei log in Amazon S3 dopo che vengono ruotati.	Log di coda e di bundle: 15 minuti dopo la loro creazione. Log ruotati: manualmente.
Configurazioni salvate	Manualmente.	Manualmente.

Eliminazione di oggetti nel bucket Amazon S3 di Elastic Beanstalk

Quando termini un ambiente o elimini un'applicazione, Elastic Beanstalk elimina la maggior parte degli oggetti correlati da Amazon S3. Per ridurre al minimo i costi di storage di un'applicazione in esecuzione, elimina regolarmente gli oggetti non più necessari per l'applicazione. Inoltre, presta attenzione agli oggetti che devi eliminare manualmente, come indicato in Contenuto del bucket di

account cliente Elastic Beanstalk Amazon S3. Per evitare che le informazioni personali vengano conservate inutilmente, elimina questi oggetti quando non ti servono più.

- Elimina le versioni dell'applicazione che non prevedi di usare più nell'applicazione. Quando elimini una versione dell'applicazione puoi selezionare Delete versions from Amazon S3 (Elimina le versioni da Amazon S3) per eliminare anche il bundle di origine correlato, ovvero una copia del codice sorgente e dei file di configurazione dell'applicazione caricati da Elastic Beanstalk in Amazon S3 durante la distribuzione di un'applicazione o il caricamento di una versione dell'applicazione. Per informazioni su come eliminare una versione dell'applicazione, consulta Gestione delle versioni delle applicazioni.
- Elimina i log ruotati che non ti servono. In alternativa, scaricali o spostali su Amazon Glacier per ulteriori analisi.
- Elimina le configurazioni salvate che non intendi usare più in alcun ambiente.

Eliminazione del bucket Amazon S3 di Elastic Beanstalk

Quando Elastic Beanstalk crea un bucket, crea anche una policy del bucket che applica al nuovo bucket. Questa policy ha due scopi:

- consentire agli ambienti di scrivere nel bucket;
- evitare la cancellazione accidentale del bucket.

A causa della policy che Elastic Beanstalk applica ai bucket che crea per gli ambienti, non è possibile eliminare questi bucket a meno che prima non venga intenzionalmente eliminata la policy dei bucket. È possibile eliminare la policy del bucket dalla sezione Autorizzazioni delle proprietà del bucket nella console Amazon S3.

> **Warning**
>
> Ti consigliamo di eliminare oggetti specifici non necessari dal bucket Elastic Beanstalk Amazon S3, anziché eliminare l'intero bucket.
> Se elimini un bucket creato da Elastic Beanstalk nel tuo account, ma continuano a essere presenti applicazioni e ambienti in esecuzione nella regione corrispondente, le applicazioni potrebbero smettere di funzionare correttamente. Esempio:

- Quando un ambiente è scalabile, Elastic Beanstalk dovrebbe essere in grado di trovare la versione dell'applicazione dell'ambiente nel bucket Amazon S3 e utilizzarla per avviare nuove istanze Amazon. EC2
- Quando crei una piattaforma personalizzata, Elastic Beanstalk utilizza temporaneamente lo storage Amazon S3 durante il processo di creazione.

Per ulteriori informazioni sulle implicazioni dell'eliminazione di un bucket S3, consulta le considerazioni elencate in Eliminazione di un bucket nella Amazon S3 User Guide.

Per eliminare un bucket di storage Elastic Beanstalk (console)

La procedura generale per eliminare un bucket S3 è descritta anche in Eliminazione di un bucket nella Amazon S3 User Guide. Poiché nella procedura seguente elimineremo un bucket creato da Elastic Beanstalk, includeremo passaggi aggiuntivi per eliminare prima la policy del bucket.

1. Apri la console Amazon S3.
2. Apri la pagina del bucket di storage Elastic Beanstalk scegliendo il nome del bucket.
3. Scegli la scheda Autorizzazioni.
4. Scegli Bucket Policy (Policy del bucket).
5. Scegli Elimina.
6. Torna alla pagina principale della console Amazon S3, quindi seleziona il bucket di archiviazione Elastic Beanstalk.
7. Scegli Delete Bucket (Elimina bucket).
8. Conferma che desideri eliminare il bucket immettendo il nome del bucket nel campo di testo, quindi scegli Elimina bucket.

Utilizzo di Elastic Gestione dei segreti AWS AWS Systems Manager Beanstalk con e Parameter Store

Questo argomento spiega come utilizzare Gestione dei segreti AWS e AWS Systems Manager Parameter Store con l'ambiente Elastic Beanstalk per archiviare e recuperare in modo sicuro informazioni sensibili, come credenziali e chiavi API. L'applicazione può recuperare i segreti e i

parametri memorizzati direttamente da questi archivi, utilizzando gli strumenti APIs o la riga di comando di questi servizi.

Elastic Beanstalk offre anche la possibilità di fare riferimento ai dati di Secrets Manager e Systems Manager Parameter Store nelle variabili di ambiente. Si tratta di un'opzione sicura per consentire all'applicazione di accedere in modo nativo ai segreti e ai parametri archiviati da questi servizi senza dover gestire le chiamate API verso di essi.

Argomenti

- Recupero di segreti e parametri nelle variabili di ambiente Elastic Beanstalk
- Autorizzazioni IAM richieste per Elastic Beanstalk per accedere a segreti e parametri
- Utilizzo Gestione dei segreti AWS e AWS Systems Manager Parameter Store
- Risoluzione dei problemi relativi all'integrazione dei segreti con le variabili di ambiente Elastic Beanstalk

Recupero di segreti e parametri nelle variabili di ambiente Elastic Beanstalk

Elastic Beanstalk può recuperare Gestione dei segreti AWS valori AWS Systems Manager da Parameter Store durante il bootstrap dell'istanza e assegnarli a variabili di ambiente utilizzabili dall'applicazione.

I seguenti punti riassumono la configurazione, la sincronizzazione e l'accesso per l'utilizzo delle variabili di ambiente come segreti:

- Configura le variabili di ambiente per archiviare i segreti specificando Amazon Resource Names (ARNs) per i segreti e i parametri che verranno archiviati.
- Quando i valori segreti vengono aggiornati o ruotati in Secrets Manager o Systems Manager Parameter Store, è necessario aggiornare manualmente le variabili di ambiente.
- Le variabili di ambiente secret sono disponibili per i comandi del contenitore ebextension e gli hook della piattaforma.

Versioni delle piattaforma supportate

Le versioni della piattaforma rilasciate a partire dal 26 marzo 2025 supportano Gestione dei segreti AWS i segreti e i parametri AWS Systems Manager Parameter Store configurati come variabili di ambiente.

> **Note**
>
> Ad eccezione delle piattaforme docker basate su Docker ed ECS, le versioni della piattaforma Amazon Linux 2 non supportano valori di variabili multilinea. Per ulteriori informazioni sul supporto per variabili multilinea, consulta. Valori multilinea

Argomenti

- Prezzi
- Configura i segreti come variabili di ambiente Elastic Beanstalk
- Estrazione delle chiavi JSON dai segreti di Secrets Manager
- Procedure consigliate per la sincronizzazione dei segreti con le variabili di ambiente Elastic Beanstalk
- Valori multilinea nelle variabili di ambiente Amazon Linux 2

Prezzi

Per l'utilizzo di Secrets Manager e Systems Manager Parameter Store si applicano le tariffe standard. Per ulteriori informazioni sui prezzi, consulta i seguenti siti Web:

- Gestione dei segreti AWS prezzi
- AWS Systems Manager prezzi (seleziona Parameter Store dall'elenco dei contenuti)

Elastic Beanstalk non addebita alcun costo per l'applicazione per fare riferimento ai segreti dell'ambiente tramite variabili di ambiente. Tuttavia, le tariffe standard si applicano alle richieste che Elastic Beanstalk invia a questi servizi per tuo conto.

Configura i segreti come variabili di ambiente Elastic Beanstalk

Puoi utilizzare la console Elastic Beanstalk, i file di configurazione `.ebextensions` in, AWS la CLI e l'SDK per configurare segreti AWS e parametri come variabili di ambiente.

Argomenti

- Prerequisiti
- Utilizzo della console

- Configurazione tramite file in .ebextensions
- Configurazione tramite AWS CLI
- Configurazione tramite l'SDK AWS

Prerequisiti

Prima di poter configurare le variabili di ambiente per fare riferimento ai segreti, devi prima completare i seguenti passaggi.

Procedura generale prima della configurazione delle variabili di ambiente

1. Crea i segreti di Secrets Manager o i parametri Parameter Store per archiviare i tuoi dati sensibili. Per ulteriori informazioni, consulta uno o entrambi i seguenti argomenti:
   - Creare segreti in the section called “Utilizzo di Secrets Manager”
   - Creazione di parametri in the section called “Utilizzo dell'archivio parametri di Systems Manager”
2. Configura le autorizzazioni IAM richieste per le istanze EC2 del tuo ambiente per recuperare i segreti e i parametri. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste.

Utilizzo della console

Puoi utilizzare la console Elastic Beanstalk per configurare i segreti come variabili di ambiente.

Per configurare i segreti come variabili di ambiente nella console Elastic Beanstalk

1. Apri la console Elastic Beanstalk e, nell'elenco Regioni, seleziona la tua. Regione AWS
2. Nel pannello di navigazione selezionare Environments (Ambienti), quindi selezionare il nome dell'ambiente dall'elenco.
3. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
4. Nella categoria di configurazione Updates, monitoring, and logging (Aggiornamenti, monitoraggio e registrazione), scegli Edit (Modifica).
5. Scorri verso il basso fino alle variabili di ambiente Runtime.
6. Seleziona Aggiungi variabile di ambiente.
7. Per Source, selezionare Secrets Manager o SSM Parameter Store.

> **Note**
>
> Per ulteriori informazioni sull'opzione Testo normale nel menu a discesa, consulta. Configurazione delle proprietà dell'ambiente (variabili di ambiente)

8. Per Nome della variabile di ambiente, immettete il nome della variabile di ambiente che contiene il valore segreto o del parametro.
9. Per Valore della variabile di ambiente, immettere l'ARN del parametro Systems Manager Parameter Store o il segreto Secrets Manager. Durante il bootstrap dell'istanza, Elastic Beanstalk avvierà il valore della variabile immessa nel passaggio 8 sul valore memorizzato in questa risorsa ARN.

   La console verifica se il valore immesso è un formato ARN valido per il negozio selezionato nel passaggio 7. Tuttavia, non convalida l'esistenza della risorsa specificata dall'ARN o se disponi delle autorizzazioni IAM necessarie per accedervi.
10. Se è necessario aggiungere altre variabili, ripetere i passaggi da 6 a 9.
11. Per salvare le modifiche scegli Apply (Applica) nella parte inferiore della pagina.

Configurazione tramite file in .ebextensions

Puoi utilizzare i file di configurazione di Elastic Beanstalk per configurare i segreti come variabili di ambiente. Usa lo aws: elasticbeanstalk:application:environmentsecrets spazio dei nomi per definire le proprietà dell'ambiente.

> **Note**
>
> Secrets Manager aggiunge automaticamente 6 caratteri casuali ai nomi segreti in formato ARN per garantire l'unicità.

Example.ebextensions/options.config per i segreti ambientali (sintassi abbreviata)

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    MY_SECRET: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-AbCd12
    MY_PARAMETER: arn:aws:ssm:us-east-1:111122223333:parameter/myparam
```

Example.ebextensions/options.config per i segreti ambientali (sintassi standard)

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environmentsecrets
    option_name: MY_SECRET
    value: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-AbCd12
  - namespace: aws:elasticbeanstalk:application:environmentsecrets
    option_name: MY_PARAMETER
    value: arn:aws:ssm:us-east-1:111122223333:parameter/myparam
```

Configurazione tramite AWS CLI

È possibile utilizzare AWS Command Line Interface (AWS CLI) per configurare i segreti come variabili di ambiente Elastic Beanstalk. Questa sezione fornisce esempi dei comandi create-environment e update-environment con lo spazio dei nomi. aws:elasticbeanstalk:application:environmentsecrets Quando Elastic Beanstalk avvia le istanze EC2 per gli ambienti a cui fanno riferimento questi comandi, inizializza le variabili di ambiente con il segreto recuperato e i valori dei parametri. Recupera questi valori dai rispettivi ARNs Secrets Manager e Systems Manager Parameter Store.

I due esempi seguenti utilizzano il comando create-environment per aggiungere un segreto e un parametro, configurati come variabili di ambiente denominate and. `MY_SECRET` `MY_PARAMETER`

Example di create-environment con segreti configurati come variabili di ambiente (opzioni dello spazio dei nomi in linea)

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--platform-arn "arn:aws:elasticbeanstalk:::platform/Node.js 24 running on 64bit Amazon
 Linux 2023" \
--option-settings \
Namespace=aws:autoscaling:launchconfiguration,OptionName=IamInstanceProfile,Value=aws-
elasticbeanstalk-ec2-role \
Namespace=aws:elasticbeanstalk:application:environmentsecrets,OptionName=MY_SECRET,Value=arn:aw
east-1:111122223333:secret:mysecret-AbCd12 \
Namespace=aws:elasticbeanstalk:application:environmentsecrets,OptionName=MY_PARAMETER,Value=arn
east-1:111122223333:parameter/myparam
```

In alternativa, utilizza un file `options.json` per specificare le opzioni dello spazio dei nomi invece di includerle in riga.

Example di create-environment con segreti configurati come variabili di ambiente (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk create-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--platform-arn "arn:aws:elasticbeanstalk:::platform/Node.js 24 running on 64bit Amazon
 Linux 2023" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": "aws-elasticbeanstalk-ec2-role"
  },
  {
    "Namespace": "aws:elasticbeanstalk:application:environmentsecrets",
    "OptionName": "MY_SECRET",
    "Value": "arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-AbCd12"
  },
  {
    "Namespace": "aws:elasticbeanstalk:application:environmentsecrets",
    "OptionName": "MY_PARAMETER",
    "Value": "arn:aws:ssm:us-east-1:111122223333:parameter/myparam"
  }
]
```

L'esempio successivo configura le variabili di ambiente, denominate `MY_SECRET` and`MY_PARAMETER`, per memorizzare un segreto e un parametro per un ambiente esistente. Il comando update-environment passa le opzioni con la stessa sintassi del `create-environment` comando, in linea o con un file. `options.json` L'esempio seguente dimostra che il comando utilizza lo stesso `options.json` file utilizzato anche nell'esempio precedente.

Example di update-environment con segreti configurati come variabili di ambiente (opzioni dello spazio dei nomi nel file) **`options.json`**

```
aws elasticbeanstalk update-environment \
--region us-east-1 \
--application-name my-app \
--environment-name my-env \
--platform-arn "arn:aws:elasticbeanstalk:::platform/Node.js 24 running on 64bit Amazon
 Linux 2023" \
--option-settings file://options.json
```

Configurazione tramite l'SDK AWS

È possibile configurare segreti e parametri come variabili di ambiente utilizzando. AWS SDKs Analogamente ai `create-environment` AWS CLI comandi `update-environment` and menzionati nella sezione precedente, è possibile utilizzare le azioni CreateEnvironmente UpdateEnvironmentAPI. Utilizzate il parametro `OptionSettings` request per specificare le opzioni del aws: elasticbeanstalk:application:environmentsecrets namespace.

Estrazione delle chiavi JSON dai segreti di Secrets Manager

Le versioni della piattaforma rilasciate a partire dal 13 gennaio 2026 supportano l'estrazione di campi specifici dai segreti di Secrets Manager in formato JSON aggiungendo due punti e il nome della chiave JSON all'ARN segreto. Ciò consente di fare riferimento a singole coppie chiave-valore all'interno di un segreto anziché all'intero segreto.

Sintassi

Per estrarre una chiave JSON specifica da un segreto, aggiungi `:json-key-name` all'ARN segreto:

```
arn:aws:secretsmanager:region:account:secret:secret-name-XXXXXX:json-key-name
```

L'estrazione delle chiavi JSON può essere configurata utilizzando gli stessi metodi dei normali segreti di ambiente: console, file di configurazione in `.ebextensions`, AWS CLI o. AWS SDKs

Limitazioni

- L'estrazione delle chiavi JSON è supportata solo per i segreti di Secrets Manager, non per i parametri di Systems Manager Parameter Store.

- Sono supportate solo le chiavi JSON di primo livello. L'accesso con chiave annidata (ad esempio**`config.database.host`**) e l'indicizzazione degli array (ad esempio) non sono supportati. **`servers[0]`** Se avete bisogno di accedere a valori annidati, estraete l'oggetto principale e analizzatelo nel codice dell'applicazione. Gli oggetti e gli array annidati a cui si accede utilizzando la chiave di primo livello vengono serializzati nuovamente in formato JSON.
- I nomi delle chiavi JSON non possono contenere caratteri con i due punti (). **`:`** I due punti sono riservati come delimitatori nella sintassi ARN. Se il nome di una chiave JSON contiene due punti, solo la parte che precede i primi due punti viene utilizzata come nome della chiave e tutti i caratteri dopo i due punti vengono ignorati.
- La piattaforma Docker gestita da ECS utilizza la sintassi ECS nativa per fare riferimento ai segreti. Per ulteriori informazioni, consulta Passare i segreti di Secrets Manager tramite le variabili di ambiente Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Esempio

L'esempio seguente mostra come vengono estratti diversi tipi di valore da un segreto JSON. Dato questo segreto:

```
{
  "stringKey": "value1",
  "numberKey": 42,
  "objectKey": {
    "nested": "data"
  },
  "arrayKey": ["item1", "item2"]
}
```

Puoi configurare l'estrazione in`.ebextensions`:

Example esempio.ebextensions con estrazione di chiavi JSON

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    STRING_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:stringKey
    NUMBER_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:numberKey
    OBJECT_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:objectKey
```

```
    ARRAY_VAR: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret-
AbCd12:arrayKey
```

Ciò si traduce nelle seguenti variabili di ambiente:

- `STRING_VAR=value1`
- `NUMBER_VAR=42`
- `OBJECT_VAR={"nested":"data"}`
- `ARRAY_VAR=["item1","item2"]`

Procedure consigliate per la sincronizzazione dei segreti con le variabili di ambiente Elastic Beanstalk

Questo argomento consiglia le migliori pratiche per l'applicazione per utilizzare i segreti dell'ambiente con Secrets Manager o Systems Manager Parameter Store. L'applicazione Elastic Beanstalk non riceverà automaticamente i valori aggiornati se i dati dell'archivio segreto vengono aggiornati o ruotati. Elastic Beanstalk inserisce i segreti nelle variabili di ambiente solo al momento del bootstrap dell'istanza.

Aggiornamento delle variabili di ambiente

Per attivare l'ambiente Elastic Beanstalk in modo che recuperi i valori più recenti dei segreti dai relativi archivi segreti, ti consigliamo di eseguire l'operazione or. `UpdateEnvironment` `RestartAppServer` Puoi eseguire queste operazioni utilizzando la console Elastic Beanstalk AWS CLI, o l'API Elastic Beanstalk. Per ulteriori informazioni, consulta AWS CLI gli esempi di Elastic Beanstalk o l'API Reference.AWS Elastic Beanstalk

Gestione degli effetti di ridimensionamento automatico sulla sincronizzazione segreta

Se si verifica un evento di scalabilità out o la sostituzione dell'istanza dopo l'aggiornamento dell'archivio segreto, la nuova istanza che verrà visualizzata conterrà i valori segreti più recenti di Secrets Manager o Systems Manager Parameter Store. Un evento di questo tipo può verificarsi anche se non tutte le altre istanze dell'ambiente sono state aggiornate per recuperare i nuovi segreti.

> **Important**
>
> È necessario assicurarsi che l'applicazione sia in grado di utilizzare due valori segreti diversi per la stessa variabile di ambiente. Ciò consente di gestire gli eventi in cui si verifica un

aggiornamento segreto in Secrets Manager o Systems Manager Parameter Store, seguito da una scalabilità orizzontale o dalla sostituzione dell'istanza nell'ambiente, mentre le altre istanze sono in attesa di aggiornamento della variabile di ambiente. Durante il periodo di attesa per l'aggiornamento, non tutte le istanze di ambiente avranno gli stessi valori per le variabili di ambiente dell'archivio segreto.

Un esempio di tale caso d'uso è la rotazione delle credenziali del database. Quando un evento di scalabilità out segue la rotazione delle credenziali, i segreti di ambiente a cui fanno riferimento le nuove istanze avviate contengono le credenziali del database aggiornate. Tuttavia, i segreti di ambiente a cui fanno riferimento le istanze esistenti mantengono il vecchio valore fino a quando non vengono aggiornati dalle operazioni o. `UpdateEnvironment` `RestartAppServer`

Valori multilinea nelle variabili di ambiente Amazon Linux 2

I valori su più righe sono composti da più di una riga e includono un carattere di nuova riga. Ad eccezione delle piattaforme Docker e Docker basate su ECS, le piattaforme eseguite su Amazon Linux 2 non supportano valori multilinea per le variabili di ambiente

> **Note**
>
> Elastic Beanstalk fallirà la distribuzione degli ambienti interessati se rileva un valore multilinea.

Le seguenti opzioni possono servire come soluzioni alternative o soluzioni al problema multilinea:

- Aggiorna il tuo ambiente Amazon Linux 2 ad Amazon Linux 2023. Per ulteriori informazioni, consulta Migrazione da Amazon Linux 2 ad Amazon Linux 2023.
- Rimuovi i caratteri di nuova riga dai tuoi valori segreti. Un esempio di approccio consiste nel codificare in Base64 i valori prima di archiviarli nell'archivio segreto. L'applicazione dovrebbe quindi decodificare nuovamente il valore nel formato originale quando vi fa riferimento dalla variabile segreta di ambiente.
- Progetta il codice dell'applicazione per recuperare i dati direttamente da Secrets Manager o Systems Manager Parameter Store. Per ulteriori informazioni, vedere Recupero dei segreti in Utilizzo di Secrets Manager o Recupero dei parametri. Utilizzo dell'archivio parametri di Systems Manager

Autorizzazioni IAM richieste per Elastic Beanstalk per accedere a segreti e parametri

È necessario concedere le autorizzazioni necessarie alle istanze EC2 del proprio ambiente per recuperare i segreti e i parametri di Parameter Store. Gestione dei segreti AWS AWS Systems Manager Le autorizzazioni vengono fornite alle istanze EC2 tramite un ruolo di profilo di istanza EC2.

Le sezioni seguenti elencano le autorizzazioni specifiche da aggiungere a un profilo di istanza EC2, a seconda del servizio utilizzato. Segui i passaggi forniti in Aggiornamento della politica di autorizzazione per un ruolo nella Guida per l'utente IAM per aggiungere queste autorizzazioni.

> **Autorizzazioni IAM per la piattaforma Docker gestita da ECS**
>
> La piattaforma Docker gestita da ECS richiede autorizzazioni IAM aggiuntive rispetto a quelle fornite in questo argomento. Per ulteriori informazioni su tutte le autorizzazioni necessarie per l'ambiente della piattaforma Docker gestito ECS per supportare l'integrazione delle variabili di ambiente Elastic Beanstalk con i segreti, consulta. Formato ARN del ruolo di esecuzione

Argomenti

- Autorizzazioni IAM richieste per Secrets Manager
- Autorizzazioni IAM richieste Systems Manager Parameter Store

Autorizzazioni IAM richieste per Secrets Manager

Le seguenti autorizzazioni garantiscono l'accesso per recuperare i segreti crittografati dall'archivio: Gestione dei segreti AWS

- gestore dei segreti: GetSecretValue
- kms:Decrypt

L'autorizzazione a decrittografare un file AWS KMS key è richiesta solo se il segreto utilizza una chiave gestita dal cliente anziché la chiave predefinita. L'aggiunta della chiave personalizzata ARN aggiunge l'autorizzazione a decrittografare la chiave gestita dal cliente.

Example policy con autorizzazioni per Secrets Manager e chiavi KMS

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:my-secret",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Autorizzazioni IAM richieste Systems Manager Parameter Store

Le seguenti autorizzazioni concedono l'accesso per recuperare i parametri crittografati dal Parameter Store: AWS Systems Manager

- ssm: GetParameter
- kms:Decrypt

L'autorizzazione a decrittografare un AWS KMS key è richiesta solo per i tipi di `SecureString` parametri che utilizzano una chiave gestita dal cliente anziché una chiave predefinita. L'aggiunta della chiave personalizzata ARN aggiunge l'autorizzazione a decrittografare la chiave gestita dal cliente. I normali tipi di parametri che non sono crittografati `String` e non `StringList` richiedono un. AWS KMS key

Example policy con Systems Manager e autorizzazioni AWS KMS chiave

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-1:111122223333:parameter/my-parameter",
                "arn:aws:kms:us-east-1:111122223333:key/my-key"
            ]
        }
    ]
}
```

Utilizzo Gestione dei segreti AWS e AWS Systems Manager Parameter Store

Questo argomento fornisce una breve introduzione a AWS Systems Manager Parameter Store, informazioni sui prezzi e riferimenti per ulteriori informazioni sulla creazione e il recupero dei segreti, utilizzando sia la console che le opzioni programmatiche. Gestione dei segreti AWS

Informazioni su Secrets Manager

Gestione dei segreti AWS ti aiuta a gestire, recuperare e far ruotare i segreti durante il loro ciclo di vita. Esempi di dati segreti che puoi gestire includono le credenziali del database, le credenziali delle applicazioni, i token e le chiavi API. OAuth Secrets Manager ti consente di configurare una pianificazione di rotazione automatica per i tuoi segreti.

Informazioni su Systems Manager Parameter Store

Parameter Store è uno strumento in AWS Systems Manager. Fornisce uno storage sicuro e gerarchico per la gestione dei dati di configurazione e la gestione dei segreti. È possibile gestire importanti dati di configurazione come valori di parametro. Esempi di dati che puoi gestire con

Parameter Store includono Amazon Machine Image (AMI) IDs, codici di licenza, password e stringhe di database.

> **Prezzi**
>
> Per l'utilizzo di Secrets Manager e Systems Manager Parameter Store si applicano le tariffe standard. Per ulteriori informazioni sui prezzi, consulta i seguenti siti Web:
>
> - Gestione dei segreti AWS prezzi
> - AWS Systems Manager prezzi (seleziona Parameter Store dall'elenco dei contenuti)

Argomenti

- Utilizzo di Secrets Manager per creare e recuperare segreti
- Utilizzo di Systems Manager Parameter Store per creare e recuperare parametri

Utilizzo di Secrets Manager per creare e recuperare segreti

È possibile creare e recuperare i segreti di Secrets Manager utilizzando la Gestione dei segreti AWS console AWS CLI, l'SDK o l' AWS SDK. Consulta le seguenti risorse per saperne di più sui diversi metodi per creare e recuperare i segreti di Secrets Manager.

Creare segreti

- Console: crea un Gestione dei segreti AWS segreto (console) nella Guida Gestione dei segreti AWS per l'utente
- AWS CLI — AWS CLI nella guida per l'utente Gestione dei segreti AWS
- AWS SDK: AWS SDK nella guida per l'utente Gestione dei segreti AWS

Recupero di segreti

- Console: ottieni un valore segreto (console) nella Guida per l'utente Gestione dei segreti AWS
- AWS CLI: ottieni un valore segreto (AWS CLI) nella Guida per l'utente Gestione dei segreti AWS
- AWS SDK: esempi di codice per Secrets Manager utilizzati AWS SDKs nella libreria di codici AWS SDK Code Examples
- Altri metodi: scopri i segreti Gestione dei segreti AWS nella Guida per l'utente Gestione dei segreti AWS

Per ulteriori informazioni su Gestione dei segreti AWS, consulta What is Gestione dei segreti AWS? nella Guida Gestione dei segreti AWS per l'utente.

Utilizzo di Systems Manager Parameter Store per creare e recuperare parametri

È possibile creare e recuperare i parametri di Parameter Store utilizzando la AWS Systems Manager console, l'SDK o l' AWS CLI AWS SDK. Fate riferimento alle seguenti risorse per saperne di più sui diversi metodi per creare e recuperare i parametri del Parameter Store.

Creazione di parametri

- Console: crea un parametro Systems Manager (console) nella Guida AWS Systems Manager per l'utente
- AWS CLI — Creazione di un parametro Systems Manager (AWS CLI) nella Guida per l'utente AWS Systems Manager
- AWS SDK: da utilizzare PutParameter con un AWS SDK o AWS CLI nella libreria di codici SDK Code AWS Examples

Recupero dei parametri

- Console: ricerca di un parametro (console) nella Guida per l'utente AWS Systems Manager
- AWS CLI: utilizzo GetParameter con un AWS SDK o AWS CLI nella libreria di codici SDK Code AWS Examples
- AWS SDK: da utilizzare GetParameter con un AWS SDK o AWS CLI nella libreria di codici SDK Code Examples AWS

Per ulteriori informazioni, consulta Archivio dei parametri AWS Systems Manager nella Guida per l'utente di AWS Systems Manager .

Risoluzione dei problemi relativi all'integrazione dei segreti con le variabili di ambiente Elastic Beanstalk

> **Prova Amazon Q Developer CLI per la risoluzione dei problemi assistita dall'intelligenza artificiale**
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli

eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Risoluzione dei problemi degli ambienti Elastic Beanstalk con Amazon Q Developer CLI nei blog. AWS

Questa sezione fornisce indicazioni per la risoluzione dei problemi relativi ai segreti nell'ambiente Elastic Beanstalk.

Evento: la distribuzione dell'istanza non è riuscita a ottenere uno o più segreti

Questo messaggio indica che Elastic Beanstalk non è stato in grado di recuperare uno o più segreti specificati durante la distribuzione dell'applicazione.

- Verifica che le risorse specificate dai valori ARN nella configurazione della variabile di ambiente esistano.
- Verifica che il ruolo del profilo dell'istanza Elastic Beanstalk EC2 disponga delle autorizzazioni IAM necessarie per accedere alle risorse.
- Se questo evento è stato attivato dall'`RestartAppServer`operazione, una volta risolto il problema, riprova a effettuare la chiamata per risolverlo. `RestartAppServer`
- Se l'evento è stato attivato tramite una `UpdateEnvironment` chiamata, riprova l'operazione. `UpdateEnvironment`

Per esempi di questi comandi, vedi AWS CLI esempi per Elastic Beanstalk. Per ulteriori informazioni sulle azioni API per queste operazioni, consulta l'AWS Elastic Beanstalk API Reference.

Evento: la distribuzione dell'istanza ha rilevato uno o più valori di ambiente multilinea, che non sono supportati per questa piattaforma

Le variabili multilinea non sono supportate per le piattaforme Amazon Linux 2, escluse le piattaforme Docker e Docker gestite da ECS. Per le opzioni disponibili per procedere, consulta. Valori multilinea

Evento: CreateEnvironment fallisce quando viene specificato un segreto

In caso di `CreateEnvironment` errore e si dispone di segreti come variabili di ambiente, è necessario risolvere il problema sottostante e quindi `UpdateEnvironment` utilizzarlo per completare la configurazione dell'ambiente. Non utilizzarlo`RestartAppServer`, poiché non sarà sufficiente a far apparire l'ambiente in questa situazione. Per esempi di questi comandi, vedi AWS CLI esempi per

Elastic Beanstalk. Per ulteriori informazioni sulle azioni API per queste operazioni, consulta l'AWS Elastic Beanstalk API Reference.

Utilizzo di Elastic Beanstalk con Amazon VPC

Questo argomento spiega i vantaggi dell'utilizzo degli endpoint VPC con Elastic Beanstalk e i diversi tipi di configurazioni che puoi implementare.

Puoi utilizzare un Amazon Virtual Private Cloud (Amazon VPC) per creare una rete sicura per l'applicazione Elastic Beanstalk e le relative risorse. AWS Quando crei l'ambiente, scegli VPC, sottoreti e gruppi di sicurezza usati per le istanze dell'applicazione e il sistema di bilanciamento del carico. Puoi usare qualsiasi configurazione VPC desiderata, a condizione che soddisfi i requisiti seguenti.

Requisiti VPC

- Accesso a Internet: le istanze possono accedere a Internet tramite uno dei metodi seguenti.
  - Sottorete pubblica: le istanze dispongono di un indirizzo IP pubblico e utilizzano un gateway Internet per accedere a Internet.
  - Sottorete privata: le istanze utilizzano un dispositivo NAT per accedere a Internet.

  > **Note**
  >
  > Se si configurano endpoint VPC nel VPC per connettersi sia ai servizi `elasticbeanstalk` sia ai servizi `elasticbeanstalk-health`, l'accesso a Internet è facoltativo ed è richiesto solo se necessario per l'applicazione. Senza endpoint VPC, il VPC deve avere accesso a Internet.
  > Il VPC predefinito impostato automaticamente da Elastic Beanstalk fornisce accesso a Internet.

  Elastic Beanstalk non supporta le impostazioni proxy, ad esempio `HTTPS_PROXY`, per configurare un proxy Web.
- NTP: le istanze nell'ambiente Elastic Beanstalk utilizzano NTP (Network Time Protocol) per sincronizzare l'orologio di sistema. Se le istanze non sono in grado di comunicare sulla porta UDP 123, l'orologio potrebbe non essere sincronizzato, causando problemi con il reporting dello stato di Elastic Beanstalk. Assicurati che i gruppi di sicurezza e la rete VPC ACLs consentano il traffico UDP in entrata e in uscita sulla porta 123 per evitare questi problemi.

Il elastic-beanstalk-samplesrepository fornisce CloudFormation modelli che puoi utilizzare per creare un VPC da utilizzare con i tuoi ambienti Elastic Beanstalk.

Per creare risorse con un modello CloudFormation

1. Clonare i repository di esempio o scaricare un modello tramite i collegamenti disponibili in README.
2. Apri la CloudFormation console.
3. Seleziona Crea stack.
4. Selezionare Upload a template to Amazon S3 (Carica un modello in Amazon S3).
5. Selezionare Upload file (Carica file), quindi caricare il file modello dalla macchina locale.
6. Scegliere Next (Successivo) e seguire le istruzioni per creare uno stack con le risorse all'interno del modello.

Una volta completata la creazione dello stack, seleziona la scheda Output per trovare l'ID VPC e la sottorete. IDs Usa queste informazioni per configurare il VPC nella categoria di configurazione della rete nella procedura guidata del nuovo ambiente.

Argomenti

- VPC pubblico
- VPC pubblico/privato
- VPC privato
- Esempio: avvio di un'applicazione Elastic Beanstalk in un VPC con host bastione
- Esempio: avvio di un Elastic Beanstalk in un VPC con Amazon RDS
- Utilizzo di Elastic Beanstalk con endpoint VPC
- Utilizzo dei criteri endpoint per controllare l'accesso con gli endpoint VPC

VPC pubblico

CloudFormation modello — vpc-public.yaml

Impostazioni (carico bilanciato)

- Visibilità del sistema di bilanciamento del carico: pubblica
- Sottoreti del sistema di bilanciamento del carico: entrambe le sottoreti pubbliche

- IP pubblico istanza: abilitato
- Sottoreti dell'istanza: entrambe le sottoreti pubbliche
- Gruppi di sicurezza dell'istanza: aggiungi il gruppo di sicurezza predefinito

Impostazioni (istanza singola)

- Sottoreti dell'istanza: una delle sottoreti pubbliche
- Gruppi di sicurezza dell'istanza: aggiungi il gruppo di sicurezza predefinito

Un layout VPC solo pubblico include una o più sottoreti pubbliche, un gateway Internet e un gruppo di sicurezza predefinito che permette il traffico tra le risorse nel VPC. Quando crei un ambiente nel VPC, Elastic Beanstalk crea risorse aggiuntive che variano a seconda del tipo di ambiente.

Risorse VPC

- Istanza singola: Elastic Beanstalk crea un gruppo di sicurezza per l'istanza dell'applicazione che permette il traffico sulla porta 80 da Internet e assegna all'istanza un IP elastico per fornire un indirizzo IP pubblico. Il nome di dominio dell'ambiente si risolve nell'indirizzo IP pubblico dell'istanza.
- Carico bilanciato: Elastic Beanstalk crea un gruppo di sicurezza per il sistema di bilanciamento del carico che permette il traffico sulla porta 80 da Internet e un gruppo di sicurezza per le istanze dell'applicazione che permette il traffico dal gruppo di sicurezza del sistema di bilanciamento del carico. Il nome di dominio dell'ambiente si risolve nel nome di dominio pubblico del sistema di bilanciamento del carico.

Ciò è simile al modo in cui Elastic Beanstalk gestisce le reti quando si usa il VPC predefinito. La sicurezza in una sottorete pubblica dipende dal sistema di bilanciamento del carico e dai gruppi di sicurezza delle istanze creati da Elastic Beanstalk. Si tratta della configurazione meno costosa e non richiede un gateway NAT.

VPC pubblico/privato

CloudFormation modello — vpc-privatepublic.yaml

Impostazioni (carico bilanciato)

- Visibilità del sistema di bilanciamento del carico: pubblica

- Sottoreti del sistema di bilanciamento del carico: entrambe le sottoreti pubbliche
- IP pubblico dell'istanza: disabilitato
- Sottoreti dell'istanza: entrambe le sottoreti private
- Gruppi di sicurezza dell'istanza: aggiungi il gruppo di sicurezza predefinito

Per maggiore sicurezza, aggiungi sottoreti private al VPC per creare un layout pubblico-privato. Questo layout richiede un sistema di bilanciamento del carico e il gateway NAT nelle sottoreti pubbliche e permette di eseguire le istanze dell'applicazione, il database e tutte le altre risorse nelle sottoreti private. Le istanze nelle sottoreti private possono comunicare con Internet solo tramite il sistema di bilanciamento del carico e il gateway NAT.

VPC privato

CloudFormation modello — vpc-private.yaml

Impostazioni (carico bilanciato)

- Visibilità del sistema di bilanciamento del carico: privata
- Sottoreti del sistema di bilanciamento del carico: entrambe le sottoreti sono private
- IP pubblico dell'istanza: disabilitato
- Sottoreti dell'istanza: entrambe le sottoreti private
- Gruppi di sicurezza dell'istanza: aggiungi il gruppo di sicurezza predefinito

Per le applicazioni interne che non devono essere accessibili da Internet, è possibile impostare l'esecuzione nelle sottoreti private e configurare il sistema di bilanciamento del carico come interno, modificando l'impostazione di Load balancer visibility (Visibilità sistema di bilanciamento del carico) su Internal (Interna). Questo modello crea un VPC senza sottoreti pubbliche e senza gateway Internet. Usa questo layout per le applicazioni che devono essere accessibili solo dallo stesso VPC o da una VPN collegata.

Esecuzione di un ambiente Elastic Beanstalk in un VPC privato

L'ambiente Elastic Beanstalk creato in un VPC privato non dispone di accesso a Internet. L'applicazione potrebbe richiedere l'accesso al servizio Elastic Beanstalk o ad altri servizi. L'ambiente potrebbe utilizzare il reporting dello stato avanzato e, in questo caso, le istanze dell'ambiente inviano informazioni sullo stato al servizio di stato avanzato. Inoltre, il codice Elastic Beanstalk sulle istanze di ambiente invia traffico AWS ad altri servizi e altro traffico a punti diversi dagli endpoint (ad

esempio, per scaricare pacchetti di dipendenza per l'applicazione).AWS Di seguito sono riportati alcuni passaggi che potrebbe essere necessario eseguire in questo caso per garantire che l'ambiente funzioni correttamente.

- Configurare gli endpoint VPC per Elastic Beanstalk: Elastic Beanstalk e il relativo servizio di stato avanzato supportano gli endpoint VPC. Ciò garantisce che il traffico verso questi servizi rimanga all'interno della rete Amazon e non richieda l'accesso a Internet. Per ulteriori informazioni, consulta the section called “Endpoint VPC”.
- Configura gli endpoint VPC per servizi aggiuntivi: le istanze Elastic Beanstalk inviano traffico a diversi altri servizi AWS per tuo conto: Amazon Simple Storage Service (Amazon S3), Amazon Simple Queue Service (Amazon SQS) e Amazon Logs. AWS CloudFormation CloudWatch È necessario configurare gli endpoint VPC anche per questi servizi. Per informazioni dettagliate sugli endpoint VPC, inclusi i collegamenti per servizio, consulta Endpoint VPC nella Guida per l'utente di Amazon VPC.

> **Note**
>
> Alcuni AWS servizi, tra cui Elastic Beanstalk, supportano gli endpoint VPC in un numero limitato di regioni. AWS Quando progetti la tua soluzione VPC privata, verifica che Elastic Beanstalk e gli altri servizi dipendenti menzionati qui supportino gli endpoint VPC nella regione scelta. AWS

- Fornire un'immagine Docker privata: in un ambiente Docker, è possibile che il codice nelle istanze dell'ambiente tenti, senza riuscirci, di estrarre l'immagine Docker configurata da Internet durante la creazione dell'ambiente. Per evitare questo errore, crea un'immagine Docker personalizzata nel tuo ambiente o utilizza un'immagine Docker archiviata in Amazon Elastic Container Registry (Amazon ECR) e configura un endpoint VPC per il servizio Amazon ECR.
- Abilita i nomi DNS: il codice Elastic Beanstalk sulle istanze di ambiente invia il traffico AWS a tutti i servizi utilizzando i relativi endpoint pubblici. Per garantire il passaggio del traffico, scegliere l'opzione Enable DNS name (Abilita nome DNS) quando si configurano tutti gli endpoint VPC dell'interfaccia. In questo modo viene aggiunta una voce DNS nel VPC che associa l'endpoint del servizio pubblico all'endpoint VPC dell'interfaccia.

> **Important**
>
> Se il VPC non è privato e dispone di accesso a Internet pubblico e se Enable DNS name (Abilita nome DNS) è disabilitato per qualsiasi endpoint VPC, il traffico verso il rispettivo

servizio passa attraverso Internet pubblico. Probabilmente non quanto si desiderava ottenere. È facile rilevare questo problema con un VPC privato, perché impedisce il passaggio del traffico e si ricevono errori. Tuttavia, con un VPC pubblico, non si ottiene alcuna indicazione.

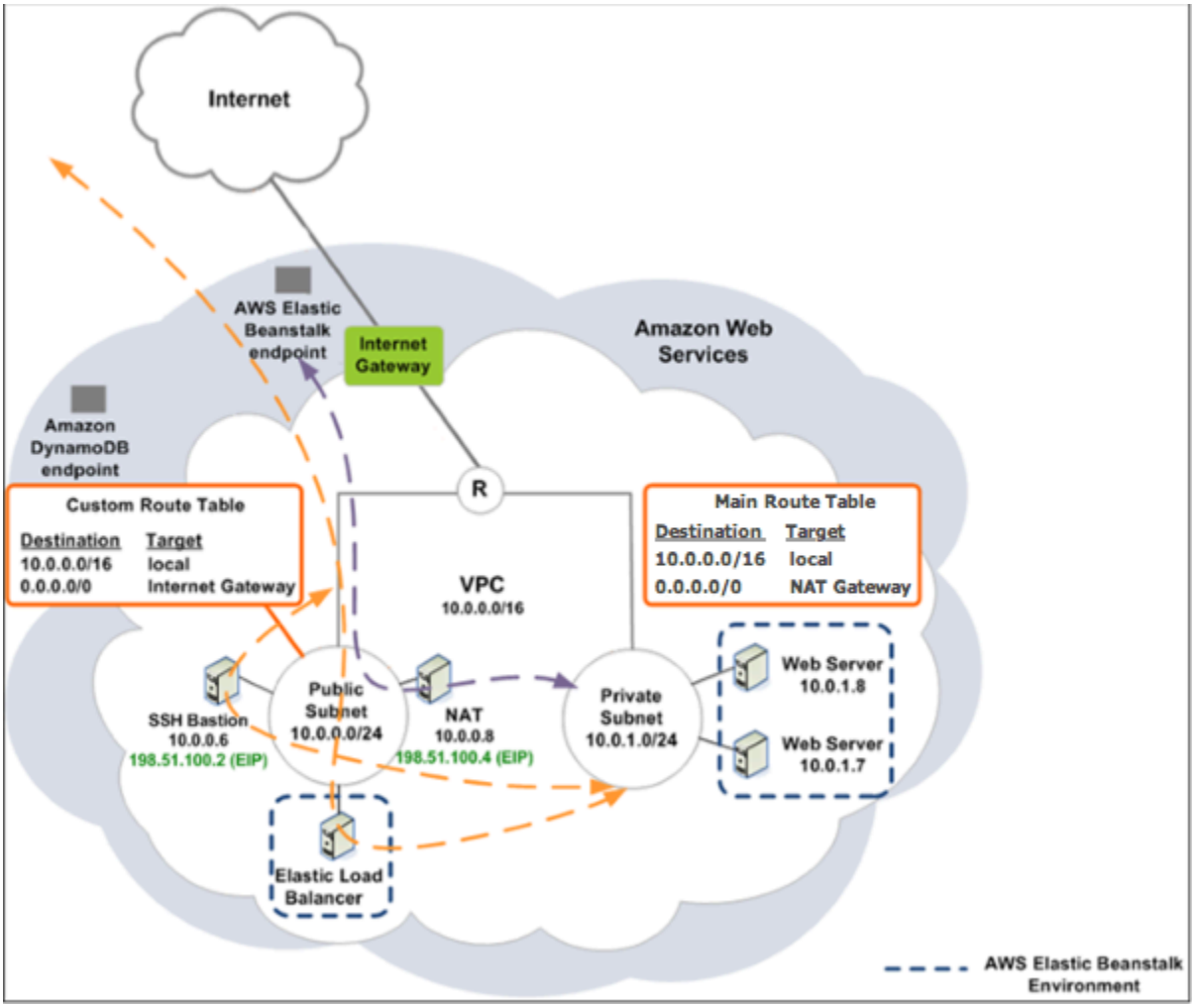
- Includere dipendenze dell'applicazione: se l'applicazione dispone di dipendenze, ad esempio pacchetti di runtime del linguaggio, potrebbe tentare, senza riuscirci, di scaricarle da Internet e installarle durante la creazione dell'ambiente. Per evitare questo errore, includere tutti i pacchetti di dipendenza nel bundle di origine dell'applicazione.
- Utilizzare una versione corrente della piattaforma: assicurarsi che l'ambiente utilizzi una versione della piattaforma rilasciata il 24 febbraio 2020 o in seguito. In particolare, utilizzare una versione della piattaforma rilasciata a partire da uno di questi due aggiornamenti: Linux Update 2020-02-28, Windows Update 2020-02-24.

> **Note**
>
> Una versione aggiornata della piattaforma è necessaria a causa di un problema delle versioni precedenti che impedisce alle voci DNS create dall'opzione Enable DNS name (Abilita nome DNS) di funzionare correttamente per Amazon SQS.

Esempio: avvio di un'applicazione Elastic Beanstalk in un VPC con host bastione

Questa sezione spiega come implementare un'applicazione Elastic Beanstalk all'interno di un VPC utilizzando un host bastion e perché implementare questa topologia.

Se le tue EC2 istanze Amazon si trovano all'interno di una sottorete privata, non potrai connetterti ad esse da remoto. Per connetterti alle tue istanze, puoi configurare gli host bastione nella sottorete pubblica in modo che fungano da proxy. Ad esempio, puoi configurare le unità di inoltro alla porta SSH o i gateway RDP nella sottorete pubblica per fungere da proxy per il traffico verso i server del database dalla tua rete. Questa sezione fornisce un esempio di come creare un VPC con una sottorete privata e pubblica. Le istanze si trovano all'interno della sottorete privata, mentre l'host bastione, il gateway NAT e il sistema di bilanciamento del carico si trovano all'interno della sottorete pubblica. L'infrastruttura si presenta in maniera analoga al diagramma riportato di seguito.



Per distribuire un'applicazione Elastic Beanstalk all'interno di un VPC utilizzando un host bastione, completa le fasi descritte nelle sottosezioni seguenti.

Fasi

- Creazione di un VPC con una sottorete pubblica e privata
- Creazione e configurazione del gruppo di sicurezza host bastione
- Aggiornamento del gruppo di sicurezza delle istanze
- Creazione di un host bastione

Creazione di un VPC con una sottorete pubblica e privata

Completa tutte le procedure descritte in VPC pubblico/privato. Quando distribuisci l'applicazione, devi specificare una coppia di EC2 chiavi Amazon per le istanze in modo da poterti connettere in remoto.

Per ulteriori informazioni su come specificare la coppia di chiavi dell'istanza, consulta Le EC2 istanze Amazon per il tuo ambiente Elastic Beanstalk.

Creazione e configurazione del gruppo di sicurezza host bastione

Crea un gruppo di sicurezza per l'host bastion e aggiungi regole che consentano il traffico SSH in entrata da Internet e il traffico SSH in uscita verso la sottorete privata che contiene le istanze Amazon. EC2

Per creare il gruppo di sicurezza dell'host bastione

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Fare clic su Security Groups (Gruppi di sicurezza) nel pannello di navigazione.
3. Scegli Crea gruppo di sicurezza.
4. Nella finestra di dialogo Create Security Group (Crea gruppo di sicurezza) inserire le seguenti informazioni, quindi selezionare Yes, Create (Sì, crea).

   Name tag (Tag nome) (opzionale)

   Inserire un tag nome per il gruppo di sicurezza.

   Group name (Nome gruppo)

   Inserire il nome del gruppo di sicurezza.

   Descrizione

   Inserire una descrizione per il gruppo di sicurezza.

   VPC

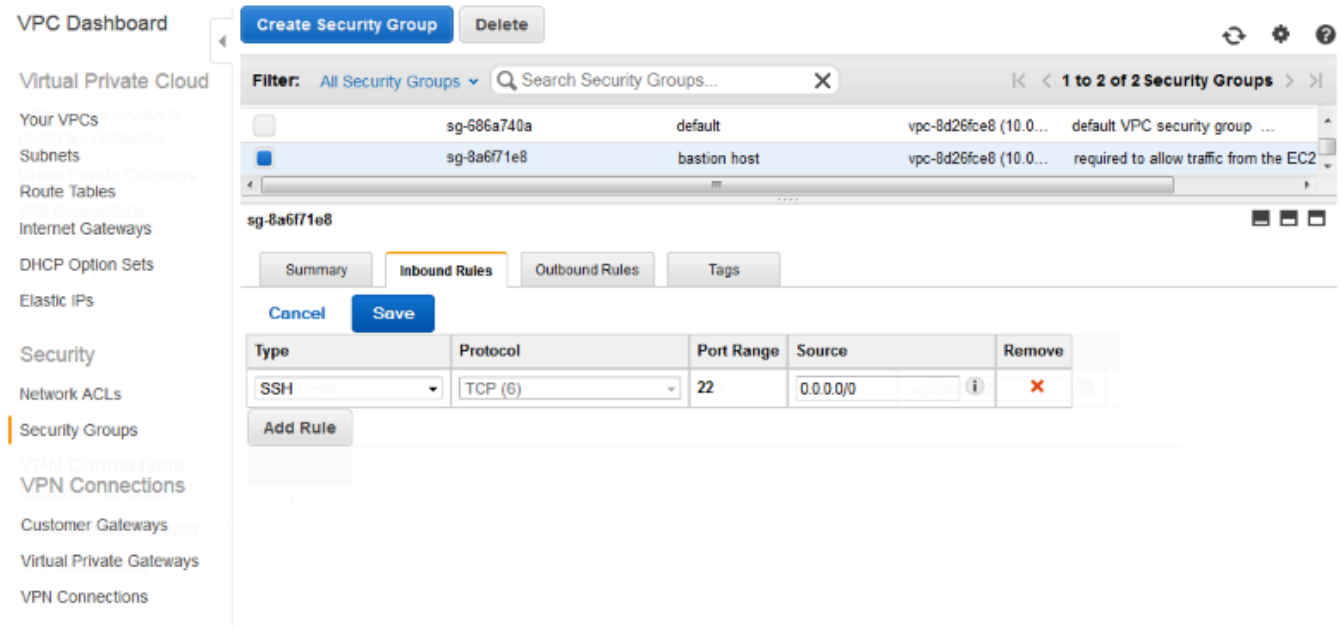
   Seleziona il tuo VPC.

   Il gruppo di sicurezza viene creato e incluso nella pagina Security Groups (Gruppi di sicurezza). Ricorda che il gruppo presenta un ID, ad esempio `sg-xxxxxxxx`. Potrebbe essere necessario attivare la visualizzazione della colonna Group ID (ID gruppo) facendo clic su Show/Hide (Mostra/nascondi) nell'angolo superiore destro della pagina.

Per configurare il gruppo di sicurezza dell'host bastione

1. Nell'elenco dei gruppi di sicurezza, selezionare la casella del gruppo di sicurezza appena creato per l'host bastione.
2. Nella scheda Inbound Rules (Regole in entrata), selezionare Edit (Modifica).
3. Se necessario, selezionare Add another rule (Aggiungi un'altra regola).
4. Se l'host bastione è un'istanza Linux, in Type (Tipo), selezionare SSH.

   Se l'host bastione è un'istanza Windows, in Type (Tipo), selezionare RDP.
5. Inserire l'intervallo CIDR dell'origine desiderata nel campo Source (Origine), quindi selezionare Save (Salva).



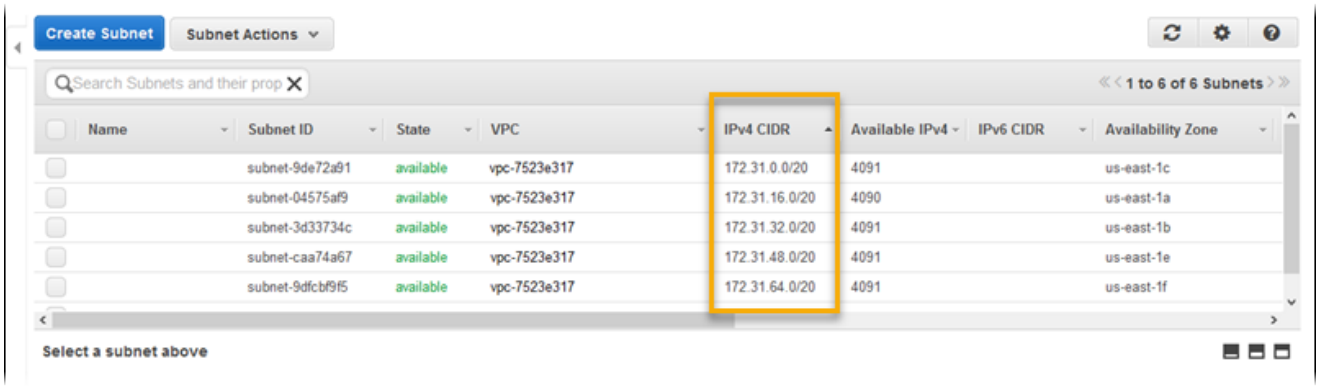
6. Nella scheda Outbound Rules (Regole in uscita), selezionare Edit (Modifica).
7. Se necessario, selezionare Add another rule (Aggiungi un'altra regola).
8. In Type (Tipo), selezionare il tipo specificato per la regola in entrata.
9. Nel campo Source (Origine), immettere l'intervallo CIDR della sottorete degli host nella sottorete privata del VPC.

   Per trovarlo:

   a. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
   b. Nel riquadro di navigazione, seleziona Sottoreti.

c. Annota il valore in IPv4 CIDR per ogni zona di disponibilità in cui hai host verso i quali desideri che l'host bastion effettui il bridge.

> **Note**
>
> Se sono disponibili host in più zone di disponibilità, crea una regola in uscita per ognuna di esse.



10. Scegli Save (Salva).

Aggiornamento del gruppo di sicurezza delle istanze

Per impostazione predefinita, il gruppo di sicurezza creato per le istanze non consente il traffico in entrata. Mentre Elastic Beanstalk si occupa di modificare il gruppo predefinito per le istanze per consentire il traffico SSH, il tuo compito è modificare il gruppo di sicurezza dell'istanza personalizzato per consentire il traffico RDP, se utilizzi istanze Windows.

Per aggiornare il gruppo di sicurezza delle istanze per RDP

1. Nell'elenco dei gruppi di sicurezza, selezionare la casella per il gruppo di sicurezza delle istanze.
2. Nella scheda Inbound (In entrata), selezionare Edito (Modifica).
3. Se necessario, selezionare Add another rule (Aggiungi un'altra regola).
4. Inserire i valori seguenti, quindi selezionare Save (Salva).

   Tipo

   `RDP`

Protocollo

`TCP`

Port Range (Intervallo porte)

`3389`

Origine

Inserire l'ID del gruppo di sicurezza dell'host bastione (ad esempio, `sg-8a6f71e8`), quindi scegliere Save (Salva).

Creazione di un host bastione

Per creare un bastion host, lanci un' EC2 istanza Amazon nella tua sottorete pubblica che fungerà da host bastion.

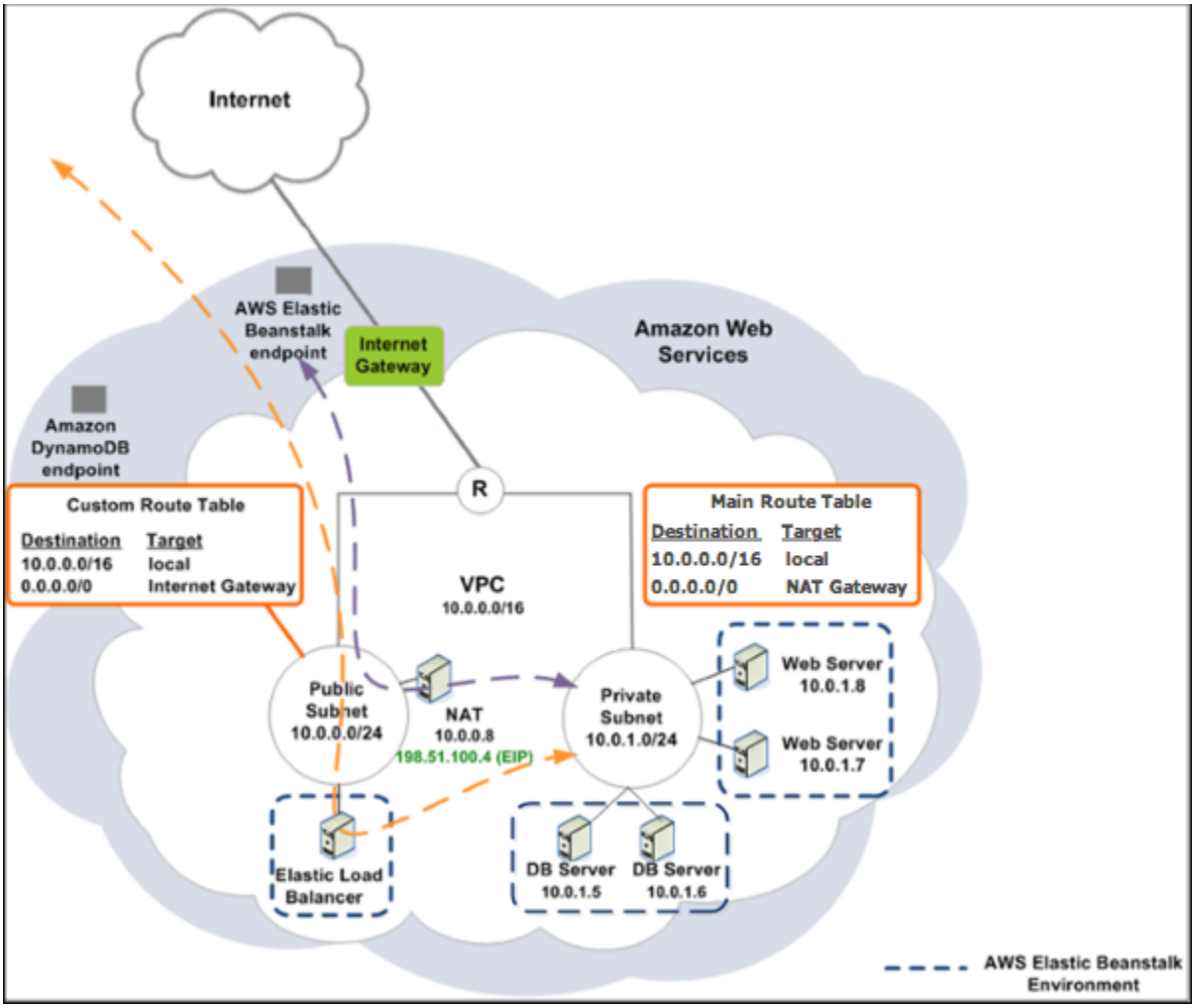
Per ulteriori informazioni sulla configurazione di un bastion host per istanze Windows nella sottorete privata, consulta Controllo dell'accesso di rete alle EC2 istanze utilizzando un server Bastion.

Per ulteriori informazioni sulla configurazione di un host bastione per le istanze Linux in una sottorete privata, consulta la pagina relativa alla connessione sicura alle istanze Linux in esecuzione in un VPC Amazon privato.

Esempio: avvio di un Elastic Beanstalk in un VPC con Amazon RDS

Questa sezione illustra le attività necessarie per distribuire un'applicazione Elastic Beanstalk con Amazon RDS in un VPC utilizzando un gateway NAT.

L'infrastruttura si presenta in maniera analoga al diagramma riportato di seguito.



> **Note**
>
> Se non hai mai utilizzato un'istanza database con la tua applicazione, prova ad aggiungerne una in un ambiente di test ed esegui la connessione a un'istanza database esterna prima di aggiungere una configurazione del VPC.

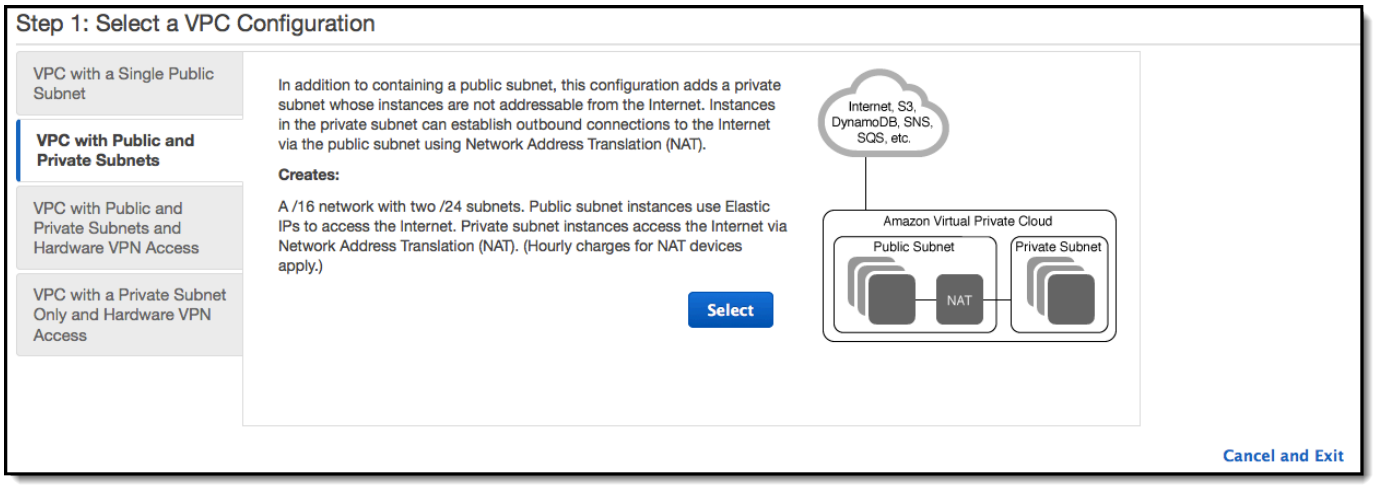
Creazione di un VPC con una sottorete pubblica e privata

Puoi utilizzare la console Amazon VPC per creare un VPC.

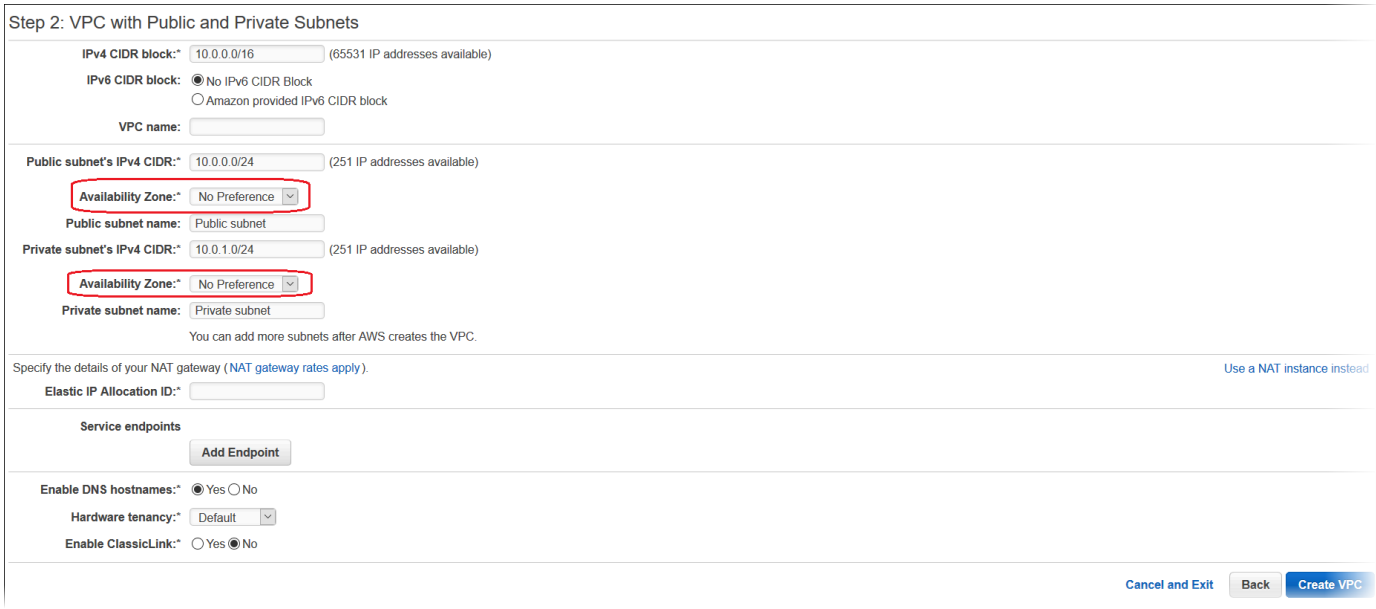
Per creare un VPC

1. Accedere alla console Amazon VPC.

2. Nel riquadro di navigazione, selezionare VPC Dashboard (Pannello di controllo VPC). Quindi, scegliere Create VPC (Crea VPC).
3. Selezionare VPC with Public and Private Subnets (VPC con sottoreti pubblica e privata), quindi Select (Seleziona).



4. Il sistema di bilanciamento del carico Elastic Load Balancing e le istanze Amazon EC2 devono trovarsi nella stessa zona di disponibilità affinché siano in grado di comunicare tra loro. Scegliere la stessa zona di disponibilità in ciascun elenco Availability Zone (Zona di disponibilità).

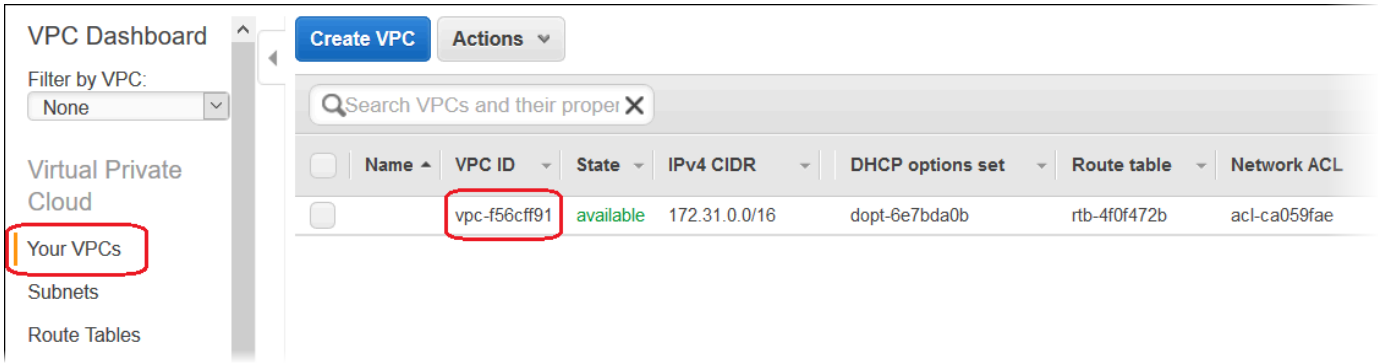


5. Scegliere un indirizzo IP elastico per il gateway NAT.
6. Seleziona Crea VPC.

La procedura guidata inizia a creare il VPC, le sottoreti e il gateway Internet. Inoltre, aggiorna la tabella di routing principale e crea una tabella di routing personalizzata. Infine, la procedura guidata crea un gateway NAT nella sottorete pubblica.

> **Note**
>
> Puoi scegliere di avviare un'istanza NAT nella sottorete pubblica invece di un gateway NAT. Per ulteriori informazioni, consulta Scenario 2: VPC con sottoreti pubbliche e private (NAT) nella Guida per l'utente di Amazon VPC.

7. Dopo aver creato il VPC, si riceverà un ID VPC che servirà per la fase successiva. Per visualizzare l'ID VPC, selezionare Your VPCs (I tuoi VPC) nel riquadro sinistro della console Amazon VPC.



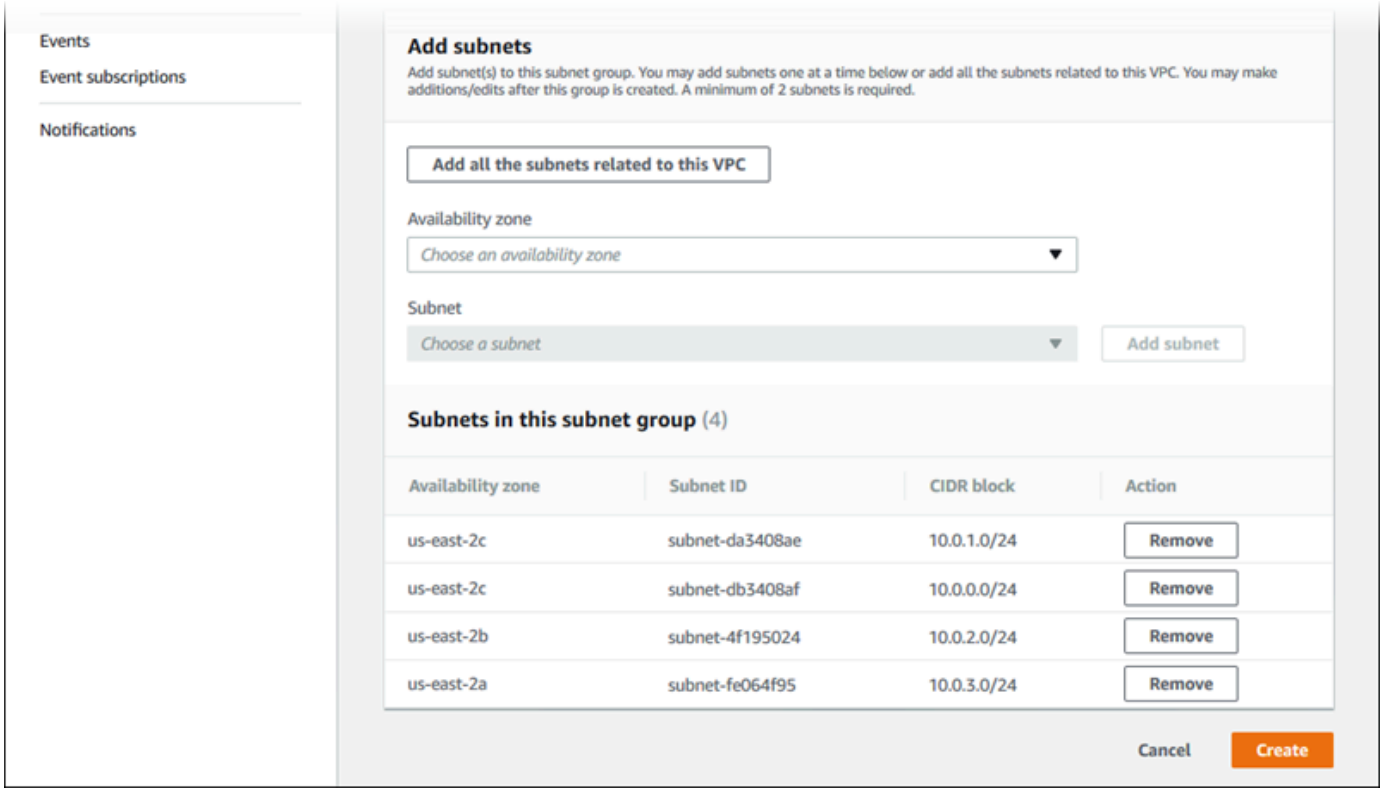
Per creare un gruppo di sottoreti del database

Un gruppo di sottoreti database per un VPC è una raccolta di sottoreti (di solito private) che è possibile configurare come istanze database RDS di back-end. Ogni gruppo di sottoreti DB deve avere almeno una sottorete per ogni zona di disponibilità in una determinata regione. AWS Per ulteriori informazioni, consulta Creazione di una sottorete nel VPC.

Per creare un gruppo di sottoreti del database

1. Apri la console Amazon RDS.
2. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
3. Scegli Create DB Subnet Group (Crea gruppo di sottoreti del database).
4. Selezionare Name (Nome), quindi digitare il nome del gruppo di sottoreti database.
5. Scegliere Description (Descrizione), quindi descrivere il gruppo di sottoreti database.

6. In VPC, selezionare l'ID del VPC creato in precedenza.
7. In Add subnets (Aggiungi sottoreti), selezionare Add all the subnets related to this VPC (Aggiungi tutte le sottoreti correlate a questo VPC).



8. Al termine, scegliere Create (Crea).

   Il nuovo gruppo di sottoreti database viene visualizzato nella lista dei gruppi di sottoreti della console Amazon RDS. È possibile selezionarlo per visualizzare i dettagli, ad esempio tutte le sottoreti associate a tale gruppo, nel riquadro dei dettagli nella parte inferiore della pagina.

Distribuzione in Elastic Beanstalk

Dopo aver configurato il VPC, puoi creare al suo interno il tuo ambiente e distribuire l'applicazione in Elastic Beanstalk. Puoi farlo utilizzando la console Elastic Beanstalk oppure puoi utilizzare i toolkit AWS CLI, l'interfaccia della riga di comando EB o AWS l'API Elastic Beanstalk. Se utilizzi la console Elastic Beanstalk, devi soltanto caricare il file `.war` o `.zip` e selezionare le impostazioni VPC all'interno della procedura guidata. Elastic Beanstalk crea quindi il tuo ambiente nel VPC e distribuisce l'applicazione. In alternativa, puoi utilizzare i AWS toolkit AWS CLI, EB CLI o l'API Elastic Beanstalk per distribuire la tua applicazione. Per farlo, devi definire le impostazioni delle opzioni del

tuo VPC in un file di configurazione da distribuire con il bundle di origine. Questo argomento descrive nel dettaglio entrambi i metodi.

Distribuzione con la console Elastic Beanstalk

La console Elastic Beanstalk illustra le fasi per creare il nuovo ambiente all'interno del VPC. Devi fornire un file `.war` (per le applicazioni Java) o un file `.zip` (per tutte le altre applicazioni). Nella pagina VPC Configuration (Configurazione VPC) della procedura guidata dell'ambiente Elastic Beanstalk, effettua le selezioni riportate di seguito:

VPC

Seleziona il tuo VPC.

Gruppo di sicurezza VPC

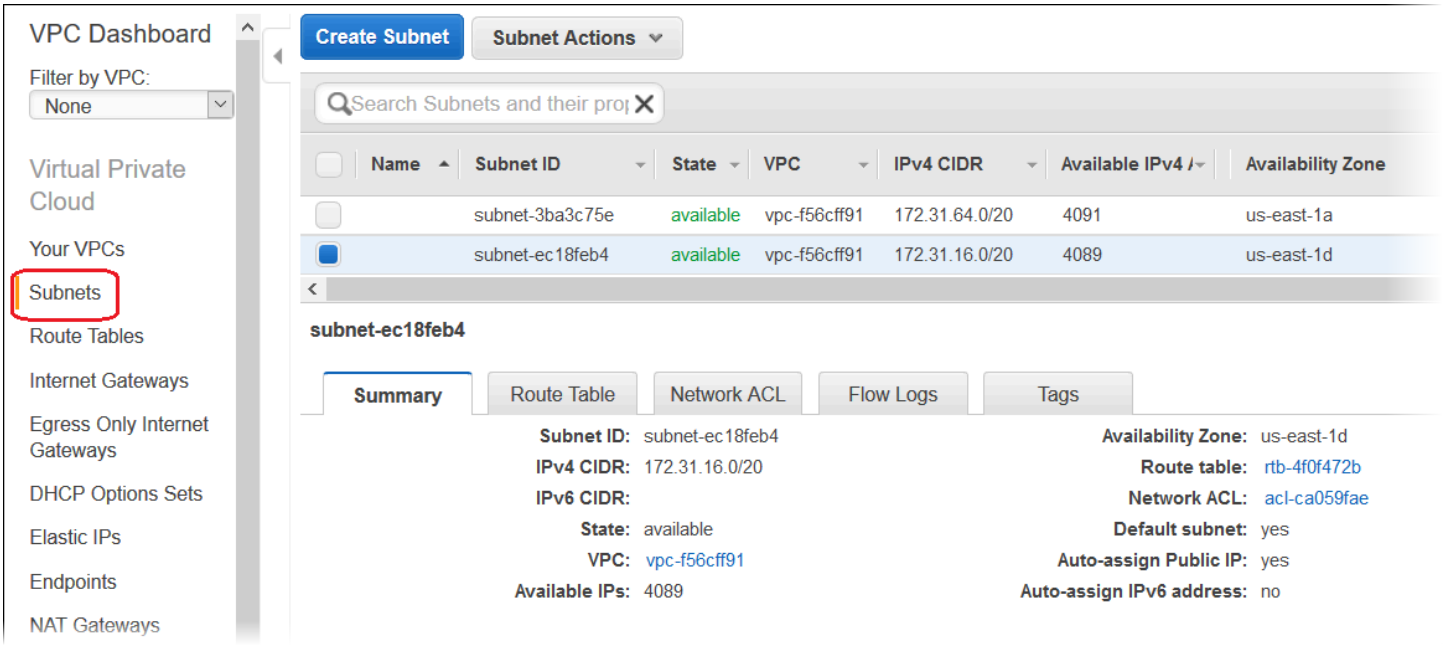
Seleziona il gruppo di sicurezza dell'istanza creato in precedenza.

ELB visibility (Visibilità ELB)

Seleziona `External` se il tuo sistema di bilanciamento del carico deve essere disponibile pubblicamente oppure seleziona `Internal` se invece deve essere disponibile solo nel tuo VPC.

Seleziona le sottoreti per il tuo sistema di bilanciamento del carico e per le istanze EC2. Assicurati di selezionare la sottorete pubblica per il sistema di bilanciamento del carico e la sottorete privata per le tue istanze Amazon EC2. Per impostazione predefinita, la procedura di creazione guidata del VPC crea la sottorete pubblica in `10.0.0.0/24` e la sottorete privata in `10.0.1.0/24`.

Puoi visualizzare gli ID delle sottoreti selezionando Subnets (Sottoreti) nella console Amazon VPC.



Implementazione con AWS kit di strumenti, CLI Web, AWS CLI o API

Quando distribuisci l'applicazione su Elastic Beanstalk utilizzando i toolkit, AWS l'EB AWS CLI CLI o l'API, puoi specificare le impostazioni delle opzioni VPC in un file e distribuirlo con il tuo bundle sorgente. Per ulteriori informazioni, consulta Personalizzazione avanzata dell'ambiente con i file di configurazione (`.ebextensions`).

Quando aggiorni le impostazioni delle opzioni, è necessario specificare almeno le seguenti informazioni:

- VPCId: contiene l'ID del VPC.
- Subnets (Sottoreti): contiene l'ID della sottorete del gruppo Auto Scaling. In questo esempio, questo è l'ID della sottorete privata.
- ELBSubnets (Sottoreti ELB): contiene l'ID della sottorete per il sistema di bilanciamento del carico. In questo esempio, questo è l'ID della sottorete pubblica.
- SecurityGroups—Contiene l'ID dei gruppi di sicurezza.
- DBSubnets (Sottoreti database): contiene l'ID delle sottoreti database.

> **Note**
>
> Quando si utilizzano sottoreti DB, è necessario creare sottoreti aggiuntive nel VPC per coprire tutte le zone di disponibilità della regione. AWS

Eventualmente, è anche possibile specificare le informazioni riportate di seguito:

- ELBScheme (Schema ELB): specifica `internal` per creare un sistema di bilanciamento del carico nel VPC in modo da impedire l'accesso all'applicazione Elastic Beanstalk dall'esterno del VPC.

Di seguito viene mostrato un esempio delle impostazioni delle opzioni che puoi utilizzare durante la distribuzione di un'applicazione Elastic Beanstalk all'interno di un VPC. Per ulteriori informazioni sulle impostazioni delle opzioni del VPC (inclusi esempi di specifica, valori predefiniti e valori validi), consulta la tabella degli spazi dei nomi aws:ec2:vpc in Opzioni di configurazione.

```
option_settings:
  - namespace: aws:autoscaling:launchconfiguration
    option_name: EC2KeyName
    value: ec2keypair

  - namespace: aws:ec2:vpc
    option_name: VPCId
    value: vpc-170647c

  - namespace: aws:ec2:vpc
    option_name: Subnets
    value: subnet-4f195024

  - namespace: aws:ec2:vpc
    option_name: ELBSubnets
    value: subnet-fe064f95

  - namespace: aws:ec2:vpc
    option_name: DBSubnets
    value: subnet-fg148g78

  - namespace: aws:autoscaling:launchconfiguration
    option_name: InstanceType
    value: m1.small

  - namespace: aws:autoscaling:launchconfiguration
    option_name: SecurityGroups
    value: sg-7f1ef110
```

> **Note**
>
> Quando usi le sottoreti DB, assicurati di avere delle sottoreti nel tuo VPC per coprire tutte le zone di disponibilità della regione. AWS

Utilizzo di Elastic Beanstalk con endpoint VPC

Questo argomento spiega i vantaggi che un endpoint VPC può offrire all'applicazione Elastic Beanstalk. Fornisce inoltre istruzioni per creare un endpoint VPC di interfaccia verso un servizio Elastic Beanstalk.

Un endpoint VPC ti consente di connettere privatamente il tuo VPC ai servizi supportati AWS e ai servizi endpoint VPC forniti da AWS PrivateLink, senza richiedere un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. Direct Connect

Le istanze nel VPC non richiedono indirizzi IP pubblici per comunicare con risorse nel servizio. Il traffico tra il VPC e gli altri servizi non lascia la rete Amazon. Per informazioni complete sugli endpoint VPC, consulta Endpoint VPC nella Guida per l'utente di Amazon VPC.

AWS Elastic Beanstalk supporta AWS PrivateLink, che fornisce connettività privata al servizio Elastic Beanstalk ed elimina l'esposizione del traffico verso la rete Internet pubblica. Per consentire all'applicazione di inviare richieste a Elastic AWS PrivateLink Beanstalk utilizzando, è necessario configurare un tipo di endpoint VPC noto come endpoint VPC di interfaccia (endpoint di interfaccia). Per ulteriori informazioni, consulta Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida per l'utente di Amazon VPC.

> **Note**
>
> Elastic AWS PrivateLink Beanstalk supporta e interfaccia gli endpoint VPC in un numero limitato di. Regioni AWS Stiamo lavorando per estendere il supporto a più Regioni AWS persone nelle prossime future.

IPv6 supporto con configurazione dual-stack

Elastic Beanstalk supporta il traffico in entrata su e. IPv4 IPv6 Questa sezione descrive gli endpoint pubblici che supportano entrambi IPv4 IPv6 e spiega anche come configurare gli endpoint VPC Elastic Beanstalk per supportare il traffico dual-stack.

Endpoint pubblici

Il servizio Elastic Beanstalk dispone di due set di endpoint, costituiti dagli endpoint più vecchi e dagli endpoint più recenti con funzionalità dual-stack. IPv4 Entrambi i set di endpoint seguono gli standard di denominazione: AWS

- IPv4 gli endpoint utilizzano il formato domain `amazonaws.com` — per l'endpoint di servizio generale: `elasticbeanstalk.region.amazonaws.com`
- Gli endpoint dual-stack utilizzano il formato domain `api.aws` — per l'endpoint di servizio generale:: `elasticbeanstalk.region.api.aws`

Gli endpoint for service health e FIPS hanno nomi host diversi, ma seguono lo stesso schema di nomi di dominio. Per un elenco degli endpoint, consulta gli endpoint del servizio Elastic Beanstalk in. Riferimenti generali di Amazon Web Services

Richieste a Elastic Beanstalk

Quando invii richieste al servizio Elastic Beanstalk AWS CLIcon o l'SDK, puoi IPv4 specificare un endpoint AWS o un endpoint dual-stack. L'SDK AWS CLI e l' AWS SDK utilizzano gli endpoint IPv4 - only per impostazione predefinita se non viene specificato un URL dell'endpoint.

L'esempio seguente dimostra l' AWS CLI invio di una richiesta a un endpoint dual-stack:

Example

```
aws elasticbeanstalk list-available-solution-stacks \
    --endpoint-url "https://elasticbeanstalk.us-east-1.api.aws"
```

L'esempio seguente mostra l'SDK AWS Python che invia una richiesta a un endpoint dual-stack:

Example

```
import boto3

dual_stack_eb_client = boto3.client(
    service_name='elasticbeanstalk',
    region_name='us-east-1',
    endpoint_url='https://elasticbeanstalk.us-east-1.api.aws';
)

print(dual_stack_eb_client.list_available_solution_stacks())
```

Endpoint VPC per dual-stack IPs

Per configurare gli endpoint VPC Elastic Beanstalk per supportare il traffico dual-stack, specifica dualstack per il parametro del tipo di indirizzo IP dell'endpoint VPC. Puoi specificare questo campo tramite, l'SDK o la console. AWS CLIAWS AWS PrivateLink Per istruzioni su come eseguire questa operazione nella AWS PrivateLink console, consulta Creare un endpoint VPC nella Guida.AWS PrivateLink

> **Note**
>
> È necessario specificare il tipo di indirizzo IP dell'endpoint VPC come uno o due IPv4stack. Al momento gli endpoint VPC Elastic Beanstalk non supportano un tipo di indirizzo IPv6IP di, il che indicherebbe il supporto solo. IPv6 L'opzione dualstack consente sia il protocollo che quello Internet. IPv4 IPv6

L'esempio seguente dimostra come creare un endpoint VPC dual-stack con: AWS CLI

Example

```
aws ec2 create-vpc-endpoint \
    --vpc-id "vpc-example"
    --service-name "com.amazonaws.us-east-1.elasticbeanstalk"
    --ip-address-type "dualstack"
```

Impostazione di un endpoint VPC per Elastic Beanstalk

Per creare l'endpoint VPC di interfaccia per il servizio Elastic Beanstalk nel VPC, segui la procedura Creazione di un endpoint di interfaccia.

- Per Service Name, scegli com.amazonaws. ***region***. elasticbeanstalk.
- Per il tipo di indirizzo IP, scegli uno dei due o Dualstack. IPv4 Al momento gli endpoint VPC Elastic Beanstalk non supportano un tipo di indirizzo IPv6IP di, il che indicherebbe il supporto solo. IPv6 L'opzione Dualstack consente sia i protocolli Internet che quelli di Internet. IPv4 IPv6

Se il tuo VPC è configurato con l'accesso pubblico a Internet, l'applicazione può comunque accedere a Elastic Beanstalk su Internet utilizzando l'endpoint o l'endpoint pubblico. `elasticbeanstalk.region.amazonaws.com` `elasticbeanstalk.region.api.aws` È

possibile evitare questo problema assicurando che Enable DNS name (Abilita nome DNS) sia abilitato durante la creazione dell'endpoint (true per impostazione predefinita). In questo modo viene aggiunta una voce DNS nel VPC che associa l'endpoint del servizio pubblico all'endpoint VPC dell'interfaccia.

Impostazione di un endpoint VPC per l'integrità avanzata

Se hai abilitato la reportistica sanitaria avanzata per il tuo ambiente, puoi configurare anche l'invio di informazioni sanitarie avanzate. AWS PrivateLink Le informazioni di stato avanzate vengono inviate dal daemon `healthd`, un componente Elastic Beanstalk delle istanze dell'ambiente, a un servizio di stato avanzato Elastic Beanstalk separato. Per creare un endpoint VPC di interfaccia per questo servizio nel VPC, segui la procedura Creazione di un endpoint di interfaccia.

- Per Nome del servizio, scegli com.amazonaws. ***`region`***. elasticbeanstalk-health.
- Per il tipo di indirizzo IP, scegli uno dei due o Dualstack. IPv4 Al momento gli endpoint VPC Elastic Beanstalk non supportano un tipo di indirizzo IPv6IP di, il che indicherebbe il supporto solo. IPv6 L'opzione Dualstack consente sia i protocolli Internet che quelli di Internet. IPv4 IPv6

> **Important**
>
> Il `healthd` demone invia informazioni sanitarie avanzate all'endpoint pubblico o. `elasticbeanstalk-health.`*`region`*`.amazonaws.com` `elasticbeanstalk-health.`*`region`*`.api.aws` Se il VPC è configurato con accesso a Internet pubblico e Enable DNS name (Abilita nome DNS) è disabilitato per l'endpoint VPC, le informazioni di integrità avanzate viaggiano attraverso Internet pubblico. Questa probabilmente non è la tua intenzione quando configuri un endpoint VPC con integrità avanzata. Assicurarsi che Enable DNS name (Attiva nome DNS) sia abilitato (true per impostazione predefinita).

Utilizzo di endpoint VPC in un VPC privato

Un VPC privato, o una sottorete privata in un VPC, non ha accesso a Internet pubblico. Potrebbe essere necessario eseguire l'ambiente Elastic Beanstalk in un VPC privato e configurare gli endpoint VPC di interfaccia per la sicurezza avanzata. In questo caso, tieni presente che l'ambiente in uso potrebbe tentare di connettersi a Internet per altri motivi che non siano contattare il servizio Elastic Beanstalk. Per ulteriori informazioni sull'esecuzione di un ambiente in un VPC privato, consulta the section called “Esecuzione di un ambiente Elastic Beanstalk in un VPC privato”.

Utilizzo dei criteri endpoint per controllare l'accesso con gli endpoint VPC

Questo argomento spiega come collegare una policy agli endpoint VPC per controllare l'accesso alla tua applicazione (il tuo servizio) e all'ambiente Elastic Beanstalk.

Una policy per gli endpoint è una politica delle risorse AWS Identity and Access Management (IAM) che controlla l'accesso dall'endpoint al servizio specificato. La policy endpoint è specifica per l'endpoint. Inoltre, è separata da tutte le policy IAM dell'utente o dell'istanza disponibili per l'ambiente e non le sovrascrive né le sostituisce.

Per impostazione predefinita, un endpoint VPC consente l'accesso completo al servizio a cui è associato. Quando si crea o si modifica un endpoint, è possibile allegare ad esso una policy per controllare l'accesso a risorse specifiche associate al servizio. Per dettagli sulla creazione e l'utilizzo delle policy degli endpoint VPC, consulta Controllare l'accesso agli endpoint VPC utilizzando le policy degli endpoint nella Guida.AWS PrivateLink

> **Note**
>
> Quando si creano policy restrittive per gli endpoint, potrebbe essere necessario aggiungere autorizzazioni specifiche alle risorse richieste, in modo che l'accesso a tali risorse non sia bloccato dalla policy dell'endpoint. In questo modo si garantisce che l'ambiente continui a essere distribuito e a funzionare correttamente.

L'esempio seguente nega a tutti gli utenti l'autorizzazione per terminare un ambiente tramite l'endpoint VPC e consente l'accesso completo a tutte le altre azioni.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticbeanstalk:TerminateEnvironment",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
```

```
        }
    ]
}
```

Autorizzazioni richieste per i bucket Amazon S3 per policy restrittive sugli endpoint VPC

Se aggiungi restrizioni alle policy degli endpoint VPC, devi includere autorizzazioni specifiche per i bucket Amazon S3 per garantire che l'ambiente continui a essere distribuito e funzionare correttamente. Questa sezione spiega i bucket S3 richiesti e include policy di esempio.

Argomenti

- Bucket S3 che archiviano le risorse per gestire le piattaforme ambientali
- S3 Buckets di proprietà di CloudFormation
- Bucket S3 di proprietà degli account dei clienti per archiviare il codice sorgente e altri articoli
- Bucket S3 di proprietà degli account dei clienti per supportare l'autenticazione del registro Docker
- Aggiornamento della policy degli endpoint VPC

Bucket S3 che archiviano le risorse per gestire le piattaforme ambientali

Il servizio Elastic Beanstalk possiede bucket S3 che memorizzano le risorse associate a uno stack di soluzioni (versione della piattaforma). Queste risorse includono i file di configurazione, l'applicazione di esempio e i tipi di istanze disponibili. Quando Elastic Beanstalk crea e gestisce l'ambiente, recupera le informazioni richieste per la versione specifica della piattaforma dal bucket di risorse per ogni corrispondente. Regione AWS

Secchio S3 ARN

`arn:aws:s3:::elasticbeanstalk-samples-`*`region`*

Amazon Linux 2 e versioni successive

- `arn:aws:s3:::elasticbeanstalk-platform-assets-`*`region`*

> **Note**
>
> Il nome del bucket segue una convenzione diversa per la regione BJS. La stringa public-beta-cn-north-1 viene utilizzata al posto di. *`region`* Ad esempio,

`arn:aws:s3:::elasticbeanstalk-platform-assets-public-beta-cn-north-1`.

Windows Server, Amazon Linux (AMI), Amazon Linux 2 e versioni successive

- `arn:aws:s3:::elasticbeanstalk-env-resources-`*`region`*
- `arn:aws:s3:::elasticbeanstalk-`*`region`*

> **Note**
>
> I nomi dei bucket per platform-assets e env-resources seguono convenzioni diverse in alcune regioni. Per maggiori dettagli, consulta la sezione dedicata ai modelli di denominazione dei bucket specifici della regione riportata di seguito.

Modelli ARN di bucket specifici per regione

Region	Tipo di benna	Modello ARN del secchio
me-central-1	risorse della piattaforma	`arn:aws:s3:::elasticbeanstalk-platform-assets-me-central-1-f08b818c`
	risorse env	`arn:aws:s3:::elasticbeanstalk-env-resources-me-central-1-f08b818c`

Operazioni

GetObject

Esempio di policy per gli endpoint VPC

L'esempio seguente illustra come fornire l'accesso ai bucket S3 necessari per le operazioni di Elastic Beanstalk nella regione degli Stati Uniti orientali (Ohio) (us-east-2). L'esempio elenca tutti i bucket per entrambe le piattaforme Amazon Linux e Windows Server. Aggiorna la tua policy per includere solo i bucket che si applicano al sistema operativo del tuo ambiente.

Policy di esempio

> **Important**
>
> Consigliamo di evitare l'uso di caratteri jolly (*) al posto delle Regioni specifiche in questa policy. Ad esempio, è preferibile usare `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` anziché `arn:aws:s3:::cloudformation-waitcondition-*/*`. L'utilizzo di caratteri jolly potrebbe fornire l'accesso ai bucket S3 a cui non si intende concedere l'accesso. Se desideri utilizzare la politica per più di una regione, ti consigliamo di ripetere il primo `Statement` blocco per ogni regione.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToAWSResources",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["s3:GetObject"],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-platform-assets-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-env-resources-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-env-resources-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-samples-us-east-2/*"
            ]
         }
    ]
}
```

S3 Buckets di proprietà di CloudFormation

Elastic CloudFormation Beanstalk utilizza per creare risorse per il tuo ambiente. CloudFormation possiede ciascuno bucket S3 Regione AWS per monitorare le risposte alle condizioni di attesa.

Servizi come Elastic Beanstalk CloudFormation comunicano con loro inviando richieste a un URL Amazon S3 predefinito per il bucket S3 di cui è proprietario. CloudFormation CloudFormation crea l'URL Amazon S3 predefinito utilizzando `cloudformation.amazonaws.com` il service principal.

Per informazioni più dettagliate, consulta Considerazioni sugli endpoint CloudFormation VPC nella AWS CloudFormation Guida per l'utente. Per ulteriori informazioni sulle impostazioni predefinite URLs, consulta Working with presigned URLs nella Amazon S3 User Guide.

Secchio S3 ARN

- `arn:aws:s3:::cloudformation-waitcondition-`*`region`*

  Quando si utilizzano condizioni di attesa, i nomi delle Regioni contengono trattini. Ad esempio, us-west-2.

- `arn:aws:s3:::cloudformation-custom-resource-response-`*`region`*

  Quando si utilizzano risorse personalizzate, i nomi delle Regioni non contengono trattini. Ad esempio, uswest2.

Operazioni

GetObject

Esempio di policy per gli endpoint VPC

L'esempio seguente illustra come fornire l'accesso ai bucket S3 necessari per le operazioni di Elastic Beanstalk nella regione degli Stati Uniti orientali (Ohio) (us-east-2).

Policy di esempio

> **Important**
>
> Consigliamo di evitare l'uso di caratteri jolly (*) al posto delle Regioni specifiche in questa policy. Ad esempio, è preferibile usare `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` anziché `arn:aws:s3:::cloudformation-waitcondition-*/*`. L'utilizzo di caratteri jolly potrebbe fornire l'accesso ai bucket S3 a cui non si intende concedere l'accesso. Se desideri utilizzare la policy per più di una regione, ti consigliamo di ripetere il primo blocco per ogni regione. `Statement`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToCloudFormation",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["s3:GetObject"],
            "Resource": [
                "arn:aws:s3:::cloudformation-waitcondition-us-east-2/*",
                "arn:aws:s3:::cloudformation-custom-resource-response-us-east-2/
*"
            ]
         }
    ]
}
```

Bucket S3 di proprietà degli account dei clienti per archiviare il codice sorgente e altri articoli

Questo bucket è di proprietà dell'account AWS cliente proprietario dell'ambiente. Memorizza risorse specifiche dell'ambiente, come il codice sorgente e i registri richiesti.

Secchio S3 ARN

`arn:aws:s3:::elasticbeanstalk-region-account-id`

Operazioni

- GetObject
- GetObjectAcl
- PutObject
- PutObjectAcl
- ListBucket

Esempio di policy per gli endpoint VPC

L'esempio seguente illustra come fornire l'accesso ai bucket S3 necessari per le operazioni di Elastic Beanstalk nella regione degli Stati Uniti orientali (Ohio) (us-east-2) e per l'ID account di esempio 123456789012. AWS

Policy di esempio

> **Important**
>
> Consigliamo di evitare l'uso di caratteri jolly (*) al posto delle Regioni specifiche in questa policy. Ad esempio, è preferibile usare `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` anziché `arn:aws:s3:::cloudformation-waitcondition-*/*`. L'utilizzo di caratteri jolly potrebbe fornire l'accesso ai bucket S3 a cui non si intende concedere l'accesso. Se desideri utilizzare la policy per più di una regione, ti consigliamo di ripetere il primo blocco per ogni regione. `Statement`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToCustomerItems",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": [
               "s3:GetObject",
               "s3:GetObjectAcl",
               "s3:PutObject",
               "s3:PutObjectAcl",
               "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/*"
            ]
         }
    ]
}
```

Bucket S3 di proprietà degli account dei clienti per supportare l'autenticazione del registro Docker

Questo bucket si applica solo agli ambienti basati sulla piattaforma Docker. Il bucket memorizza un file utilizzato per l'autenticazione in un registro Docker privato che risiede su un bucket S3 fornito dal cliente. Per ulteriori informazioni, consulta il capitolo Utilizzo del file `Dockerrun.aws.json v3` nella piattaforma Docker di questa guida.

Secchio S3 ARN

L'ARN varia in base all'account del cliente.

L'ARN del bucket S3 ha il seguente formato: `arn:aws:s3:::`*`bucket-name`*

Operazioni

GetObject

Esempio di policy per gli endpoint VPC

L'esempio seguente illustra come fornire l'accesso a un bucket S3 con il nome amzn-s3-demo-bucket1.

Policy di esempio

JSON

```
  {
  "Version":"2012-10-17",
  "Statement": [
     {
        "Sid": "AllowRequestsToDockerRegistryAuth",
         "Effect": "Allow",
         "Action": [
            "s3:GetObject"
          ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1"
      ]
    }
  ]
}
```

Aggiornamento della policy degli endpoint VPC

Poiché a un endpoint VPC è associata una sola policy, è necessario combinare tutte le autorizzazioni in un'unica policy. L'esempio seguente fornisce tutti gli esempi precedenti combinati in uno solo.

Per dettagli sulla creazione e l'utilizzo delle policy degli endpoint VPC, consulta Controllare l'accesso agli endpoint VPC utilizzando le policy degli endpoint nella Guida.AWS PrivateLink

Come negli esempi precedenti, quello seguente illustra come fornire l'accesso ai bucket S3 necessari per le operazioni di Elastic Beanstalk nella regione degli Stati Uniti orientali (Ohio) (us-east-2). Include anche bucket con l'ID account di esempio 123456789012 e il nome del bucket di esempio amzn-s3-demo-bucket1. AWS

> **Important**
>
> Consigliamo di evitare l'uso di caratteri jolly (*) al posto delle Regioni specifiche in questa policy. Ad esempio, è preferibile usare `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` anziché `arn:aws:s3:::cloudformation-waitcondition-*/*`. L'utilizzo di caratteri jolly potrebbe fornire l'accesso ai bucket S3 a cui non si intende concedere l'accesso. Se desideri utilizzare la politica per più di una regione, ti consigliamo di ripetere il primo blocco per ogni regione. `Statement`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToAWSResources",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-platform-assets-us-east-2/*",
                "arn:aws:s3:::elasticbeanstalk-env-resources-us-east-2/*",
```

```
                "arn:aws:s3:::elasticbeanstalk-samples-us-east-2/*"
            ]
        },
        {
            "Sid": "AllowRequestsToCloudFormation",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::cloudformation-waitcondition-us-east-2/*",
                "arn:aws:s3:::cloudformation-custom-resource-response-us-east-2/
*"
            ]
        },
        {
            "Sid": "AllowRequestsToCustomerItems",
            "Effect": "Allow",
            "Action": [
               "s3:GetObject",
               "s3:GetObjectAcl",
              "s3:PutObject",
               "s3:PutObjectAcl",
               "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/*"
            ]
        },
        {
            "Sid": "AllowRequestsToDockerRegistryAuth",
            "Effect": "Allow",
            "Action": [
            "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        }
    ]
}
```

AWS Elastic Beanstalk sicurezza

Utilizza questo capitolo per saperne di più sulle attività di sicurezza di cui è responsabile Elastic Beanstalk, insieme alle configurazioni di sicurezza da prendere in considerazione quando utilizzi Elastic Beanstalk per raggiungere i tuoi obiettivi di sicurezza e conformità.

La sicurezza del cloud è la massima priorità. AWS In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo concetto come sicurezza del cloud e sicurezza nel cloud.

Security of the Cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce tutti i servizi offerti nel AWS Cloud e della fornitura di servizi che è possibile utilizzare in modo sicuro. La nostra responsabilità in AWS materia di sicurezza è la massima priorità e l'efficacia della nostra sicurezza viene regolarmente testata e verificata da revisori di terze parti nell'ambito dei Programmi di AWS conformità. Consulta i programmi AWS Services in Scope of AWS assurance per informazioni relative a Elastic Beanstalk.

Sicurezza nel cloud: la responsabilità dell'utente è determinata dal AWS servizio utilizzato e da altri fattori, tra cui la sensibilità dei dati, i requisiti dell'organizzazione e le leggi e i regolamenti applicabili. Questa documentazione consente di comprendere come applicare il modello di responsabilità condivisa quando si usa Elastic Beanstalk.

Argomenti

- Protezione dei dati in Elastic Beanstalk
- Identity and Access Management per Elastic Beanstalk
- Registrazione e monitoraggio in Elastic Beanstalk
- Convalida della conformità per Elastic Beanstalk
- Resilienza in Elastic Beanstalk
- Sicurezza dell'infrastruttura in Elastic Beanstalk
- Analisi della configurazione e delle vulnerabilità in Elastic Beanstalk
- Best practice relative alla sicurezza di Elastic Beanstalk

Protezione dei dati in Elastic Beanstalk

Il modello di responsabilità AWS condivisa Il modello di si applica alla protezione dei dati in AWS Elastic Beanstalk. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati . Per informazioni sulla protezione dei dati in Europa, consulta il General Data Protection Regulation (GDPR) Center.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Elastic Beanstalk Servizi AWS o altro utilizzando la console, AWS CLI l'API o gli SDK. AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Argomenti

- Protezione dei dati tramite crittografia
- Riservatezza del traffico Internet

Protezione dei dati tramite crittografia

Puoi utilizzare diverse forme di crittografia dei dati per proteggere i tuoi dati Elastic Beanstalk. La protezione dei dati si riferisce alla protezione dei dati in transito (mentre viaggiano da e verso Elastic Beanstalk) e a riposo (mentre sono archiviati nei data center). AWS

Crittografia dei dati in transito

È possibile ottenere la protezione dei dati in transito in due modi: crittografare la connessione utilizzando Secure Sockets Layer (SSL) o utilizzare la crittografia lato client (dove l'oggetto viene crittografato prima dell'invio). Entrambi i metodi sono validi per proteggere i dati dell'applicazione. Per proteggere la connessione, crittografala utilizzando SSL ogni volta che la tua applicazione, i suoi sviluppatori e amministratori e i suoi utenti finali inviano o ricevono oggetti. Per ulteriori informazioni sulla crittografia del traffico Web da e verso la tua applicazione, consulta the section called “HTTPS”.

Client-side la crittografia non è un metodo valido per proteggere il codice sorgente nelle versioni delle applicazioni e nei pacchetti di sorgenti caricati. Elastic Beanstalk necessita dell'accesso a questi oggetti, pertanto non possono essere crittografati. Di conseguenza, assicurati di proteggere la connessione tra l'ambiente di sviluppo o distribuzione e Elastic Beanstalk.

Crittografia dei dati inattivi

Per proteggere i dati inattivi dell'applicazione, scopri di più sulla protezione dei dati nel servizio di storage utilizzato dall'applicazione. Ad esempio, consultare la sezione Protezione dei dati in Amazon RDS nella Guida per l'utente di Amazon RDS, Protezione dei dati in Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service o Crittografia dei dati e dei metadati in EFS nella Guida per l'utente di Amazon Elastic File System.

Elastic Beanstalk archivia vari oggetti in un bucket crittografato Amazon Simple Storage Service (Amazon S3) creato per ogni regione in cui crei ambienti. AWS Poiché Elastic Beanstalk mantiene la crittografia predefinita fornita da Amazon S3, crea bucket Amazon S3 crittografati. Per informazioni dettagliate, vedi the section called “Simple Storage Service (Amazon S3)”. Puoi fornire alcuni degli oggetti archiviati e inviarli a Elastic Beanstalk, ad esempio, versioni dell'applicazione e bundle di

origine. Elastic Beanstalk genera altri oggetti, ad esempio file di log. Oltre ai dati archiviati da Elastic Beanstalk, l'applicazione and/or può trasferire i dati dell'archivio come parte del suo funzionamento.

Per proteggere i dati archiviati su volumi Amazon Elastic Block Store (Amazon EBS) collegati alle istanze del tuo ambiente, abilita la crittografia Amazon EBS per impostazione predefinita nel tuo AWS account e nella tua regione. Se abilitati, tutti i nuovi volumi Amazon EBS e le relative istantanee vengono crittografati automaticamente tramite AWS Key Management Service chiavi. Per ulteriori informazioni, consulta Encryption by default nella Amazon EBS User Guide.

Per ulteriori informazioni sulla protezione dei dati, consulta il post del blog AWS Modello di responsabilità condivisa e GDPR su AWS Security Blog.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Riservatezza del traffico Internet

Puoi utilizzare Amazon Virtual Private Cloud (Amazon VPC) per creare delimitatori tra le risorse nella tua applicazione Elastic Beanstalk e controllare il traffico tra di loro, la rete locale e Internet. Per informazioni dettagliate, consulta the section called “Amazon VPC”.

Per ulteriori informazioni sulla sicurezza di Amazon VPC, consulta Sicurezza nella Guida per l'utente di Amazon VPC.

Per ulteriori informazioni sulla protezione dei dati, consulta il post del blog AWS Modello di responsabilità condivisa e GDPR su AWS Security Blog.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Identity and Access Management per Elastic Beanstalk

AWS Identity and Access Management (IAM) è un AWS servizio che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) a utilizzare risorse AWS Elastic Beanstalk . IAM è un AWS servizio che puoi utilizzare senza costi aggiuntivi.

Per informazioni dettagliate su come lavorare con IAM, consulta Utilizzo di Elastic Beanstalk con AWS Identity and Access Management.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

AWS politiche gestite per AWS Elastic Beanstalk

Una policy AWS gestita è una policy autonoma creata e amministrata da. AWS AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Si consiglia pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

Elastic AWS Beanstalk: aggiornamenti alle policy gestite

Visualizza i dettagli sugli aggiornamenti alle policy AWS gestite per Elastic Beanstalk dal 1° marzo 2021.

Per visualizzare il codice sorgente JSON per una politica gestita specifica, consulta la Managed Policy Reference Guide AWS .

Modifica	Descrizione	Data
AWSElasticBeanstalkWebTier—Politica esistente aggiornata	Questa politica è stata aggiornata per aggiungere le autorizzazioni Amazon Nova e AWS Marketplace per l'analisi dell'ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	29 aprile 2026

Modifica	Descrizione	Data
AWSElasticBeanstalkWorkerTier—Politica esistente aggiornata	Questa politica è stata aggiornata per aggiungere le autorizzazioni Amazon Nova e AWS Marketplace per l'analisi dell'ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	29 aprile 2026
AWSElasticBeanstalkMulticontainerDocker—Politica esistente aggiornata	Questa politica è stata aggiornata per aggiungere le autorizzazioni Amazon Nova e AWS Marketplace per l'analisi dell'ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	29 aprile 2026
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di eseguire aggiornamenti gestiti quando è abilitata. Propagazione di tag ai modelli di avvio Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	17 marzo 2026

Modifica	Descrizione	Data
AWSElasticBeanstalkWebTier—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di utilizzare Amazon Bedrock per l'analisi dell'ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	11 marzo 2026
AWSElasticBeanstalkWorkerTier—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di utilizzare Amazon Bedrock per l'analisi dell'ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	11 marzo 2026
AWSElasticBeanstalkMulticontainerDocker—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di utilizzare Amazon Bedrock per l'analisi dell'ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	11 marzo 2026

Modifica	Descrizione	Data
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di eseguire Propagazione di tag ai modelli di avvio aggiornamenti gestiti quando è abilitata per una singola istanza. Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	27 gennaio 2026
AdministratorAccess-AWSElastic Beanstalk: politica esistente aggiornata	Questa policy è stata aggiornata per consentir e a Elastic Beanstalk di configurare le impostazioni dei blocchi di accesso pubblico e i controlli di proprietà dei bucket sui bucket S3. Per ulteriori informazioni, consulta Gestione delle policy utente Elastic Beanstalk.	12 novembre 2025
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di eseguire aggiornamenti gestiti quando è abilitata. Propagazione di tag ai modelli di avvio Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	27 febbraio 2025

Modifica	Descrizione	Data
AdministratorAccess-AWSElastic Beanstalk: politica esistente aggiornata	Questa politica è stata aggiornata per sostituire l'StringLikeoperatore con l'ArnLikeoperatore per valutare le chiavi di tipo ARN nel blocco delle condizioni. `iam:PolicyArn` Ciò garantisce un'applicazione più sicura. Per ulteriori informazioni, consulta Gestione delle policy utente Elastic Beanstalk.	11 dicembre 2024
Le seguenti politiche sono state aggiornate: • AWSElasticBeanstalkInternalMaintenanceRolePolicy • AWSElasticBeanstalkMaintenance • AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy • AWSElasticBeanstalkManagedUpdatesServiceRolePolicy • AWSElasticBeanstalkRoleCore	Queste policy sono state aggiornate per consentire a Elastic Beanstalk di aggiungere o rimuovere tag quando crea o AWS CloudFormation aggiorna uno stack o un set di modifiche. Per ulteriori informazioni su `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy`, consultare Autorizzazioni dei ruoli collegati ai servizi per Elastic Beanstalk. Per ulteriori informazioni su `AWSElasticBeanstalkRoleCore`, consultare Policy per l'integrazione con altri servizi.	30 aprile 2024

Modifica	Descrizione	Data
AWSElasticBeanstalkService—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di taggare le risorse in seguito alla creazione per Elastic Load Balancing, gruppi con dimensionamento automatico (ASG) ed Amazon ECS. **Note** Questa policy è stata sostituita da `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` . Sebbene questa policy non sia più disponibile per essere collegata a nuovi utenti, gruppi o ruoli IAM, può essere collegata a quelli già esistenti. Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	10 maggio 2023

Modifica	Descrizione	Data
AWSElasticBeanstalkMulticontainerDocker—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di taggare le risorse in seguito alla creazione per Amazon ECS. Per ulteriori informazioni, consulta Gestione dei profili dell'istanza Elastic Beanstalk.	23 marzo 2023
AWSElasticBeanstalkRoleECS: politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di taggare le risorse in seguito alla creazione per Amazon ECS. Per ulteriori informazioni, consulta Policy per l'integrazione con altri servizi.	23 marzo 2023
AdministratorAccess-AWSElastic Beanstalk: politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di taggare le risorse in seguito alla creazione per Amazon ECS. Per ulteriori informazioni, consulta Gestione delle policy utente Elastic Beanstalk.	23 marzo 2023

Modifica	Descrizione	Data
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy —Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di aggiungere tag alle risorse Amazon ECS durante la loro creazione. Per ulteriori informazioni, consulta Autorizzazioni dei ruoli collegati ai servizi per Elastic Beanstalk.	23 marzo 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di aggiungere tag alle risorse Amazon ECS durante la loro creazione. Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	23 marzo 2023
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di aggiungere tag ai gruppi con scalabilità automatica durante la loro creazione. Per ulteriori informazioni, consulta Ruolo collegato ai servizi di aggiornamenti gestiti..	27 gennaio 2023

Modifica	Descrizione	Data
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di aggiungere tag per la creazione di un gruppo di scalabilità automatica (Auto Scaling group, ASG). Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	23 gennaio 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di aggiungere tag per la creazione di un Elastic Load Balancing (ELB). Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	21 dicembre 2022

Modifica	Descrizione	Data
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy—Politica esistente aggiornata	Le autorizzazioni sono state aggiunte a questa policy per consentire a Elastic Beanstalk di completare le seguenti operazioni durante gli aggiornamenti gestiti: • Crea ed elimina modelli di avvio e versioni dei modelli. • Avvia le istanze Amazon EC2 con i modelli di avvio. • Se è presente un Amazon RDS, recupera un elenco dei motori DB disponibili e le informazioni sulle istanze RDS con provisioning. Per ulteriori informazioni, consulta Ruolo collegato ai servizi di aggiornamenti gestiti..	23 agosto 2022

Modifica	Descrizione	Data
AWSElasticBeanstalkReadOnlyAccess: obsoleta GovCloud (STATI UNITI) Regione AWS	Questa policy è stata sostituita da `AWSElasticBeanstalkReadOnly` . Questa politica verrà gradualmente eliminata negli GovCloud (Stati Uniti). Regione AWS Quando questa policy verrà eliminata gradualmente, non sarà più disponibile per essere collegata a nuovi utenti, gruppi o ruoli IAM dopo il 17 giugno 2021. Per ulteriori informazioni consulta Policy utente.	17 giugno 2021
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy—Politica esistente aggiornata	Questa policy è stata aggiornata per consentire a Elastic Beanstalk di leggere gli attributi per le zone di disponibilità EC2. Consente a Elastic Beanstalk di fornire una convalida più efficace della selezione del tipo di istanza nelle zone di disponibilità. Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	16 giugno 2021

Modifica	Descrizione	Data
AWSElasticBeanstalkFullAccess: obsoleta GovCloud (STATI UNITI) Regione AWS	Questa policy è stata sostituita da `AdministratorAccess-AWSElasticBeanstalk` . Questa politica verrà gradualmente eliminata negli GovCloud (Stati Uniti). Regione AWS Quando questa policy verrà eliminata gradualmente, non sarà più disponibile per essere collegata a nuovi utenti, gruppi o ruoli IAM dopo il 10 giugno 2021. Per ulteriori informazioni consulta Policy utente.	10 giugno 2021

Modifica	Descrizione	Data
Le seguenti politiche gestite sono state dichiarate obsolete in tutta la Cina: Regione AWS • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess	La policy `AWSElasticBeanstalkFullAccess` è stata sostituita da `AdministratorAccess-AWSElasticBeanstalk`. La policy `AWSElasticBeanstalkReadOnlyAccess` è stata sostituita da `AWSElasticBeanstalkReadOnly`. Queste politiche sono state gradualmente eliminate in tutta la Cina Regione AWS. Queste policy non saranno più disponibili per essere collegate a nuovi utenti, gruppi o ruoli IAM dopo il 3 giugno 2021. Per ulteriori informazioni consulta Policy utente.	3 giugno 2021

Modifica	Descrizione	Data
AWSElasticBeanstalkService: obsoleta	Questa policy è stata sostituita da `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` . Questa policy verrà eliminata gradualmente, non sarà più disponibile per essere collegata a nuovi utenti, gruppi o ruoli IAM. Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	Giugno 2021 - Gennaio 2022

Modifica	Descrizione	Data
Le seguenti politiche gestite sono state obsolete in tutti Regione AWS i paesi, ad eccezione di Cina e GovCloud (Stati Uniti): • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess	La policy `AWSElasticBeanstalkFullAccess` è stata sostituita da `AdministratorAccess-AWSElasticBeanstalk`. La policy `AWSElasticBeanstalkReadOnlyAccess` è stata sostituita da `AWSElasticBeanstalkReadOnly`. Queste politiche sono state gradualmente eliminate in tutti gli Regione AWS anni S, ad eccezione di Cina e GovCloud (Stati Uniti). Queste policy non saranno più disponibili per essere collegate a nuovi utenti, gruppi o ruoli IAM dopo il 16 aprile 2021. Per ulteriori informazioni consulta Policy utente.	16 aprile 2021

Modifica	Descrizione	Data
Sono state aggiornate le seguenti policy gestite: • AdministratorAccess-Beanstalk AWSElastic • AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	Entrambe queste politiche ora supportano le PassRole autorizzazioni in Cina Regioni AWS. Per ulteriori informazioni su `AdministratorAccess-AWSElasticBeanstalk`, consulta Policy utente. Per ulteriori informazioni su `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`, consulta Policy gestite dei ruoli dei servizi.	9 marzo 2021
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy: nuova policy	Elastic Beanstalk ha aggiunto una nuova policy per sostituire la policy gestita `AWSElasticBeanstalkService`. Questa nuova policy gestita migliora la sicurezza delle risorse applicando un set di autorizzazioni più restrittivo. Per ulteriori informazioni, consulta Policy del ruolo del servizio gestito.	3 marzo 2021
Elastic Beanstalk ha iniziato a monitorare le modifiche	Elastic Beanstalk ha iniziato a tracciare le modifiche per le policy gestite. AWS	1° marzo 2021

Registrazione e monitoraggio in Elastic Beanstalk

AWS fornisce diversi strumenti per monitorare le risorse Elastic Beanstalk e rispondere a potenziali incidenti. Il monitoraggio è importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle vostre soluzioni. AWS Elastic Beanstalk AWS È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica uno.

Per ulteriori informazioni sul monitoraggio, consulta Ambienti di monitoraggio in Elastic Beanstalk.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Reporting sullo stato migliorato

Il reporting sullo stato migliorato è una caratteristica che puoi abilitare nel tuo ambiente per consentire a Elastic Beanstalk di raccogliere informazioni aggiuntive sulle risorse del tuo ambiente. Elastic Beanstalk analizza le informazioni per offrire un quadro più chiaro dello stato dell'intero ambiente e per facilitare l'identificazione di problemi che possono causare la non disponibilità della tua applicazione. Per ulteriori informazioni, consulta Reportistica e monitoraggio dello stato di salute migliorati in Elastic Beanstalk.

Registri delle EC2 istanze di Amazon

Le EC2 istanze Amazon nel tuo ambiente Elastic Beanstalk generano log che puoi visualizzare per risolvere problemi relativi all'applicazione o ai file di configurazione. I log creati dal server Web, dal server delle applicazioni, dagli script CloudFormation della piattaforma Elastic Beanstalk e vengono archiviati localmente su singole istanze. Puoi recuperarle con facilità utilizzando la console di gestione dell'ambiente o EB CLI. Puoi anche configurare il tuo ambiente per lo streaming dei log su Amazon CloudWatch Logs in tempo reale. Per ulteriori informazioni, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Notifiche per l'ambiente

Puoi configurare il tuo ambiente Elastic Beanstalk per utilizzare Amazon Simple Notification Service (Amazon SNS) per notificarti eventi importanti che influiscono sulla tua applicazione. Specificate un indirizzo e-mail durante o dopo la creazione dell'ambiente per ricevere e-mail da AWS quando si verifica un errore o quando lo stato dell'ambiente cambia. Per ulteriori informazioni, consulta Notifiche dell'ambiente Elastic Beanstalk con Amazon SNS.

CloudWatch Allarmi Amazon

Utilizzando gli CloudWatch allarmi, osservi una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o una policy di Amazon SNS. AWS Auto Scaling CloudWatch gli allarmi non richiamano azioni perché si trovano in uno stato particolare. Al contrario, gli allarmi richiamano operazioni quando lo stato è cambiato ed è stato mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta Utilizzo di Elastic Beanstalk con Amazon CloudWatch.

AWS CloudTrail registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Elastic Beanstalk. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a Elastic Beanstalk, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta Registrazione delle chiamate API Elastic Beanstalk con AWS CloudTrail.

AWS X-Ray debug

X-Ray è un AWS servizio che raccoglie dati sulle richieste servite dall'applicazione e li utilizza per creare una mappa dei servizi che è possibile utilizzare per identificare problemi con l'applicazione e opportunità di ottimizzazione. È possibile utilizzare la AWS Elastic Beanstalk console o un file di configurazione per eseguire il demone X-Ray sulle istanze del proprio ambiente. Per ulteriori informazioni, consulta Configurazione del debug AWS X-Ray.

Convalida della conformità per Elastic Beanstalk

La sicurezza e la conformità di AWS Elastic Beanstalk vengono valutate da revisori di terze parti nell'ambito di più programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri. AWS fornisce un elenco frequentemente aggiornato di servizi nell'ambito di specifici programmi di conformità nella pagina AWS Services in Scope by Compliance Program AWS .

I report di audit di terze parti possono essere scaricati utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact.

Per ulteriori informazioni sui programmi di AWS conformità, vedere Programmi di AWS conformità.

La responsabilità di conformità durante l'utilizzo di Elastic Beanstalk è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'organizzazione e dalle leggi e normative applicabili. Se l'uso

di Elastic Beanstalk è soggetto alla conformità a standard come HIPAA, PCI o AWS FedRAMP, fornisce risorse per aiutarti a:

- Guide introduttive su sicurezza e conformità: guide all'implementazione che illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e sulla conformità. AWS
- Architecting for HIPAA Security and Compliance on Amazon Web Services — Un white paper che descrive in che modo le aziende possono utilizzare AWS per creare applicazioni conformi allo standard HIPAA.
- AWS Risorse sulla conformità: una raccolta di cartelle di lavoro e guide sulla conformità che potrebbero riguardare il tuo settore e la tua area geografica.
- AWS Config Questo servizio valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti industriali.
- AWS Security Hub CSPM— Una panoramica completa dello stato di sicurezza AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Resilienza in Elastic Beanstalk

AWS Elastic Beanstalk gestisce e automatizza l'uso dell'infrastruttura AWS globale per tuo conto. Quando si utilizza Elastic Beanstalk, si traggono vantaggio dalla disponibilità e dai meccanismi di tolleranza ai guasti offerti. AWS

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità.

AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti.

Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Sicurezza dell'infrastruttura in Elastic Beanstalk

In quanto servizio gestito, AWS Elastic Beanstalk è protetto dalle procedure di sicurezza di rete AWS globali descritte nel nostro sito Web sulle migliori pratiche per la sicurezza, l'identità e la conformità.

Utilizzi chiamate API AWS pubblicate per accedere a Elastic Beanstalk attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. I client devono inoltre supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte delle piattaforme moderne supporta queste modalità.

Inoltre, le richieste devono essere firmate utilizzando un chiave di accesso ID e una chiave di accesso segreta associata a un account principale IAM. In alternativa è possibile utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Analisi della configurazione e delle vulnerabilità in Elastic Beanstalk

AWS e i nostri clienti condividono la responsabilità di raggiungere un elevato livello di sicurezza e conformità dei componenti software. AWS Elastic Beanstalk ti aiuta a fare la tua parte del modello di responsabilità condivisa fornendo una funzionalità di aggiornamento gestita. Questa caratteristica applica automaticamente aggiornamenti secondari e patch per una versione della piattaforma supportata da Elastic Beanstalk.

Per ulteriori informazioni, consulta Modello di responsabilità condivisa per la manutenzione della piattaforma Elastic Beanstalk.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Best practice relative alla sicurezza di Elastic Beanstalk

AWS Elastic Beanstalk offre diverse funzionalità di sicurezza da considerare durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per il tuo ambiente, gestiscile come considerazioni utili anziché prescrizioni.

Per altri argomenti sulla sicurezza di Elastic Beanstalk, consulta AWS Elastic Beanstalk sicurezza.

Best practice relative alla sicurezza preventiva

I controlli di sicurezza preventivi tentano di prevenire gli incidenti prima che si verifichino.

Implementazione dell'accesso con privilegi minimi

Elastic AWS Identity and Access Management Beanstalk fornisce policy gestite (IAM) perprofili di esempio, ruoli di servizio e utenti IAM. Queste policy gestite specificano tutte le autorizzazioni che potrebbero essere necessarie per il corretto funzionamento dell'ambiente e dell'applicazione.

La tua applicazione potrebbe non richiedere tutte le autorizzazioni nelle nostre policy gestite. Puoi personalizzarle e concedere solo le autorizzazioni necessarie per le istanze del tuo ambiente, il servizio Elastic Beanstalk e gli utenti per eseguire le loro attività. Ciò è particolarmente rilevante per le policy utente, dove ruoli utente diversi potrebbero avere esigenze di autorizzazione diverse. L'applicazione dell'accesso con privilegio minimo è fondamentale per ridurre i rischi di sicurezza e l'impatto risultante da errori o intenzioni dannose.

Proteggi i dati sensibili delle applicazioni

Quando l'applicazione deve accedere a informazioni sensibili come credenziali, chiavi API o dati di configurazione, segui queste pratiche per mantenere la sicurezza:

- Recupera i dati sensibili direttamente dal Gestione dei segreti AWS nostro AWS Systems Manager Parameter Store utilizzando il rispettivo codice SDKs o APIs nel codice dell'applicazione. Ciò fornisce il modo più sicuro e flessibile per accedere alle informazioni sensibili.
- Se trasmetti dati sensibili dal Gestione dei segreti AWS nostro AWS Systems Manager Parameter Store come variabili di ambiente (vediRecupera i segreti nelle variabili di ambiente), limita attentamente l'accesso alle coppie di EC2 chiavi e configura i ruoli IAM appropriati con i permessi con privilegi minimi per le tue istanze.
- Non stampare, registrare o esporre dati sensibili nel codice dell'applicazione, poiché questi valori potrebbero finire nei file di registro o nei messaggi di errore che potrebbero essere visibili agli utenti non autorizzati.

Aggiorna regolarmente le tue piattaforme

Elastic Beanstalk rilascia regolarmente nuove versioni della piattaforma per aggiornare tutte le sue piattaforme. Le nuove versioni della piattaforma forniscono aggiornamenti del sistema operativo, del runtime, del server applicazioni e del server Web e aggiornamenti ai componenti Elastic

Beanstalk. Molti di questi aggiornamenti della piattaforma includono importanti correzioni di sicurezza. Assicurati che gli ambienti Elastic Beanstalk siano in esecuzione su una versione della piattaforma supportata (in genere la versione più recente per la tua piattaforma). Per informazioni dettagliate, vedi Aggiornamento della versione della piattaforma dell'ambiente Elastic Beanstalk.

Il modo più semplice per mantenere aggiornata la piattaforma del tuo ambiente è configurare l'ambiente in modo che utilizzi gli aggiornamenti gestiti della piattaforma.

Applica su istanze di ambiente IMDSv2

Le istanze Amazon Elastic Compute Cloud (Amazon EC2) nei tuoi ambienti Elastic Beanstalk utilizzano il servizio di metadati delle istanze (IMDS), un componente su istanza, per accedere in modo sicuro ai metadati delle istanze. IMDS supporta due metodi per accedere ai dati: e. IMDSv1 IMDSv2 IMDSv2 utilizza richieste orientate alla sessione e mitiga diversi tipi di vulnerabilità che potrebbero essere utilizzate per tentare di accedere all'IMDS. Per i dettagli sui vantaggi di IMDSv2, consulta i miglioramenti per aggiungere una difesa approfondita all'Instance Metadata Service. EC2

IMDSv2 è più sicuro, quindi è una buona idea imporne l'uso sulle istanze IMDSv2 . Per applicarlo IMDSv2, assicurati che tutti i componenti dell'applicazione siano supportati IMDSv2, quindi disattivali. IMDSv1 Per ulteriori informazioni, consulta the section called “IMDS”.

Best practice relative alla sicurezza di rilevamento

I controlli di sicurezza di rilevamento identificano le violazioni della sicurezza dopo che si sono verificate. Possono aiutarti a rilevare potenziali minacce o incidenti alla sicurezza.

Implementazione del monitoraggio

Il monitoraggio è una parte importante per mantenere l'affidabilità, la sicurezza, la disponibilità e le prestazioni delle soluzioni Elastic Beanstalk. AWS fornisce diversi strumenti e servizi per aiutarti a monitorare i tuoi servizi. AWS

Di seguito sono elencati alcuni esempi di elementi da monitorare:

- CloudWatch Parametri Amazon per Elastic Beanstalk: imposta allarmi per i parametri chiave di Elastic Beanstalk e per i parametri personalizzati della tua applicazione. Per informazioni dettagliate, vedi Utilizzo di Elastic Beanstalk con Amazon CloudWatch.
- AWS CloudTrail voci: tieni traccia delle azioni che potrebbero influire sulla disponibilità, ad esempio o. `UpdateEnvironment` `TerminateEnvironment` Per informazioni dettagliate, vedi Registrazione delle chiamate API Elastic Beanstalk con AWS CloudTrail.

Abilita AWS Config

AWS Config fornisce una visualizzazione dettagliata della configurazione delle AWS risorse del tuo account. Puoi vedere in che modo le risorse sono correlate, ottenere una cronologia delle modifiche alla configurazione ed esaminare come cambiano le relazioni e le configurazioni nel tempo.

È possibile utilizzare AWS Config per definire regole che valutano le configurazioni delle risorse per la conformità dei dati. AWS Config le regole rappresentano le impostazioni di configurazione ideali per le tue risorse Elastic Beanstalk. Se una risorsa viola una regola e viene contrassegnata come non conforme, può AWS Config avvisarti utilizzando un argomento di Amazon Simple Notification Service (Amazon SNS). Per informazioni dettagliate, vedi Ricerca e tracciamento delle risorse Elastic Beanstalk con AWS Config.

Ruoli, profili di istanza e policy utente di Elastic Beanstalk Service

I ruoli sono entità create con AWS Identity and Access Management (IAM) per applicare le autorizzazioni. Esistono ruoli necessari per il corretto funzionamento dell'ambiente Elastic Beanstalk. Hai anche la possibilità di creare politiche e ruoli personalizzati che puoi assegnare a utenti o gruppi.

Ruoli richiesti per il tuo ambiente Elastic Beanstalk

Quando crei un ambiente, ti AWS Elastic Beanstalk chiede di fornire i seguenti ruoli AWS Identity and Access Management (IAM):

- Ruolo di servizio: Elastic Beanstalk assume un ruolo di servizio per utilizzarne altri per conto dell'utente. Servizi AWS
- Profilo dell'istanza Elastic Beanstalk applica un profilo di istanza alle istanze Amazon EC2 del tuo ambiente. Questa azione consente loro di eseguire le attività richieste, come il recupero di informazioni da Amazon Simple Storage Service (Amazon S3) e il caricamento dei log su S3.

Crea il ruolo di servizio e il ruolo del profilo dell'istanza EC2

Se il tuo AWS account non dispone di un profilo di istanza EC2 o di un ruolo di servizio, devi crearne uno per ciascuno utilizzando il servizio IAM. Puoi quindi assegnare il profilo e il ruolo di servizio dell'istanza EC2 ai nuovi ambienti che crei. La procedura guidata di creazione dell'ambiente ti guida al servizio IAM, in modo che tu possa creare questi ruoli con le autorizzazioni richieste.

Politiche e ruoli opzionali per gestire l'ambiente Elastic Beanstalk

Facoltativamente, puoi creare policy utente e applicarle agli utenti e ai gruppi IAM del tuo account. In questo modo gli utenti possono creare e gestire applicazioni e ambienti Elastic Beanstalk. Puoi anche assegnare policy gestite da Elastic Beanstalk per l'accesso completo e in sola lettura a utenti o gruppi. Per ulteriori informazioni su questo tipo di policy, consulta the section called “Policy utente”.

Puoi creare i tuoi profili di istanza e le policy utente per scenari avanzati. Se le tue istanze devono accedere ai servizi che non sono inclusi nelle policy di default, puoi creare una nuova policy o aggiungerne di altre a quella di default. Inoltre, puoi creare policy utente più restrittive se la policy

gestita è troppo permissiva. Per ulteriori informazioni sulle AWS autorizzazioni, consulta la IAM User Guide.

Ruolo di servizio Elastic Beanstalk

Un ruolo di servizio è il ruolo IAM che la console Elastic Beanstalk assume quando chiama altri servizi per conto dell'utente. Ad esempio, Elastic Beanstalk utilizza un ruolo di servizio quando chiama Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing e Amazon EC2 Auto Scaling per raccogliere informazioni. APIs Il ruolo di servizio utilizzato da Elastic Beanstalk è quello specificato quando viene creato l'ambiente Elastic Beanstalk.

Sono presenti due policy gestite collegate al ruolo di servizio: Queste policy forniscono le autorizzazioni che consentono a Elastic Beanstalk di accedere alle risorse necessarie per creare e gestire AWS i tuoi ambienti. Una policy gestita fornisce le autorizzazioni per il monitoraggio avanzato dell'integrità e il supporto Amazon SQS del livello worker, mentre l'altra fornisce le autorizzazioni aggiuntive necessarie per gli aggiornamenti gestiti della piattaforma.

`AWSElasticBeanstalkEnhancedHealth`

Questa policy concede le autorizzazioni a Elastic Beanstalk per monitorare l'integrità dell'istanza e dell'ambiente. Inoltre include anche operazioni Amazon SQS per consentire a Elastic Beanstalk di monitorare le attività di coda per gli ambienti worker. Per visualizzare il contenuto di questa policy gestita, consulta la AWSElasticBeanstalkEnhancedHealthpagina della Managed Policy Reference AWS Guide.

`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Questa policy concede le autorizzazioni a Elastic Beanstalk per aggiornare gli ambienti per tuo conto ed eseguire gli aggiornamenti gestiti della piattaforma. Per visualizzare il contenuto di questa policy gestita, consulta la AWSElasticBeanstalkManagedUpdatesCustomerRolePolicypagina della AWS Managed Policy Reference Guide.

Raggruppamenti di autorizzazioni a livello di servizio

Questa policy è raggruppata in istruzioni in base al set di autorizzazioni fornite.

- *`ElasticBeanstalkPermissions`*— Questo gruppo di autorizzazioni serve a richiamare le azioni del servizio Elastic Beanstalk (Elastic Beanstalk). APIs

- *`AllowPassRoleToElasticBeanstalkAndDownstreamServices`*: questo gruppo di autorizzazioni consente di passare qualsiasi ruolo a Elastic Beanstalk e ad altri servizi downstream come CloudFormation.
- *`ReadOnlyPermissions`*: questo gruppo di autorizzazioni serve a raccogliere informazioni sull'ambiente in esecuzione.
- *`*OperationPermissions`*: i gruppi con questo modello di denominazione servono a chiamare le operazioni necessarie per eseguire gli aggiornamenti della piattaforma.
- *`*BroadOperationPermissions`*: i gruppi con questo modello di denominazione servono a chiamare le operazioni necessarie per eseguire gli aggiornamenti della piattaforma. Includono inoltre autorizzazioni estese per il supporto degli ambienti legacy.
- *`*TagResource`*— I gruppi con questo modello di denominazione sono destinati alle chiamate che utilizzano per tag-on-create APIs allegare tag alle risorse che vengono create in un ambiente Elastic Beanstalk.

Puoi creare un ambiente Elastic Beanstalk con uno dei seguenti approcci. Ogni sezione descrive come l'approccio gestisce il ruolo di servizio.

Console Elastic Beanstalk

Se crei un ambiente utilizzando la console Elastic Beanstalk, Elastic Beanstalk ti chiederà di creare un ruolo di servizio denominato `aws-elasticbeanstalk-service-role`. Quando creato tramite Elastic Beanstalk, questo ruolo include una policy di attendibilità che consenta a Elastic Beanstalk di assumere il ruolo di servizio. Al ruolo sono allegate anche le due policy gestite descritte in precedenza in questo argomento.

Interfaccia a riga di comando di Elastic Beanstalk (CLI EB)

È possibile creare un ambiente utilizzando il comando the section called “eb create” dell'Interfaccia della linea di comando Elastic Beanstalk (EB CLI). Se non si specifica un ruolo di servizio attraverso l'opzione `--service-role`. Elastic Beanstalk crea lo stesso ruolo di servizio predefinito `aws-elasticbeanstalk-service-role`. Se il ruolo di servizio predefinito esiste già, Elastic Beanstalk lo utilizza per il nuovo ambiente. Quando creato tramite Elastic Beanstalk, questo ruolo include una policy di attendibilità che consenta a Elastic Beanstalk di assumere il ruolo di servizio. Al ruolo sono allegate anche le due policy gestite descritte in precedenza in questo argomento.

API di Elastic Beanstalk

È possibile creare un ambiente utilizzando l'operazione `CreateEnvironment` dell'API Elastic Beanstalk. Se non si specifica un ruolo di servizio, Elastic Beanstalk crea un ruolo collegato al servizio di monitoraggio. Si tratta di un tipo unico di ruolo di servizio predefinito da Elastic Beanstalk per includere tutte le autorizzazioni richieste dal servizio per chiamare altri utenti per conto dell'utente. Servizi AWS Il ruolo collegato al servizio è associato al tuo account. Elastic Beanstalk lo crea una volta, quindi lo riutilizza quando crea altri ambienti. Puoi utilizzare IAM anche per creare in anticipo il ruolo collegato ai servizi di monitoraggio del tuo account. Quando l'account dispone di un ruolo collegato al servizio di monitoraggio, è possibile utilizzarlo per creare un ambiente tramite la console Elastic Beanstalk, l'API Elastic Beanstalk o la EB CLI. Per le istruzioni su come utilizzare i ruoli collegati ai servizi con ambienti Elastic Beanstalk, consultaUtilizzo dei ruoli collegati ai servizi per Elastic Beanstalk.

Per ulteriori informazioni sui ruoli di servizio, consulta Gestione dei ruoli del servizio Elastic Beanstalk.

Profilo dell'istanza Elastic Beanstalk

Un profilo dell'istanza è un ruolo IAM che viene applicato alle istanze Amazon EC2 avviate nell'ambiente Elastic Beanstalk. Quando si crea un ambiente Elastic Beanstalk, specifica il profilo dell'istanza utilizzato quando le istanze EC2 completano le seguenti operazioni:

- Recupero delle versioni delle applicazioni da Amazon Simple Storage Service (Amazon S3)
- Scrittura di log su Amazon S3
- Negli ambienti integrati AWS X-Ray, carica i dati di debug su X-Ray
- Negli ambienti Docker gestiti da Amazon ECS, coordina le implementazioni dei container con Amazon Elastic Container Service (Amazon ECS)
- Negli ambienti worker (operatore), lettura da una coda Amazon Simple Queue Service (Amazon SQS)
- Negli ambienti worker (operatore), esecuzione dell'elezione del leader con Amazon DynamoDB
- Negli ambienti di lavoro, pubblica i parametri relativi allo stato delle istanze su Amazon CloudWatch

Policy gestite

Elastic Beanstalk fornisce una serie di policy gestite che consentono alle istanze EC2 presenti nell'ambiente di eseguire le operazioni richieste. Di seguito sono elencate le policy gestite richieste per i casi d'uso più comuni.

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Se la tua applicazione web richiede l'accesso ad altri servizi aggiuntivi Servizi AWS, aggiungi istruzioni o politiche gestite al profilo dell'istanza che consentono l'accesso a tali servizi. Per ulteriori informazioni, consulta Aggiunta delle autorizzazioni al profilo dell'istanza predefinito.

Creazione di un profilo di istanza EC2

Se il tuo AWS account non dispone di un profilo di istanza EC2, devi crearne uno utilizzando il servizio IAM. Potrai quindi assegnare il profilo dell'istanza EC2 ai nuovi ambienti che vengono creati. I passaggi di creazione dell'ambiente nella console Elastic Beanstalk forniscono l'accesso alla console IAM, in modo da poter creare un profilo di istanza EC2 con le autorizzazioni richieste.

Puoi anche creare un profilo di istanza EC2 accedendo direttamente alla console IAM, senza passare dalla console Elastic Beanstalk. Per i passaggi dettagliati per creare un profilo di istanza Elastic Beanstalk EC2 nella console IAM, consulta. Creazione di un profilo dell'istanza

Policy utente Elastic Beanstalk

Crea utenti IAM per ogni utente che utilizza Elastic Beanstalk per evitare di utilizzare il tuo account root o di condividere le credenziali. Come best practice di sicurezza, concedi a questi utenti solo le autorizzazioni per accedere ai servizi e alle funzionalità di cui hanno bisogno.

Elastic Beanstalk richiede le autorizzazioni non solo per le proprie operazioni API, ma anche per molti altri servizi AWS . Elastic Beanstalk utilizza le autorizzazioni utente per avviare le risorse in un ambiente. Queste risorse includono le istanze EC2, un load balancer Elastic Load Balancing e un gruppo Auto Scaling. Elastic Beanstalk utilizza anche le autorizzazioni degli utenti per salvare log e modelli su Amazon Simple Storage Service (Amazon S3), inviare notifiche ad Amazon SNS, assegnare profili di istanza e pubblicare metriche a. CloudWatch Elastic CloudFormation Beanstalk richiede le autorizzazioni per orchestrare le distribuzioni e gli aggiornamenti delle risorse. Richiede inoltre le autorizzazioni Amazon RDS per creare database, se necessari, e le autorizzazioni Amazon SQS per creare le code per gli ambienti worker (operatore).

Per ulteriori informazioni sulle policy utente, consulta Gestione delle policy utente Elastic Beanstalk.

Esempi e tutorial

Tutorial specifici per linguaggio e framework sono distribuiti nella Guida per gli sviluppatori. AWS Elastic Beanstalk A questo elenco vengono aggiunti tutorial nuovi e aggiornati nel momento in cui vengono pubblicati. Gli aggiornamenti più recenti vengono mostrati per primi.

Questi tutorial sono rivolti a utenti intermedi e potrebbero non contenere le istruzioni per i passaggi di base, ad esempio la procedura di registrazione ad AWS. Se è la prima volta che usi AWS Elastic Beanstalk, dai un'occhiata alla guida introduttiva per rendere operativo il tuo primo ambiente Elastic Beanstalk.

- Ruby on Rails: Distribuzione di un'applicazione Rails su Elastic Beanstalk
- Ruby e Sinatra: Distribuzione di un'applicazione Sinatra su Elastic Beanstalk
- Configurazione di PHP e MySQL HA: Distribuzione di un'applicazione PHP a elevata disponibilità con un database Amazon RDS esterno in Elastic Beanstalk
- PHP e Laravel: Distribuzione di un'applicazione Laravel su Elastic Beanstalk
- PHP e CakePHP: Distribuzione di un'applicazione CakePHP su Elastic Beanstalk
- Configurazione di PHP e Drupal HA: Distribuzione di un sito Web Drupal a elevata disponibilità con un database Amazon RDS esterno in Elastic Beanstalk
- Configurazione PHP e HA - WordPress Implementazione di un WordPress sito Web ad alta disponibilità con un database Amazon RDS esterno su Elastic Beanstalk
- Configurazione di Node.js con DynamoDB HA: Distribuzione di un'applicazione Node.js con DynamoDB in Elastic Beanstalk
- ASP.NET Core: QuickStart: distribuzione di un' ASP.NET applicazione su Elastic Beanstalk
- Python e Flask: Implementazione di un'applicazione Flask in Elastic Beanstalk
- Python e Django: Distribuzione di un'applicazione Django in Elastic Beanstalk
- Node.js ed Express: Distribuzione di un'applicazione Node.js Express su Elastic Beanstalk
- Docker, PHP e nginx: Creazione di un ambiente Docker gestito da ECS con la console Elastic Beanstalk

È possibile scaricare le applicazioni di esempio utilizzate da Elastic Beanstalk al momento della creazione di un ambiente senza fornire un bundle di origine con i seguenti collegamenti:

- Docker: docker.zip

- Docker multicontainer — 2.zip docker-multicontainer-v
- Docker preconfigurato (Glassfish) — 1.zip docker-glassfish-v
- Go: go.zip
- Corretto: corretto.zip
- Tomcat: tomcat.zip
- .NET Core su Linux — .zip dotnet-core-linux
- .NET Core — dotnet-asp-windows .zip
- Node.js: nodejs.zip
- PHP: php.zip
- Python: python.zip
- Ruby: ruby.zip

Altre applicazioni di esempio più complesse che mostrano l'uso di framework, librerie e strumenti web aggiuntivi sono disponibili come progetti open source su: GitHub

- Load-balanced WordPress (tutorial): file di configurazione per l'installazione WordPress sicura e l'esecuzione in un ambiente Elastic Beanstalk con carico bilanciato.
- Drupal con bilanciamento del carico (tutorial): configurazione dei file e istruzioni per l'installazione di Drupal in modo sicuro e in esecuzione in un ambiente Elastic Beanstalk con bilanciamento del carico.
- Scorekeep: API RESTful web che utilizza il framework Spring e fornisce un'interfaccia AWS SDK per Java per la creazione e la gestione di utenti, sessioni e giochi. L'API è integrata con un'app Web Angular 1.5 che consuma l'API sull'HTTP. Include filiali che mostrano l'integrazione con Amazon Cognito e Amazon Relational Database Service. AWS X-Ray

  L'applicazione utilizza caratteristiche della piattaforma Java SE per scaricare dipendenze e creare on-instance, riducendo al minimo le dimensioni del bundle di origine. L'applicazione include anche file di configurazione nginx che sovrascrivono la configurazione predefinita per servire l'app Web di front-end in modo statico sulla porta 80 tramite il proxy, e instradare le richieste su percorsi in `/api` all'API in esecuzione su `localhost:5000`.
- Ha serpenti? - Applicazione Tomcat che mostra l'uso di RDS in un'applicazione web Java EE in Elastic Beanstalk. Il progetto mostra l'uso dei file di configurazione Servlets JSPs, Simple Tag Support, Tag Files, JDBC, SQL, Log4J, Bootstrap, Jackson ed Elastic Beanstalk.

- Locust Load Generator : questo progetto mostra l'uso delle caratteristiche della piattaforma Java SE per installare ed eseguire Locust, uno strumento di generazione del carico scritto in Python. Il progetto include i file di configurazione in grado di installare e configurare Locust, uno script di build che consente di configurare una tabella DynamoDB e un Procfile che esegue Locust.
- Condividi i tuoi commenti (tutorial): applicazione PHP che mostra l'uso di MySQL su Amazon RDS e Composer e i file di configurazione.
- A New Startup (tutorial) - Applicazione di esempio Node.js che mostra l'uso di DynamoDB, AWS l'SDK JavaScript per Node.js, la gestione dei pacchetti npm e i file di configurazione.

Migrazione delle applicazioni IIS su Elastic Beanstalk

AWS Elastic Beanstalk fornisce un percorso di migrazione semplificato per le applicazioni Windows in esecuzione su Internet Information Services (IIS). La funzionalità di migrazione descritta in questo capitolo riduce in modo significativo il tempo e la complessità tipicamente associati alle migrazioni cloud, aiutando a mantenere la funzionalità dell'applicazione e l'integrità della configurazione durante la transizione a. AWS

L'operazione eb migrate

Usa il eb migrate comando nell'interfaccia a riga di comando di Elastic Beanstalk (EB CLI) per scoprire, impacchettare e distribuire automaticamente le tue applicazioni IIS su. Cloud AWS Il processo mantiene la funzionalità dell'applicazione e preserva le configurazioni, inclusi i binding, i pool di applicazioni e le impostazioni di autenticazione.

I passaggi seguenti riassumono il processo eseguito dall'`eb migrate`operazione per la transizione dell'applicazione a: Cloud AWS

1. Scopri i siti IIS e le relative configurazioni.
2. Contenuto e configurazione dell'applicazione del Package.
3. Crea l'ambiente e l'applicazione Elastic Beanstalk.
4. Distribuisci l'applicazione con impostazioni preservate.

Opzioni di esecuzione del flusso di lavoro e della posizione

Il eb migrate comando fornisce opzioni per flussi di lavoro di migrazione flessibili e posizioni di esecuzione. Per impostazione predefinita, esegui il comando sul server di destinazione che contiene l'applicazione che desideri migrare su Elastic Beanstalk. Se non riesci a eseguire i comandi direttamente sul server delle applicazioni, utilizza l'`remote`opzione per eseguire il comando da un host bastion che si connette al server di destinazione contenente l'applicazione e le configurazioni. Per completare la migrazione in due passaggi, puoi anche generare il pacchetto di migrazione senza distribuirlo utilizzando l'`archive-only`opzione e poi distribuirlo in un secondo momento a tuo piacimento utilizzando l'opzione. `archive`

Per informazioni di riferimento sul eb migrate comando, vedere. the section called “eb migrate”

Argomenti

I seguenti argomenti forniscono informazioni dettagliate sulla migrazione delle applicazioni IIS a Elastic Beanstalk:

- the section called “Prerequisiti”- Comprendi il software, l'accesso e le autorizzazioni necessari per migrare le applicazioni Windows negli ambienti. AWS Elastic Beanstalk
- the section called “Glossario sulla migrazione”- Scopri come i componenti IIS vengono mappati alle risorse Elastic Beanstalk
- the section called “Mappatura della migrazione da IIS a Elastic Beanstalk”- Scopri come i componenti IIS vengono mappati alle risorse Elastic Beanstalk
- the section called “Migrazioni IIS di base”- Scopri come eseguire migrazioni di base
- the section called “Scenari di migrazione avanzati”- Gestisci scenari di migrazione complessi
- the section called “Configurazione di sicurezza”- Configura le impostazioni di sicurezza durante la migrazione
- the section called “Configurazione della rete”- Gestisci le configurazioni di rete e porte
- the section called “Risoluzione dei problemi e diagnostica”- Risolvi i problemi di migrazione più comuni
- the section called “Opzioni di migrazione: EB CLI e MGN”- Confronta due opzioni di migrazione principali.

Prerequisiti

Prima di utilizzare il eb migrate comando, assicuratevi che l'ambiente soddisfi i seguenti requisiti:

Installazione e versione di IIS

Il server da cui si esegue la migrazione deve eseguire Internet Information Services (IIS) versione 7.0 o successiva. IIS 10.0 su Windows Server 2016 o versione successiva offre l'ambiente più compatibile per la migrazione.

Per verificare la versione di IIS in uso, esegui il seguente comando:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\InetStp\"
...
SetupString             : IIS 10.0
VersionString           : Version 10.0
...
```

Requisiti di Windows Server

L'ambiente di origine deve eseguire Windows Server 2016 o versione successiva per una compatibilità ottimale. Elastic Beanstalk supporta le seguenti versioni di Windows Server come piattaforme di destinazione:

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

Installazione dell'interfaccia CLI EB

- Flusso di lavoro predefinito (senza l'`--remote`opzione):
  - Python e l'Elastic Beanstalk Command Line Interface (EB CLI) devono essere installati sul server che contiene l'applicazione che desideri migrare su Elastic Beanstalk. Sebbene non sia necessario, consigliamo di installare l'EB CLI all'interno di una `virtualenv` sandbox come descritto in. the section called “Installa in virtualenv”
- Utilizzando l'opzione: `--remote`
  - Python e l'Elastic Beanstalk Command Line Interface (EB CLI) devono essere installati sul tuo host bastion. Sebbene non sia necessario, consigliamo di installare l'EB CLI all'interno di una `virtualenv` sandbox come descritto in. the section called “Installa in virtualenv”

Autorizzazioni richieste

Sono necessarie le seguenti credenziali e autorizzazioni:

- Privilegi di amministratore sul server IIS di origine o sull'host bastion (se si utilizza l'opzione). `--remote`
- AWS credenziali con autorizzazioni per creare e gestire risorse Elastic Beanstalk

Web Deploy 3.6

Lo strumento Microsoft Web Deploy (versione 3.6 o successiva) deve essere installato sul server di origine o sull'host bastion (se si utilizza l'`--remote`opzione). Questo strumento viene utilizzato da per impacchettare eb migrate le applicazioni.

Per verificare l'installazione esegui il seguente comando:

:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\IIS
 Extensions\MSDeploy\3" -Name InstallPath

InstallPath  : C:\Program Files\IIS\Microsoft Web Deploy V3\
...
```

Per istruzioni di installazione, vedere Installazione e configurazione di Web Deploy su IIS 8.0 o versione successiva nel sito Web della documentazione del prodotto Microsoft Windows.

Requisiti di rete

- Flusso di lavoro predefinito (senza l'`--remote`opzione):
  - Il server di origine deve disporre dell'accesso Internet in uscita ai AWS servizi.
- Utilizzando l'`--remote`opzione:
  - Il server di origine deve disporre dell'accesso Internet in uscita ai AWS servizi.
  - Configura le corrette regole di accesso ai gruppi di sicurezza che consentano una connessione di rete in uscita dal tuo host bastion e una connessione in entrata alla macchina remota. Assicurati che l'IP dell'host bastion sia inserito nell'elenco consentito tramite TCP sulla porta 22 per accedere alla macchina remota.
  - Assicurati che il tuo client SSH sia installato e in esecuzione sulla tua macchina remota e sul tuo host bastion.
  - Assicurati che la configurazione del firewall contenga le regole appropriate che aprano la porta 22 o consentano la connessione al client.
  - Verifica la tua connessione effettuando manualmente l'accesso tramite SSH all'host remoto dall'host bastion prima di tentare la migrazione.

Glossario sulla migrazione

Questo glossario fornisce le definizioni dei termini e dei concetti chiave relativi a IIS, Elastic Beanstalk e alla migrazione delle applicazioni IIS verso Elastic Beanstalk.

Termini per Windows, IIS e.NET

IIS

Internet Information Services, un software server Web sviluppato da Microsoft per l'utilizzo con Windows Server. IIS ospita siti Web, applicazioni Web e servizi Web, fornendo una piattaforma

per l'esecuzione di ASP.NET e altre tecnologie Web. Durante la migrazione a Elastic Beanstalk, i siti IIS e le relative configurazioni vengono impacchettati e distribuiti su istanze di Windows Server nel cloud. AWS

Le versioni IIS 7.0 e successive sono supportate per la migrazione, con IIS 10.0 su Windows Server 2016 o versioni successive che fornisce l'ambiente più compatibile.

.NET Framework

Una piattaforma di sviluppo software sviluppata da Microsoft per la creazione e l'esecuzione di applicazioni Windows. Fornisce un'ampia libreria di classi denominata Framework Class Library (FCL) e supporta l'interoperabilità linguistica tra diversi linguaggi di programmazione.

Durante la migrazione a Elastic Beanstalk, le applicazioni create su.NET Framework continuano a funzionare sulla stessa versione del framework nell'ambiente cloud. Elastic Beanstalk supporta più versioni di .NET Framework (4.x) sulle sue piattaforme Windows Server.

.NET Core

Un successore open source multipiattaforma di .NET Framework, progettato per essere più modulare e leggero. .NET Core (ora chiamato semplicemente .NET 5 e versioni successive) consente agli sviluppatori di creare applicazioni che funzionano su Windows, Linux e macOS.

Durante la migrazione di applicazioni basate su.NET Core su Elastic Beanstalk, puoi scegliere tra piattaforme Windows Server o piattaforme basate su Linux, a seconda dei requisiti e delle dipendenze dell'applicazione.

Common Language Runtime (CLR)

Il componente della macchina virtuale di.NET Framework che gestisce l'esecuzione di programmi.NET. Il CLR fornisce servizi come la gestione della memoria, la sicurezza dei tipi, la gestione delle eccezioni, la raccolta dei rifiuti e la gestione dei thread.

Durante la migrazione a Elastic Beanstalk, la versione CLR appropriata è automaticamente disponibile sulla piattaforma Windows Server selezionata, garantendo la compatibilità con i requisiti dell'applicazione.

Sito

Un contenitore logico in IIS che rappresenta un'applicazione o un servizio Web, identificato da un'associazione univoca di indirizzo IP, porta e intestazione host. Ogni sito IIS ha il proprio pool di applicazioni, associazioni e impostazioni di configurazione e può contenere una o più applicazioni.

Applicazione

Raggruppamento di file di contenuto e codice all'interno di un sito IIS che gestisce le richieste per uno spazio URL specifico. Le applicazioni in IIS possono avere impostazioni di configurazione proprie, che in genere sono archiviate in `web.config` file.

Durante la migrazione a Elastic Beanstalk, le applicazioni vengono preservate con la struttura del percorso e le impostazioni di configurazione. Il processo di migrazione garantisce che le applicazioni annidate mantengano la gerarchia e i percorsi URL nell'ambiente cloud.

ApplicationPool

Una funzionalità IIS che isola le applicazioni Web per una migliore sicurezza, affidabilità e gestione delle prestazioni. I pool di applicazioni definiscono l'ambiente di runtime per le applicazioni, tra cui la versione.NET Framework, la modalità pipeline e le impostazioni di identità.

VirtualDirectory

Una mappatura delle directory in IIS che consente di fornire il contenuto da una posizione esterna alla directory principale del sito. Le directory virtuali consentono di organizzare i contenuti in diverse posizioni fisiche, presentando al contempo una struttura di URL unificata agli utenti.

Durante la migrazione a Elastic Beanstalk, le directory virtuali vengono preservate con le relative mappature dei percorsi. Il eb migrate comando crea la struttura di directory e le configurazioni necessarie nell'ambiente cloud per mantenere gli stessi percorsi URL.

ARR

Application Request Routing, un'estensione IIS che fornisce funzionalità di bilanciamento del carico e invio di proxy per i server Web. ARR consente il routing basato su URL, l'inoltro delle richieste HTTP e la distribuzione del carico su più server.

Durante la migrazione a Elastic Beanstalk, le configurazioni ARR vengono preservate attraverso l'installazione di funzionalità ARR sulle istanze EC2 e la configurazione di regole di routing appropriate. Per scenari di routing complessi, il processo di migrazione può anche sfruttare le regole di Application Load Balancer per implementare funzionalità simili.

Riscrittura dell'URL

Un modulo IIS che modifica le richieste URLs in base a regole definite prima che raggiungano l'applicazione Web. URL Rewrite consente la manipolazione degli URL, il reindirizzamento e la distribuzione dei contenuti in base a modelli e condizioni.

Durante la migrazione a Elastic Beanstalk, le regole di riscrittura degli URL `web.config` dai file vengono tradotte in regole di routing ALB, ove possibile, o conservate nella configurazione IIS sulle istanze EC2. Ciò garantisce che i pattern e i reindirizzamenti degli URL continuino a funzionare come previsto nell'ambiente cloud.

msdeploy.exe

Uno strumento da riga di comando utilizzato per distribuire applicazioni Web e siti Web su server IIS. Conosciuto anche come Web Deploy, offre un modo per impacchettare, sincronizzare e distribuire applicazioni Web, siti Web e configurazioni di server.

Il eb migrate comando utilizza Web Deploy (versione 3.6 o successiva) per impacchettare le applicazioni durante la migrazione a Elastic Beanstalk. Questo strumento deve essere installato sul server di origine affinché il processo di migrazione funzioni correttamente.

Percorso fisico

La posizione effettiva del file system in cui sono archiviati i file di contenuto di un sito IIS o di un'applicazione. I percorsi fisici possono puntare a directory locali, condivisioni di rete o altre posizioni di archiviazione accessibili al server IIS.

Durante la migrazione a Elastic Beanstalk, i percorsi fisici vengono mappati in posizioni appropriate sulle istanze EC2 dell'ambiente. Il processo di migrazione preserva la struttura dei contenuti garantendo al contempo che tutti i file vengano distribuiti correttamente nell'ambiente cloud.

ApplicationHost.config

Il file di configurazione principale per IIS che definisce le impostazioni a livello di server e contiene la configurazione per tutti i siti, le applicazioni e le directory virtuali. Questo file si trova nella `%windir%\System32\inetsrv\config` directory e controlla il comportamento generale del server IIS.

Durante la migrazione a Elastic Beanstalk, `applicationHost.config` le impostazioni pertinenti vengono estratte e applicate alla configurazione IIS sulle istanze EC2 dell'ambiente. Ciò garantisce che le impostazioni a livello di server vengano preservate durante la migrazione.

web.config

Un file di configurazione basato su XML utilizzato nelle applicazioni ASP.NET per controllare le impostazioni, la sicurezza e il comportamento delle applicazioni a livello di applicazione o di directory. `web.config`i file possono contenere impostazioni per l'autenticazione, l'autorizzazione, lo stato della sessione, la compilazione e i parametri personalizzati dell'applicazione.

Durante la migrazione a Elastic `web.config` Beanstalk, i file vengono conservati e distribuiti con l'applicazione. Il processo di migrazione garantisce che le configurazioni specifiche dell'applicazione continuino a funzionare come previsto nell'ambiente cloud.

DefaultDocument

Una funzionalità IIS che specifica il file predefinito da utilizzare quando un utente richiede una directory senza specificare un nome di file. I documenti predefiniti sono abilitati per impostazione predefinita e IIS 7 definisce i seguenti file di documento predefiniti nel `applicationHost.config` file come valori predefiniti a livello di server: Default.htm, Default.asp, Index.htm, Index.html, Iisstart.htm.

Durante la migrazione a Elastic Beanstalk, le impostazioni predefinite dei documenti vengono mantenute nella configurazione IIS sulle istanze EC2, garantendo che le richieste di directory vengano gestite in modo coerente nell'ambiente cloud.

System.WebServer

Una sezione di configurazione in `web.config` o `applicationHost.config` che contiene impostazioni specifiche di IIS per moduli, gestori e altri comportamenti del server. Questa sezione controlla il modo in cui IIS elabora le richieste, gestisce i moduli e configura le funzionalità del server.

Durante la migrazione a Elastic Beanstalk, le configurazioni System.WebServer vengono conservate nei file dell'`web.config`applicazione e applicate all'installazione IIS sulle istanze EC2 del tuo ambiente. Ciò garantisce che i comportamenti specifici di IIS vengano mantenuti nell'ambiente cloud.

Termini di Elastic Beanstalk

Platform (Piattaforma)

Una combinazione di sistema operativo, runtime del linguaggio di programmazione, server Web, server delle applicazioni e componenti Elastic Beanstalk che definiscono lo stack software per l'esecuzione delle applicazioni.

Per le migrazioni a Windows, Elastic Beanstalk fornisce piattaforme basate su Windows Server 2016, 2019 e 2022 con IIS e varie versioni.NET Framework per garantire la compatibilità con l'ambiente di origine.

SolutionStack

Una configurazione di piattaforma predefinita in Elastic Beanstalk che specifica il sistema operativo, il runtime e altri componenti necessari per eseguire un'applicazione. Concettualmente identica a una piattaforma e utilizzata in modo intercambiabile per gestire gli ambienti.

Durante la migrazione, il eb migrate comando seleziona uno stack di soluzioni appropriato in base alla configurazione dell'ambiente di origine, garantendo la compatibilità con le applicazioni IIS.

CreateEnvironment

Un'azione dell'API Elastic Beanstalk che crea un nuovo ambiente per ospitare una versione dell'applicazione. Questa API viene utilizzata dal eb migrate comando per fornire le AWS risorse necessarie per l'applicazione migrata.

Il processo di migrazione configura i parametri di ambiente appropriati in base all'ambiente IIS di origine, inclusi il tipo di istanza, le variabili di ambiente e le impostazioni delle opzioni.

CreateApplicationVersion

Un'azione API Elastic Beanstalk che crea una nuova versione dell'applicazione da un pacchetto sorgente archiviato in Amazon S3. Il eb migrate comando utilizza questa API per registrare l'applicazione IIS in pacchetto come versione in Elastic Beanstalk.

Durante la migrazione, i file e la configurazione dell'applicazione vengono impacchettati, caricati su Amazon S3 e registrati come versione dell'applicazione prima della distribuzione.

DescribeEvents

Un'azione dell'API Elastic Beanstalk che recupera un elenco di eventi per un ambiente, tra cui distribuzioni, modifiche alla configurazione e problemi operativi. Il eb migrate comando utilizza questa API per monitorare l'avanzamento della migrazione.

È inoltre possibile utilizzare il eb events comando dopo la migrazione per visualizzare la cronologia degli eventi del proprio ambiente.

DescribeEnvironmentHealth

Un'azione dell'API Elastic Beanstalk che fornisce informazioni dettagliate sullo stato delle istanze e degli altri componenti di un ambiente. Questa API viene utilizzata per verificare lo stato dell'applicazione migrata dopo la distribuzione.

Dopo la migrazione, è possibile utilizzare il eb health comando per verificare lo stato dell'ambiente e identificare eventuali problemi che richiedono attenzione.

HealthD

Un agente di monitoraggio in Elastic Beanstalk che raccoglie metriche, monitora i log e riporta lo stato di integrità delle istanze EC2 in un ambiente. HealthD fornisce report avanzati sullo stato delle applicazioni migrate.

Dopo la migrazione, HealthD monitora le prestazioni dell'applicazione, l'utilizzo delle risorse e le percentuali di successo delle richieste, fornendo una visione completa dello stato dell'ambiente.

Registri del pacchetto

Una funzionalità di Elastic Beanstalk che comprime e carica i log dalle istanze EC2 su Amazon S3 per l'archiviazione e l'analisi centralizzate. Questa funzionalità consente di risolvere i problemi relativi alle applicazioni migrate.

Dopo la migrazione, è possibile utilizzare il eb logs comando per recuperare e visualizzare i log dal proprio ambiente.

aws-windows-deployment-manifest.json

Un file che descrive il contenuto, le dipendenze e la configurazione di un pacchetto software o di un'applicazione. Questo manifesto viene generato durante il processo di migrazione per definire come le applicazioni IIS devono essere distribuite su Elastic Beanstalk.

sezione manifesto personalizzata

Una sezione all'interno `aws-windows-deployment-manifest.json` che fornisce un controllo personalizzato sulla distribuzione delle applicazioni. Questa sezione contiene PowerShell script e comandi che vengono eseguiti durante il processo di distribuzione.

Durante la migrazione, vengono generate sezioni manifeste personalizzate per gestire aspetti specifici della configurazione IIS, come la configurazione della directory virtuale, la gestione delle autorizzazioni e la configurazione del pool di applicazioni.

CLI EB

Uno strumento da riga di comando che fornisce comandi per creare, configurare e gestire applicazioni e ambienti Elastic Beanstalk. L'EB CLI include il eb migrate comando specifico per la migrazione delle applicazioni IIS su Elastic Beanstalk.

Dopo la migrazione, puoi continuare a utilizzare l'EB CLI per gestire l'ambiente, distribuire aggiornamenti, monitorare lo stato ed eseguire altre attività amministrative.

Impostazioni delle opzioni

Valori di configurazione che definiscono il modo in cui Elastic Beanstalk effettua il provisioning AWS e configura le risorse nell'ambiente. Le impostazioni delle opzioni sono organizzate in namespace che rappresentano diversi componenti dell'ambiente, come sistemi di bilanciamento del carico, istanze e processi ambientali.

Durante la migrazione, il eb migrate comando genera le impostazioni delle opzioni appropriate in base alla configurazione IIS per garantire che l'ambiente cloud corrisponda alle funzionalità dell'ambiente di origine. Per ulteriori informazioni, consulta Opzioni di configurazione nella Elastic Beanstalk Developer Guide.

aws:elbv2:listener:default

Uno spazio dei nomi di configurazione Elastic Beanstalk per il listener predefinito su un Application Load Balancer. Durante la migrazione, questo spazio dei nomi viene configurato in base alle associazioni del sito IIS per garantire il corretto routing del traffico.

Il listener predefinito gestisce in genere il traffico HTTP sulla porta 80, che viene quindi inoltrato alle istanze dell'applicazione in base alle regole di routing.

aws:elbv2:listener:listener_port

Uno spazio dei nomi di configurazione Elastic Beanstalk per una porta listener specifica su un Application Load Balancer. Questo spazio dei nomi viene utilizzato per configurare listener aggiuntivi per le applicazioni migrate, come HTTPS sulla porta 443.

Durante la migrazione, i listener vengono creati in base ai collegamenti delle porte dei siti IIS, garantendo che le applicazioni rimangano accessibili sulle stesse porte dell'ambiente di origine.

aws: elbv2: listenerrule:rule_name

Uno spazio dei nomi di configurazione Elastic Beanstalk per definire le regole di routing per un listener Application Load Balancer. Queste regole determinano il modo in cui le richieste in entrata vengono instradate a diversi gruppi di destinazione in base ai modelli di percorso o alle intestazioni degli host.

Durante la migrazione, le regole del listener vengono create in base alla struttura degli URL delle applicazioni IIS, garantendo che le richieste vengano instradate ai percorsi applicativi corretti.

aws: elasticbeanstalk:environment:process:default

Uno spazio dei nomi di configurazione Elastic Beanstalk per il processo predefinito in un ambiente. Questo spazio dei nomi definisce come viene configurato il processo predefinito

dell'applicazione Web, incluse le impostazioni del controllo dello stato, la mappatura delle porte e le impostazioni del proxy.

Durante la migrazione, il processo predefinito viene configurato in base alle impostazioni del sito IIS principale, garantendo il corretto monitoraggio dello stato e la gestione delle richieste.

aws: elasticbeanstalk:environment:process:process_name

Uno spazio dei nomi di configurazione Elastic Beanstalk per uno specifico processo denominato in un ambiente. Questo spazio dei nomi consente di definire più processi con configurazioni diverse, in modo simile ad avere più pool di applicazioni in IIS.

Durante la migrazione, è possibile creare processi aggiuntivi per rappresentare diverse associazioni di siti dall'ambiente di origine.

> **Note**
>
> Per ulteriori informazioni su alcuni dei termini descritti in questo argomento, consulta le seguenti risorse:
>
> - Azioni dell'API Elastic Beanstalk - Riferimento API AWS Elastic Beanstalk
> - Piattaforme Elastic Beanstalk, incluse le versioni della piattaforma supportate - Piattaforme supportate nella guida Piattaforme AWS Elastic Beanstalk
> - Namespace di configurazione Elastic Beanstalk - in questa guida Opzioni generali per tutti gli ambienti
> - L'EB CLI o comandi EB CLI specifici - in questa guida Configurazione dell'interfaccia a riga di comando EB (EB CLI) per gestire Elastic Beanstalk

Termini in Python

pip

Il programma di installazione dei pacchetti per Python, usato per installare e gestire pacchetti software scritti in Python. L'EB CLI viene installato e aggiornato tramite pip.

Durante il processo di migrazione, pip viene utilizzato per installare il pacchetto EB CLI e le sue dipendenze sul server di origine, fornendo gli strumenti necessari per la migrazione.

PyPI

Python Package Index, il repository ufficiale per i pacchetti software Python di terze parti, da cui pip recupera e installa i pacchetti. L'EB CLI e le sue dipendenze sono ospitate su PyPI.

Quando si installa l'EB CLI per la migrazione, pip si connette a PyPI per scaricare e installare i pacchetti necessari.

virtualenv

Uno strumento per creare ambienti Python isolati, che consente a diversi progetti di avere le proprie dipendenze e pacchetti senza conflitti. L'uso di virtualenv è consigliato durante l'installazione della CLI EB per evitare conflitti con altre applicazioni Python.

La creazione di un ambiente virtuale prima di installare l'EB CLI garantisce che gli strumenti di migrazione abbiano un ambiente pulito e isolato con le dipendenze corrette.

pywin32

Un insieme di estensioni Python che forniscono l'accesso a molte finestre APIs, abilitando l'interazione con il sistema operativo Windows e i suoi componenti. L'EB CLI utilizza pywin32 per interagire con le funzionalità specifiche di Windows durante la migrazione.

Durante il processo di migrazione, pywin32 viene utilizzato per accedere alla configurazione di IIS, alle impostazioni del registro di Windows e ad altre informazioni di sistema necessarie per impacchettare e migrare correttamente le applicazioni.

python.net

Un pacchetto che consente al codice Python di interagire con le applicazioni.NET Framework e.NET Core. Questa integrazione consente all'EB CLI di lavorare con i componenti.NET durante il processo di migrazione.

Il processo di migrazione può utilizzare pythonnet per interagire con gli assiemi e i componenti.NET durante l'analisi e la creazione di pacchetti delle applicazioni per la distribuzione su Elastic Beanstalk.

Esecuzione di migrazioni IIS di base

Questa sezione guida l'utente attraverso il processo di migrazione delle applicazioni IIS su Elastic eb migrate Beanstalk utilizzando il comando.

Esplorazione del tuo ambiente IIS

Prima di apportare modifiche, ti consigliamo di capire quali risorse esistono sul tuo server. Inizia esplorando i tuoi siti IIS eseguendolieb migrate explore, come mostrato nell'esempio seguente:

```
PS C:\migrations_workspace> eb migrate explore
```

Questo comando rivela i tuoi siti IIS. Fate riferimento al seguente elenco:

```
Default Web Site
Intranet
API.Internal
Reports
```

Per una visualizzazione dettagliata della configurazione di ogni sito, inclusi collegamenti, applicazioni e directory virtuali, aggiungete l'`--verbose`opzione, come mostrato in questo esempio:

```
PS C:\migrations_workspace> eb migrate explore --verbose
```

L'elenco seguente mostra le informazioni complete sull'ambiente fornite dal comando:

```
1: Default Web Site:
  - Bindings:
    - *:80:www.example.com
    - *:443:www.example.com
  - Application '/':
    - Application Pool: DefaultAppPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\inetpub\wwwroot
        - Logon Method: ClearText
  - Application '/api':
    - Application Pool: ApiPool
    - Enabled Protocols: http
    - Virtual Directories:
      - /:
        - Physical Path: C:\websites\api
        - Logon Method: ClearText
2: Intranet:
...
```

```
3. API.Internal:
...
4. Reports:
...
```

Comprensione dell'output del rilevamento

L'output dettagliato fornisce le seguenti informazioni critiche per la pianificazione della migrazione:

Siti

L'output di rilevamento elenca tutti i siti IIS sul server. Ogni sito è identificato dal proprio nome (ad esempio, «Sito Web predefinito», «Intranet», «API.Internal») e numerato in sequenza. Quando su un server sono presenti più siti, il `eb migrate` comando può impacchettarli e distribuirli separatamente o insieme, a seconda della strategia di migrazione.

Attacchi

Le associazioni di protocollo rivelano quali protocolli (HTTP/HTTPS) vengono utilizzati dai siti e su quali porte operano. Le informazioni vincolanti includono i requisiti dell'intestazione dell'host che definiscono le configurazioni di routing basate sul dominio.

Applicazioni

I percorsi delle applicazioni mostrano sia le strutture applicative root che quelle annidate all'interno della configurazione IIS. Le assegnazioni dei pool di applicazioni indicano in che modo le applicazioni sono isolate l'una dall'altra per la sicurezza e la gestione delle risorse.

Directory virtuali

Le mappature dei percorsi fisici indicano dove si trovano i contenuti nel file system. Le impostazioni di autenticazione mostrano requisiti di accesso speciali che devono essere mantenuti dopo la migrazione.

Preparativi per la migrazione

Con una conoscenza approfondita del vostro ambiente, assicuratevi che il server soddisfi i prerequisiti. Innanzitutto, verifica la versione di IIS in uso con il seguente PowerShell comando:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\InetStp\" -Name
 MajorVersion
```

È necessario IIS 7.0 o versione successiva. Lo strumento di migrazione utilizza Web Deploy 3.6 per impacchettare le applicazioni. Verificane l'installazione con il seguente comando:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\IIS Extensions
\MSDeploy\3" -Name InstallPath
```

Se Web Deploy non è installato sul server, è possibile scaricarlo dalla pagina di download di Microsoft Web Platform Installer.

La tua prima migrazione

Iniziamo con una migrazione di base del sito Web predefinito. L'esempio seguente mostra il comando più semplice,eb migrate.

```
PS C:\migrations_workspace> eb migrate
```

Questo comando avvia una serie di passaggi automatici, illustrati nell'output di esempio seguente:

```
Identifying VPC configuration of this EC2 instance (i-0123456789abcdef0)
  id: vpc-1234567890abcdef0
  publicip: true
  elbscheme: public
  ec2subnets: subnet-123,subnet-456,subnet-789
  securitygroups: sg-123,sg-456
  elbsubnets: subnet-123,subnet-456,subnet-789

Using .\migrations\latest to contain artifacts for this migration run.
Generating source bundle for sites, applications, and virtual directories...
  Default Web Site/ -> .\migrations\latest\upload_target\DefaultWebSite.zip
```

Lo strumento di migrazione crea una directory strutturata contenente gli artefatti della distribuzione. L'elenco seguente mostra la struttura delle cartelle:

```
C:\migration_workspace\
### .\migrations\latest\
    ### upload_target\
        ### DefaultWebSite.zip
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
```

```
            ### site_installer.ps1
            ### permission_handler.ps1
            ### >other helper scripts<
```

Controllo della migrazione

Per un maggiore controllo sul processo di migrazione, puoi specificare esattamente quali siti migrare con il seguente comando:

```
PS C:\migrations_workspace> eb migrate --sites "Default Web Site,Intranet"
```

È inoltre possibile personalizzare il nome dell'ambiente e il nome dell'applicazione, come illustrato nel seguente comando di esempio:

```
PS C:\migrations_workspace> eb migrate `
    --sites "Default Web Site" `
    --application-name "CorporateApp" `
    --environment-name "Production"
```

Per un elenco completo delle opzioni, vederethe section called “eb migrate”.

Monitoraggio dell'avanzamento

Durante la migrazione, eb migrate fornisce aggiornamenti sullo stato in tempo reale. Fate riferimento al seguente esempio di output:

```
...
Creating application version
Creating environment... This may take a few minutes

2024-03-18 18:12:15    INFO    Environment details for: Production
  Application name: CorporateApp
  Region: us-west-2
  Deployed Version: app-230320_153045
  Environment ID: e-abcdef1234
  Platform: 64bit Windows Server 2019 v2.7.0 running IIS 10.0
  Tier: WebServer-Standard-1.0
  CNAME: production.us-west-2.elasticbeanstalk.com
  Updated: 2024-03-20 15:30:45
2025-03-18 18:12:17    INFO    createEnvironment is starting.
```

```
2025-03-18 18:12:19    INFO    Using elasticbeanstalk-us-east-1-180301529717 as Amazon
 S3 storage bucket for environment data.
2025-03-18 18:12:40    INFO    Created security group named: sg-0fdd4d696a26b086a
2025-03-18 18:12:48    INFO    Environment health has transitioned to Pending.
 Initialization in progress (running for 7 seconds). There are no instances.
...
2025-03-18 18:23:59    INFO    Application available at EBMigratedEnv-arrreal3.us-
east-1.elasticbeanstalk.com.
2025-03-18 18:24:00    INFO    Successfully launched environment: EBMigratedEnv-
arrreal3
```

Verifica della migrazione

Una volta che l'ambiente è pronto, Elastic Beanstalk offre diversi modi per verificare la distribuzione.

Accedi alla tua applicazione

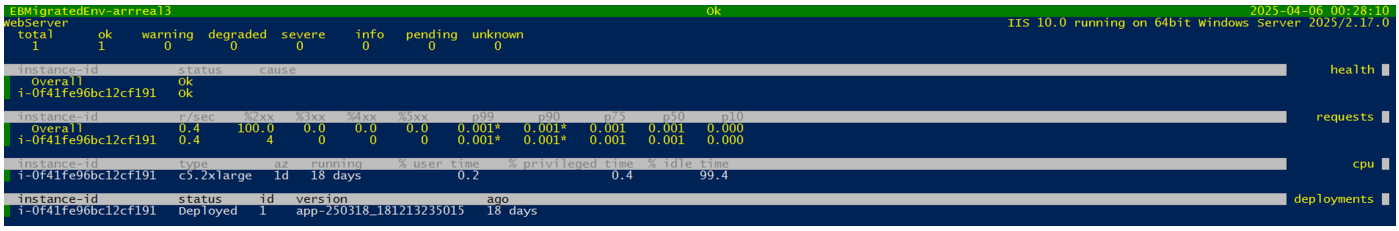
Apri l'URL dell'applicazione (CNAME) in un browser Web per verificare che funzioni correttamente.

Controlla lo stato dell'ambiente

Utilizzate il eb health comando per visualizzare lo stato del vostro ambiente.

```
PS C:\migrations_workspace> eb health
```

L'immagine della schermata seguente mostra lo stato dell'istanza, le metriche di risposta dell'applicazione e l'utilizzo delle risorse di sistema.



Usa il eb logs comando per accedere ai log e risolvere eventuali problemi:

```
PS C:\migrations_workspace> eb logs --zip
```

Il eb logs comando scarica i log nella directory. `.elasticbeanstalk/logs` Per ulteriori informazioni, consulta the section called “CloudWatch Registri”.

Connect alle istanze

Se hai specificato una key pair durante la migrazione, puoi connetterti alle tue istanze utilizzando RDP per la risoluzione diretta dei problemi.

Accedi alla console Elastic Beanstalk

Puoi visualizzare lo stato, i log e le proprietà di configurazione dell'ambiente tramite la console di gestione dell'ambiente per quell'ambiente.

Gestione degli artefatti della migrazione

Il eb migrate comando crea artefatti locali durante il processo di migrazione. Questi artefatti contengono informazioni riservate e possono consumare una notevole quantità di spazio su disco nel tempo. Utilizzate il cleanup sottocomando per gestire questi artefatti, come illustrato nell'esempio seguente:

```
PS C:\migrations_workspace> eb migrate cleanup
Are you sure you would like to cleanup older artifacts within ./migrations/? (Y/N):
```

Per forzare la pulizia senza conferma, utilizzate l'opzione: --force

```
PS C:\migrations_workspace> eb migrate cleanup --force
```

Il processo di pulizia preserva la migrazione riuscita più recente nella `./migrations/latest` directory e rimuove le directory di migrazione più vecchie

Configurazione di rete e impostazioni delle porte

Questa sezione descrive le opzioni di configurazione di rete per le migrazioni IIS, incluse le impostazioni VPC, le configurazioni delle porte e le distribuzioni multisito.

Configurazione VPC

Il eb migrate comando fornisce opzioni di configurazione VPC flessibili per l'ambiente Elastic Beanstalk. Lo strumento può rilevare le impostazioni VPC da un'istanza EC2 di origine o accettare configurazioni VPC personalizzate tramite parametri della riga di comando. Leggi Utilizzo di Elastic Beanstalk con Amazon VPC questo articolo per capire come configurare Elastic Beanstalk con VPC.

Rilevamento automatico del VPC

Quando eb migrate viene eseguito su un'istanza EC2, rileva e utilizza automaticamente la configurazione VPC dalle istanze EC2 dell'ambiente di origine. L'output di esempio seguente illustra le informazioni di configurazione rilevate:

```
PS C:\migrations_workspace > eb migrate
Identifying VPC configuration of this EC2 instance (i-0123456789abcdef0):
  id: vpc-1234567890abcdef0
  publicip: true
  elbscheme: public
  ec2subnets: subnet-123,subnet-456,subnet-789
  securitygroups: sg-123,sg-456
  elbsubnets: subnet-123,subnet-456,subnet-789
...
```

La configurazione rilevata include:

- Identificatore VPC
- Impostazioni di assegnazione IP pubblico
- Schema di bilanciamento del carico (pubblico/privato)
- Assegnazioni di sottoreti di istanze EC2
- Associazioni di gruppi di sicurezza
- Assegnazioni delle sottoreti di bilanciamento del carico

Host locali o non cloud AWS

Quando eb migrate viene eseguito da un server locale o da un host non AWS cloud, il servizio Elastic Beanstalk utilizzerà il VPC predefinito nel tuo account. AWS L'elenco seguente mostra un esempio di comando e output:

```
PS C:\migrations_worspace> eb migrate `
      -k windows-test-pem `
      --region us-east-1 `
      -a EBMigratedEnv `
      -e EBMigratedEnv-test2 `
      --copy-firewall-config
Determining EB platform based on host machine properties
```

```
Using .\migrations\latest to contain artifacts for this migration run.
...
```

Esamina Utilizzo di Elastic Beanstalk con Amazon VPC per capire come Elastic Beanstalk configura il VPC predefinito per il tuo ambiente.

Configurazione VPC personalizzata

Per qualsiasi ambiente di origine (EC2, locale o non AWS cloud) in cui sono necessarie impostazioni VPC specifiche, fornisci un file di configurazione VPC come quello nell'esempio seguente:

```
{
    "id": "vpc-12345678",
    "publicip": "true",
    "elbscheme": "public",
    "ec2subnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"],
    "securitygroups": "sg-123456,sg-789012",
    "elbsubnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"]
}
```

Applica questa configurazione utilizzando il seguente comando:

```
PS C:\migrations_workspace> eb migrate --vpc-config vpc-config.json
```

> **Note**
>
> Il file di configurazione VPC richiede il `id` campo che specifica l'ID VPC. Tutti gli altri campi sono facoltativi ed Elastic Beanstalk utilizzerà valori predefiniti per tutti i campi che non specifichi.

> **Important**
>
> La migrazione ignorerà tutte le impostazioni VPC esistenti dall'ambiente di origine quando si specifica il *`--vpc-config`* parametro. Quando utilizzi questo parametro, la migrazione utilizzerà solo le impostazioni VPC specificate nel file di configurazione che stai trasmettendo. L'utilizzo di questo parametro sostituisce il comportamento predefinito di rilevamento della configurazione VPC dell'istanza di origine o dell'utilizzo del VPC predefinito.

Utilizzate il `--vpc-config` parametro in questi scenari:

- Quando esegui la migrazione di ambienti non EC2 che non dispongono di impostazioni VPC rilevabili
- Quando si esegue la migrazione a un VPC diverso da quello utilizzato dall'ambiente di origine
- Quando è necessario personalizzare le selezioni delle sottoreti o le configurazioni dei gruppi di sicurezza
- Quando il rilevamento automatico non identifica correttamente le impostazioni VPC desiderate
- Quando esegui la migrazione da un ambiente locale e non desideri utilizzare il VPC predefinito

Configurazione della sicurezza di rete

Per impostazione predefinita, eb migrate apre la porta 80 sulle istanze di destinazione ma non copia altre regole di Windows Firewall dal computer di origine. Per includere tutte le configurazioni del firewall, utilizzare il seguente comando:

```
PS C:\migrations_workspace> eb migrate --copy-firewall-config
```

Questo comando esegue le seguenti azioni:

- Identifica le porte utilizzate dai collegamenti ai siti IIS
- Recupera le regole del firewall corrispondenti
- Genera PowerShell script per ricreare le regole sulle istanze di destinazione
- Conserva tutte le regole DENY per la porta 80 dal computer di origine (altrimenti la porta 80 è consentita per impostazione predefinita)

Considerate un caso d'uso, in cui il computer di origine ha le regole del firewall specificate nell'esempio seguente:

```
# Source machine firewall configuration
Get-NetFirewallRule | Where-Object {$_.Enabled -eq 'True'} | Get-NetFirewallPortFilter
 | Where-Object {$_.LocalPort -eq 80 -or $_.LocalPort -eq 443 -or $_.LocalPort -eq
 8081}
# Output shows rules for ports 80, 443, and 8081
```

La migrazione crea uno script (`modify_firewall_config.ps1`) che contiene la seguente configurazione:

```
New-NetFirewallRule -DisplayName "Allow Web Traffic" -Direction Inbound -Action Allow -
Protocol TCP -LocalPort 80,443
New-NetFirewallRule -DisplayName "Allow API Traffic" -Direction Inbound -Action Allow -
Protocol TCP -LocalPort 8081
```

Lo strumento di migrazione esegue automaticamente le seguenti azioni:

- Estrae HTTP/HTTPS le porte da tutte le associazioni del sito IIS
- Utilizza l'interfaccia Windows Firewall INetFwPolicy2 per enumerare le regole del firewall
- Filtra le regole per includere solo quelle che fanno riferimento esplicitamente alle porte specificate
- Elabora solo le associazioni di siti HTTP e HTTPS e le relative regole firewall associate
- Conserva le proprietà delle regole, tra cui nome visualizzato, azione, protocollo e stato abilitato
- Gestisce sia le singole porte che gli intervalli di porte nelle regole del firewall
- Aggiunge lo script di configurazione del firewall al manifesto di distribuzione

Configurazione del bilanciamento del carico

È possibile specificare la configurazione di Load Balancer tramite l'`--vpc-config`argomento. Gli esempi che seguono illustrano i parametri.

Selezione dello schema

Scegli tra schemi di bilanciamento del carico pubblici e privati:

```
{
    "id": "vpc-12345678",
    "elbscheme": "private",
    "elbsubnets": ["subnet-private1", "subnet-private2"]
}
```

Distribuzione tramite sottorete

Per un'elevata disponibilità, distribuisci le sottoreti di bilanciamento del carico tra le zone di disponibilità:

```
{
    "elbsubnets": [
        "subnet-az1", // Availability Zone 1
        "subnet-az2", // Availability Zone 2
```

```
        "subnet-az3"  // Availability Zone 3
    ]
}
```

> **Note**
>
> Sebbene Elastic Beanstalk supporti la creazione di ambienti con Application Load Balancer, Network Load Balancer e Classic Load Balancer, il comando supporta solo Application Load Balancer. eb migrate Per ulteriori informazioni sui tipi di bilanciamento del carico, consulta Sistema di bilanciamento del carico per l'ambiente Elastic Beanstalk.

Implementazioni multisite con configurazioni di porte

Il eb migrate comando gestisce distribuzioni IIS multisito complesse in cui le applicazioni possono condividere dipendenze o utilizzare porte non standard. Considerate il seguente esempio di una tipica configurazione aziendale con più siti:

```
<!-- IIS Configuration -->
<sites>
    <site name="Default Web Site" id="1">
        <bindings>
            <binding protocol="http" bindingInformation="*:80:www.example.com" />
        </bindings>
    </site>
    <site name="InternalAPI" id="2">
        <bindings>
            <binding protocol="http" bindingInformation="*:8081:api.internal" />
        </bindings>
    </site>
    <site name="ReportingPortal" id="3">
        <bindings>
            <binding protocol="http" bindingInformation="*:8082:reports.internal" />
        </bindings>
    </site>
</sites>
```

Per migrare questa configurazione, utilizzate il comando e i parametri di esempio seguenti:

```
PS C:\migrations_workspace> eb migrate `
```

```
    --sites "Default Web Site,InternalAPI,ReportingPortal" `
    --copy-firewall-config `
    --instance-type "c5.large"
```

Il eb migrate comando crea un pacchetto di distribuzione che conserva l'identità e la configurazione di ogni sito. Il comando genera un file `aws-windows-deployment-manifest.json` che definisce come questi siti devono essere distribuiti. L'esempio seguente mostra un file json generato:

```
{
    "manifestVersion": 1,
    "deployments": {
        "msDeploy": [
            {
                "name": "DefaultWebSite",
                "parameters": {
                    "appBundle": "DefaultWebSite.zip",
                    "iisPath": "/",
                    "iisWebSite": "Default Web Site"
                }
            }
        ],
        "custom": [
            {
                "name": "InternalAPI",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\\install_site_InternalAPI.ps1"
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\restart_site_InternalAPI.ps1"
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\uninstall_site_InternalAPI.ps1"
                    }
                }
            },
            {
                "name": "ReportingPortal",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\\install_site_ReportingPortal.ps1"
                    },
                    "restart": {
```

```
                        "file": "ebmigrateScripts\\restart_site_ReportingPortal.ps1"
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\uninstall_site_ReportingPortal.ps1"
                    }
                }
            }
        ]
    }
}
```

Il processo di migrazione crea le seguenti regole del listener Application Load Balancer che mantengono la logica di routing originale:

- Porta 80 rotte di traffico al sito Web predefinito
- Porta il traffico 8081 verso InternalAPI
- Le rotte di traffico della porta 8082 verso ReportingPortal

Configurazione e dipendenze condivise

Quando i siti condividono configurazioni o dipendenze, eb migrate gestisce queste relazioni in modo appropriato. Fai riferimento al seguente esempio in cui più siti condividono una configurazione comune:

```
<!-- Shared configuration in applicationHost.config -->
<location path="Default Web Site">
    <system.webServer>
        <asp enableSessionState="true" />
        <caching enabled="true" enableKernelCache="true" />
    </system.webServer>
</location>
```

Il processo di migrazione completa le seguenti attività:

1. Identifica le configurazioni condivise tra i siti
2. Genera PowerShell script appropriati per applicare queste impostazioni
3. Mantiene la gerarchia e l'ereditarietà della configurazione

Best practice

Ti consigliamo di seguire le migliori pratiche per la configurazione di rete dell'applicazione migrata. I seguenti raggruppamenti forniscono linee guida riassuntive.

Progettazione VPC

- Segui le AWS best practice di progettazione VPC
- Utilizza sottoreti separate per i sistemi di bilanciamento del carico e le istanze EC2
- Implementa tabelle di routing appropriate e NACLs
- Prendi in considerazione gli endpoint VPC per i servizi AWS

Elevata disponibilità

- Esecuzione dell'implementazione in più zone di disponibilità
- Utilizza almeno due sottoreti per i sistemi di bilanciamento del carico
- Configura l'auto-scaling su AZs
- Implementa controlli sanitari adeguati

Sicurezza

- Segui le migliori pratiche di sicurezza
- Utilizza i gruppi di sicurezza come controllo di accesso principale
- Implementa gli elenchi di controllo degli accessi alla rete (ACLs) per una maggiore sicurezza
- Monitora i log di flusso VPC

Risoluzione dei problemi

I problemi più comuni di configurazione della rete includono le seguenti aree. Dopo ogni argomento sono riportati alcuni comandi di esempio per ottenere ulteriori informazioni sulla configurazione di rete e sullo stato dell'ambiente.

Configurazione della sottorete

```
# Verify subnet availability
PS C:\migrations_workspace> aws ec2 describe-subnets --subnet-ids subnet-id

# Check available IP addresses
```

```
PS C:\migrations_workspace>aws ec2 describe-subnets --subnet-ids subnet-id --query
 'Subnets[].AvailableIpAddressCount'
```

Accesso ai gruppi di sicurezza

```
# Verify security group rules
PS C:\migrations_workspace> aws ec2 describe-security-groups --group-ids sg-id

# Test network connectivity
PS C:\migrations_workspace> aws ec2 describe-network-interfaces --filters
 Name=group-id,Values=sg-id
```

Stato del sistema di bilanciamento del carico

```
# Check load balancer health
PS C:\migrations_workspace> aws elbv2 describe-target-health --target-group-arn
 arn:aws:elasticloadbalancing:region:account-id:targetgroup/group-name/group-id
```

Configurazioni di sicurezza e ruoli IAM

Il eb migrate comando gestisce le configurazioni AWS di sicurezza tramite ruoli IAM, profili di istanza e ruoli di servizio. La comprensione di questi componenti garantisce il corretto controllo degli accessi e la conformità alla sicurezza durante la migrazione.

Configurazione del profilo dell'istanza

Un profilo di istanza funge da contenitore per un ruolo IAM che Elastic Beanstalk collega alle istanze EC2 nel tuo ambiente. Durante l'esecuzioneeb migrate, puoi specificare un profilo di istanza personalizzato:

```
PS C:\migrations_workspace> eb migrate --instance-profile "CustomInstanceProfile"
```

Se non specifichi un profilo di istanza, eb migrate crea un profilo predefinito con queste autorizzazioni:

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-*",
                "arn:aws:s3:::elasticbeanstalk-*/*"
            ]
        }
    ]
}
```

Gestione dei ruoli di servizio

Un ruolo di servizio consente a Elastic Beanstalk di gestire le risorse AWS per tuo conto. Specificate un ruolo di servizio personalizzato durante la migrazione con il seguente comando:

```
PS C:\migrations_workspace> eb migrate --service-role "CustomServiceRole"
```

Se non specificato, eb migrate crea un ruolo di servizio predefinito denominato `aws-elasticbeanstalk-service-role` con una policy di fiducia che consente a Elastic Beanstalk di assumere il ruolo. Questo ruolo di servizio è essenziale per Elastic Beanstalk per monitorare lo stato dell'ambiente ed eseguire aggiornamenti gestiti della piattaforma. Il ruolo di servizio richiede due politiche gestite:

- `AWSElasticBeanstalkEnhancedHealth`- Consente a Elastic Beanstalk di monitorare lo stato dell'istanza e dell'ambiente utilizzando il sistema di reporting sanitario avanzato
- `AWSElasticBeanstalkManagedUpdates`- Consente a Elastic Beanstalk di eseguire aggiornamenti gestiti della piattaforma, incluso l'aggiornamento delle risorse dell'ambiente quando è disponibile una nuova versione della piattaforma

Con queste politiche, il ruolo di servizio dispone delle autorizzazioni per:

- Crea e gestisci gruppi di Auto Scaling

- Crea e gestisci Application Load Balancer
- Carica i log su Amazon CloudWatch
- Gestisci le istanze EC2

Per ulteriori informazioni sui ruoli di servizio, consulta Ruolo di servizio Elastic Beanstalk la Elastic Beanstalk Developer Guide.

Configurazione del gruppo di sicurezza

Il eb migrate comando configura automaticamente i gruppi di sicurezza in base alle associazioni del sito IIS. Ad esempio, se l'ambiente di origine ha siti che utilizzano le porte 80, 443 e 8081, i seguenti risultati di configurazione:

```
<site name="Default Web Site">
    <bindings>
        <binding protocol="http" bindingInformation="*:80:" />
        <binding protocol="https" bindingInformation="*:443:" />
    </bindings>
</site>
<site name="InternalAPI">
    <bindings>
        <binding protocol="http" bindingInformation="*:8081:" />
    </bindings>
</site>
```

Il processo di migrazione completa le seguenti azioni:

- Crea un gruppo di sicurezza con sistema di bilanciamento del carico che consente il traffico in entrata sulle porte 80 e 443 da Internet (0.0.0.0/0)
- Crea un gruppo di sicurezza EC2 che consente il traffico proveniente dal sistema di bilanciamento del carico
- Configura porte aggiuntive (come 8081) se specificato `--copy-firewall-config`

Per impostazione predefinita, l'Application Load Balancer è configurato con accesso pubblico da Internet. Se è necessario personalizzare questo comportamento, ad esempio limitando l'accesso a intervalli IP specifici o utilizzando un sistema di bilanciamento del carico privato, è possibile sovrascrivere la configurazione predefinita del VPC e del gruppo di sicurezza utilizzando il parametro: `--vpc-config`

```
PS C:\migrations_workspace> eb migrate --vpc-config vpc-config.json
```

Ad esempio, la seguente `vpc-config.json` configurazione crea un sistema di bilanciamento del carico privato in una sottorete privata:

```
{
    "id": "vpc-12345678",
    "publicip": "false",
    "elbscheme": "internal",
    "ec2subnets": ["subnet-private1", "subnet-private2"],
    "elbsubnets": ["subnet-private1", "subnet-private2"]
}
```

Per ulteriori informazioni sulle opzioni di configurazione del VPC, vedere. Configurazione VPC

Integrazione del certificato SSL

Durante la migrazione di siti con collegamenti HTTPS, integra i certificati SSL tramite (ACM): AWS Certificate Manager

```
PS C:\migrations_workspace> eb migrate --ssl-certificates
 "arn:aws:acm:region:account:certificate/certificate-id"
```

Questa configurazione completa le seguenti azioni:

- Associa il certificato all'Application Load Balancer
- Mantiene la terminazione HTTPS sul sistema di bilanciamento del carico
- Conserva la comunicazione HTTP interna tra il load balancer e le istanze EC2

Autenticazione Windows

Per le applicazioni che utilizzano l'autenticazione di Windows, eb migrate conserva le impostazioni di autenticazione nell'applicazione `web.config` come segue:

```
<configuration>
    <system.webServer>
        <security>
            <authentication>
```

```
                <windowsAuthentication enabled="true">
                    <providers>
                        <add value="Negotiate" />
                        <add value="NTLM" />
                    </providers>
                </windowsAuthentication>
            </authentication>
        </security>
    </system.webServer>
</configuration>
```

> **Important**
>
> Il eb migrate comando non copia i profili utente o gli account dall'ambiente di origine alle istanze Elastic Beanstalk di destinazione. Tutti gli account o i gruppi utente personalizzati che hai creato sul server di origine dovranno essere ricreati nell'ambiente di destinazione dopo la migrazione.

Gli account `IUSR` e i gruppi predefiniti di Windows`IIS_IUSRS`, nonché tutti gli altri account e gruppi predefiniti, sono inclusi per impostazione predefinita nelle istanze di Windows Server di destinazione. Per ulteriori informazioni sugli account e i gruppi IIS incorporati, vedere Comprendere gli account utente e di gruppo incorporati in IIS nella documentazione Microsoft.

Se l'applicazione si basa su account utente Windows personalizzati o sull'integrazione con Active Directory, sarà necessario configurare questi aspetti separatamente dopo il completamento della migrazione.

Best practice e risoluzione dei problemi

Gestione del ruolo

Implementa le best practice AWS IAM nella gestione dei ruoli per i tuoi ambienti Elastic Beanstalk:

Creazione e gestione dei ruoli

- Crea ruoli utilizzando politiche AWS gestite, ove possibile
- Segui le best practice di sicurezza IAM
- Utilizza il AWS Policy Generator per politiche personalizzate
- Implementa i limiti di autorizzazione per una maggiore sicurezza

Monitoraggio e controllo

Abilita AWS CloudTrail il monitoraggio dell'utilizzo dei ruoli:

- Segui la guida AWS CloudTrail per l'utente
- Configura l'integrazione CloudWatch dei registri per il monitoraggio in tempo reale
- Imposta avvisi per chiamate API non autorizzate

Processo di revisione regolare

Stabilisci un ciclo di revisione trimestrale per svolgere le seguenti attività:

- Verifica le autorizzazioni non utilizzate utilizzando IAM Access Analyzer
- Rimuovi le autorizzazioni obsolete
- Aggiorna i ruoli in base ai principi del privilegio minimo

Gestione dei certificati

Implementa queste pratiche per SSL/TLS i certificati nei tuoi ambienti Elastic Beanstalk:

Ciclo di vita dei certificati

- Utilizzo per la gestione AWS Certificate Managerdei certificati
- Abilita il rinnovo automatico per i certificati emessi da ACM
- Imposta le notifiche di scadenza

Standard di sicurezza

- Usa TLS 1.2 o versione successiva
- Segui le politiche AWS di sicurezza per i listener HTTPS
- Implementa HTTP Strict Transport Security (HSTS) se necessario

Gestione dei gruppi di sicurezza

Implementa queste best practice per i gruppi di sicurezza:

Gestione delle regole

- Documenta tutti i requisiti di porta personalizzati
- Usa i log di flusso VPC per monitorare il traffico
- Utilizza le regole di riferimento del gruppo di sicurezza anziché gli intervalli IP, ove possibile

Controllo regolare

Stabilisci revisioni mensili per svolgere le seguenti attività:

- Identifica e rimuovi le regole non utilizzate
- Convalida source/destination i requisiti
- Verifica la presenza di regole sovrapposte

Registrazione di log e monitoraggio

Per un monitoraggio efficace della sicurezza, configura i seguenti registri:

Registri degli eventi di Windows sulle istanze EC2

```
# Review Security event log
PS C:\migrations_workspace> Get-EventLog -LogName Security -Newest 50

# Check Application event log
PS C:\migrations_workspace> Get-EventLog -LogName Application -Source "IIS*"
```

CloudWatch Integrazione dei registri

Configura CloudWatch Logs agent per lo streaming dei registri degli eventi di Windows CloudWatch per il monitoraggio e gli avvisi centralizzati.

Per problemi persistenti, raccogli questi registri e contatta Supporto AWS le seguenti informazioni:

- ID ambiente
- ID di distribuzione (se applicabile)
- Messaggi di errore pertinenti
- Cronologia delle modifiche alla sicurezza

Informazioni sulla mappatura della migrazione da IIS a Elastic Beanstalk

La migrazione da IIS a Elastic Beanstalk prevede la mappatura della configurazione del server Windows locale alle risorse cloud. AWS La comprensione di questa mappatura è fondamentale per il successo delle migrazioni e per la gestione post-migrazione.

Siti e applicazioni IIS in Elastic Beanstalk

In IIS, un sito Web rappresenta una raccolta di applicazioni Web e directory virtuali, ognuna con la propria configurazione e il proprio contenuto. Durante la migrazione a Elastic Beanstalk, questi componenti vengono trasformati come segue:

Siti Web IIS

I tuoi siti Web IIS diventano applicazioni all'interno di Elastic Beanstalk. La configurazione di ogni sito Web, inclusi i collegamenti, i pool di applicazioni e le impostazioni di autenticazione, viene preservata tramite il file manifest di distribuzione di Elastic Beanstalk (). `aws-windows-deployment-manifest.json`

Ad esempio, se disponi di più siti come Default Web Site e IntranetSite, eb migrate impacchetta il contenuto e la configurazione di ciascun sito mantenendone l'isolamento.

Il comando crea le regole del listener Application Load Balancer (ALB) appropriate per gestire le richieste di routing verso le applicazioni. Configura inoltre i gruppi di sicurezza per garantire l'accesso corretto alle porte in base ai collegamenti IIS originali.

Pool di applicazioni

I pool di applicazioni IIS forniscono funzionalità di isolamento dei processi di lavoro, gestione del runtime e riciclaggio per le applicazioni. In Elastic Beanstalk, questi sono mappati su processi di ambiente definiti `aws:elasticbeanstalk:environment:process` tramite lo spazio dei nomi e configurati tramite IIS sulle istanze. EC2

La migrazione preserva le impostazioni critiche del pool di applicazioni, tra cui:

- Configurazioni del modello di processo: identità (o account personalizzati) ApplicationPoolIdentity NetworkService, impostazioni del timeout di inattività e intervalli di riciclo dei processi
- Impostazioni della versione.NET CLR: mantiene la versione di .NET Framework specificata (v2.0, v4.0 o No Managed Code) per garantire la compatibilità delle applicazioni
- Modalità pipeline gestita: conserva le impostazioni della modalità pipeline integrata o classica per mantenere l'architettura di elaborazione delle richieste HTTP
- Impostazioni avanzate: lunghezza della coda, limiti della CPU, soglie di protezione in caso di guasto rapido e limiti di tempo di avvio

Il eb migrate comando conserva le mappature tra siti e pool di applicazioni durante la migrazione all'ambiente Elastic Beanstalk.

Se i pool di applicazioni utilizzano pianificazioni di riciclaggio personalizzate (orari o soglie di memoria specifici), queste vengono implementate tramite PowerShell script nel pacchetto di distribuzione che configurano le impostazioni IIS appropriate sulle istanze. EC2

Associazioni a siti Web

Le associazioni ai siti Web IIS, che definiscono il modo in cui i client accedono alle applicazioni, vengono trasformate nelle seguenti configurazioni di Application Load Balancer (ALB):

- Le associazioni delle porte sono mappate alle regole del listener ALB corrispondenti
- Le configurazioni dell'intestazione dell'host vengono tradotte in regole di routing ALB
- I siti abilitati per SSL utilizzano AWS Certificate Manager (ACM) per la gestione dei certificati

Gestione delle directory virtuali e dei percorsi delle applicazioni

Le directory e le applicazioni virtuali IIS forniscono la mappatura dei percorsi degli URL alle directory fisiche. Elastic Beanstalk mantiene queste relazioni attraverso i seguenti costrutti:

Directory virtuali

Il processo di migrazione preserva i percorsi fisici delle directory virtuali nel pacchetto di distribuzione.

Le mappature dei percorsi sono configurate nella configurazione IIS sulle EC2 istanze, garantendo che la struttura degli URL rimanga intatta dopo la migrazione.

Il sistema non di sistema guida percorsi fisici

> **Important**
>
> Per impostazione predefinita, gli ambienti Windows Elastic Beanstalk forniscono solo l'unità C:\ (volume root). Nella versione corrente, le applicazioni con contenuti su unità non di sistema (D:\, E:\, ecc.) non sono supportate per la migrazione.

Il eb migrate comando rileva automaticamente i percorsi fisici che si trovano su unità non di sistema e avvisa l'utente in caso di potenziali problemi, come nell'esempio seguente:

```
ERROR: Detected physical paths on drive D:\ which are not supported in the current
 version:
  - D:\websites\intranet
```

```
  - D:\shared\images

Migration of content from non-system drives is not supported. Please relocate this
 content to the C:\ drive before migration. Otherwise, select only those sites that
 are on C:\.
```

Se l'applicazione dipende da unità non di sistema, sarà necessario modificare l'applicazione per archiviare tutto il contenuto sull'unità C:\ prima della migrazione.

Applicazioni annidate

Le applicazioni annidate nei siti Web vengono distribuite con le configurazioni di percorso corrette e le assegnazioni appropriate dei pool di applicazioni. Il processo di migrazione conserva tutte le `web.config` impostazioni, garantendo che le configurazioni specifiche dell'applicazione continuino a funzionare come previsto nell'ambiente cloud.

Riscrittura degli URL e routing delle richieste di applicazioni (ARR)

Se la distribuzione IIS utilizza URL Rewrite o Application Request Routing (ARR), eb migrate gestisce queste configurazioni tramite le seguenti regole e configurazioni:

Regole di riscrittura degli URL

Le regole di riscrittura degli URL `web.config` dei file vengono tradotte in regole di routing ALB, ove possibile. Ad esempio, la voce seguente diventa una regola di listener ALB che indirizza il traffico in base alle intestazioni degli host e ai modelli di percorso. :

```
<!-- Original IIS URL Rewrite Rule -->
<rule name="Redirect to WWW" stopProcessing="true">
    <match url="(.*)" />
    <conditions>
        <add input="{HTTP_HOST}" pattern="^example.com$" />
    </conditions>
    <action type="Redirect" url="http://www.example.com/{R:1}" />
</rule>
```

Routing delle richieste di applicazione

Le configurazioni ARR vengono preservate mediante l'installazione di funzionalità ARR sulle istanze. EC2 Il processo di migrazione completa le seguenti attività:

- Configura le impostazioni del proxy in modo che corrispondano all'ambiente di origine
- Mantiene le regole di riscrittura degli URL associate a ARR

Struttura degli artefatti di migrazione

Quando viene eseguitoeb migrate, crea una directory strutturata contenente tutti i componenti di distribuzione necessari. L'elenco seguente descrive la struttura delle cartelle:

```
C:\migration_workspace\
### .\migrations\latest\
    ### upload_target\
        ### [SiteName].zip                # One ZIP per IIS site
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
            ### site_installer.ps1        # Site installation scripts
            ### arr_configuration.ps1     # ARR configuration scripts
            ### permission_handler.ps1    # Permission management
            ### firewall_config.ps1       # Windows Firewall rules
```

Il `aws-windows-deployment-manifest.json` file è il file di configurazione principale che indica a Elastic Beanstalk come distribuire le applicazioni. Fai riferimento alla seguente struttura di esempio:

```
{
    "manifestVersion": 1,
    "deployments": {
        "msDeploy": [
            {
                "name": "Primary Site",
                "parameters": {
                    "appBundle": "DefaultWebSite.zip",
                    "iisPath": "/",
                    "iisWebSite": "Default Web Site"
                }
            }
        ],
        "custom": [
            {
                "name": "ConfigureARR",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\\arr_configuration.ps1"
```

```
                    },
                    "uninstall": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    }
                }
            }
        ]
    }
}
```

Questo manifesto garantisce questi risultati per la migrazione:

- Le applicazioni vengono distribuite per correggere i percorsi IIS
- Vengono applicate configurazioni personalizzate
- Le impostazioni specifiche del sito vengono mantenute
- L'ordine di distribuzione viene mantenuto

Scenari di migrazione avanzati

Questa sezione descrive gli scenari di migrazione avanzati per implementazioni IIS complesse.

Migrazioni multisito con Application Request Routing (ARR)

Il eb migrate comando rileva e conserva automaticamente le configurazioni ARR durante la migrazione. Quando identifica le impostazioni ARR in IIS`applicationHost.config`, genera gli PowerShell script necessari per reinstallare e configurare ARR sulle istanze di destinazione. EC2

Rilevamento della configurazione ARR

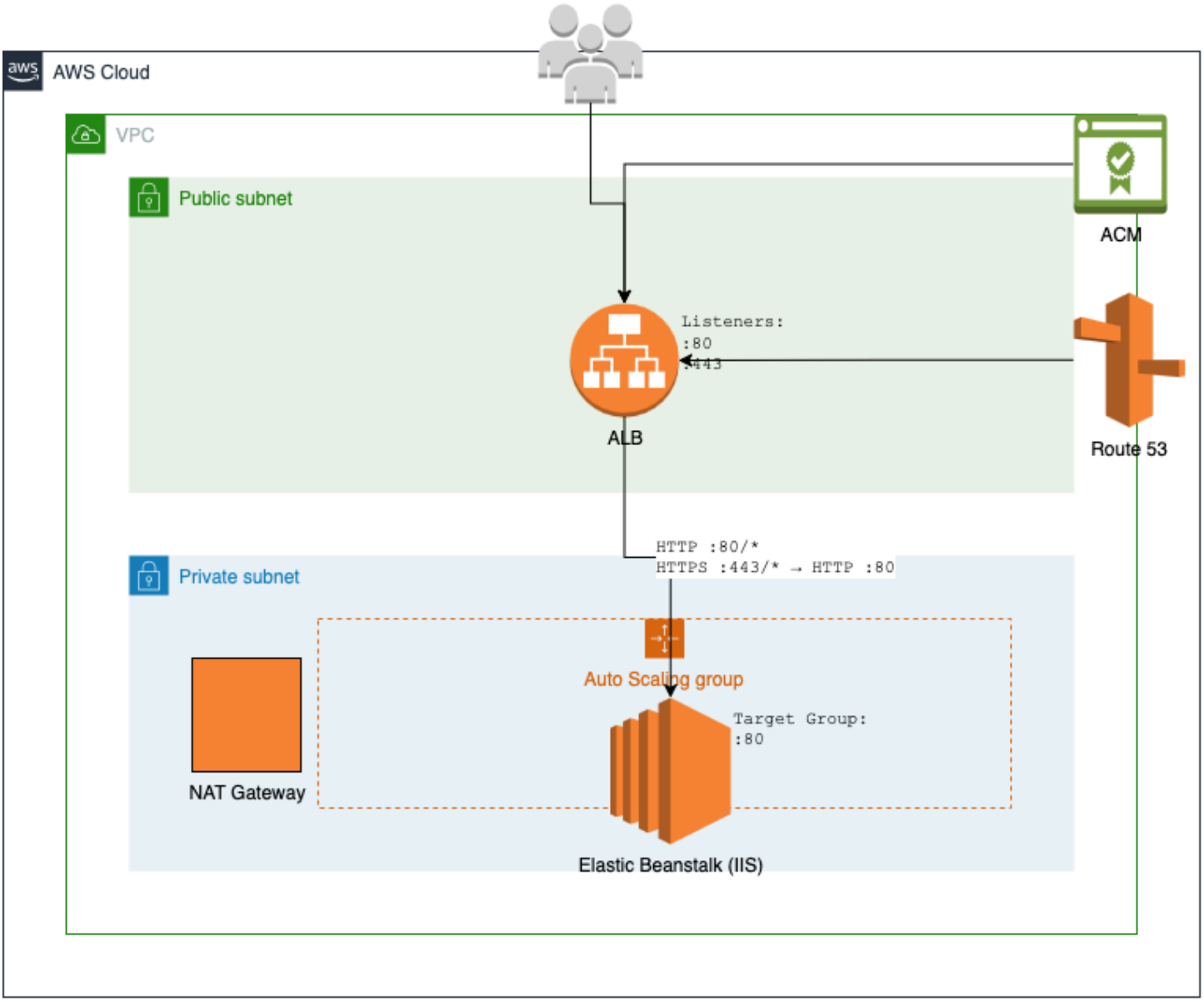
Il processo di migrazione esamina tre sezioni di configurazione chiave in IIS:

- `system.webServer/proxy`: impostazioni principali del proxy ARR
- `system.webServer/rewrite`: regole di riscrittura degli URL
- `system.webServer/caching`: configurazione della memorizzazione nella cache

Ad esempio, si consideri una configurazione ARR comune in cui un proxy `RouterSite` in esecuzione sulla porta 80 richiede e viene `AdminPortal` eseguito rispettivamente sulle porte 8081 `APIService` e 8082:

```
<!-- Original IIS ARR Configuration -->
<rewrite>
    <rules>
        <rule name="Route to API" stopProcessing="true">
            <match url="^api/(.*)$" />
            <action type="Rewrite" url="http://backend:8081/api/{R:1}" />
        </rule>
        <rule name="Route to Admin" stopProcessing="true">
            <match url="^admin/(.*)$" />
            <action type="Rewrite" url="http://backend:8082/admin/{R:1}" />
        </rule>
    </rules>
</rewrite>
```

Il diagramma seguente illustra come queste regole siano nascoste dietro la porta 80 nel server IIS e non siano esposte tramite i gruppi di sicurezza. EC2 Solo la porta 80 è accessibile all'Application Load Balancer e tutto il traffico proveniente da essa viene indirizzato al gruppo di destinazione sulla porta 80.



Il comando seguente può migrare questa configurazione:

```
PS C:\migrations_workspace> eb migrate --sites "RouterSite,APIService,AdminPortal" `
    --copy-firewall-config
```

Processo di migrazione ARR

Il processo di migrazione preserva la configurazione ARR attraverso diversi passaggi.

Esportazione della configurazione

Lo strumento esporta le impostazioni ARR esistenti dalle tre sezioni di configurazione chiave in file XML separati archiviati nella `ebmigrateScripts` directory:

```
ebmigrateScripts\
### arr_config_proxy.xml
### arr_config_rewrite.xml
### arr_config_caching.xml
```

Script di installazione

Vengono generati due PowerShell script per gestire la configurazione ARR:

1. `arr_msi_installer.ps1`: scarica e installa il modulo ARR
2. `arr_configuration_importer_script.ps1`: Importa la configurazione ARR esportata

Integrazione del manifesto di distribuzione

Gli script sono integrati nel processo di distribuzione tramite voci in`aws-windows-deployment-manifest.json`:

```
{
    "manifestVersion": 1,
    "deployments": {
        "custom": [
            {
                "name": "WindowsProxyFeatureEnabler",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\
\windows_proxy_feature_enabler.ps1"
                    }
                }
            },
            {
                "name": "ArrConfigurationImporterScript",
                "scripts": {
                    "install": {
                        "file": "ebmigrateScripts\
\arr_configuration_importer_script.ps1"
                    }
                }
```

```
            }
        ]
    }
}
```

Integrazione del sistema di bilanciamento del carico

Il processo di migrazione traduce le regole ARR in regole del listener Application Load Balancer (ALB), ove possibile. Ad esempio, la configurazione ARR di cui sopra comporta regole ALB che indirizzano il traffico in base ai modelli di percorso degli URL mantenendo il routing interno sulle istanze. EC2

L'ambiente risultante mantiene la logica di routing ARR sfruttando al contempo l'infrastruttura elastica di cui dispone. AWS Le applicazioni continuano a funzionare come prima, con ARR che gestisce il routing interno mentre Application Load Balancer gestisce la distribuzione del traffico esterno.

Migrazioni multisito senza ARR utilizzando il routing basato su host

Sebbene Application Request Routing (ARR) sia un approccio comune per la gestione di più siti in IIS, puoi anche migrare le implementazioni multisito direttamente su Elastic Beanstalk senza ARR sfruttando le funzionalità di routing basato su host di Application Load Balancer. Questo approccio può ridurre la complessità e migliorare le prestazioni eliminando un livello di routing aggiuntivo.

Panoramica del routing basato su host

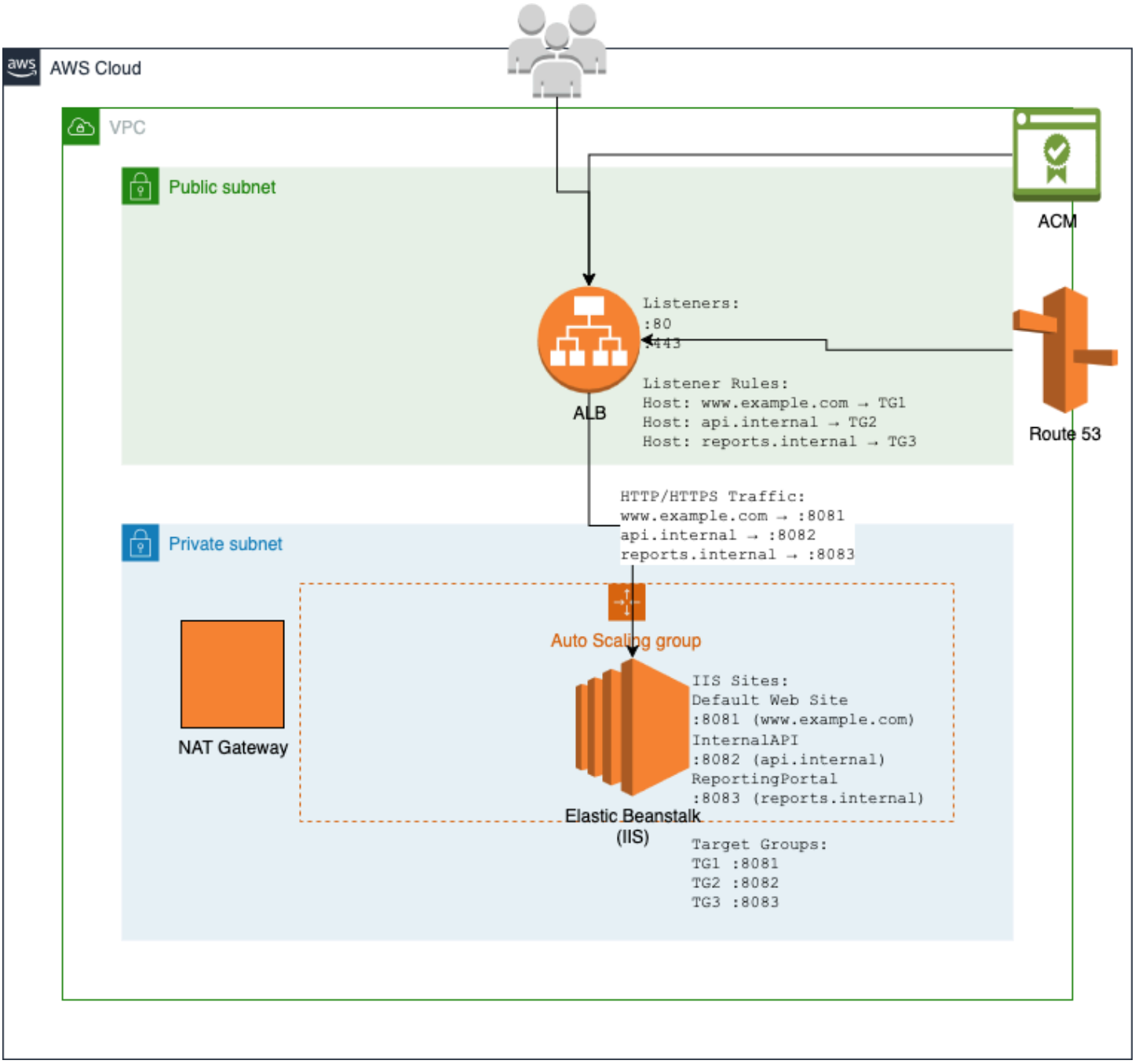
In questo approccio, ogni sito IIS è esposto all'esterno dell' EC2 istanza e l'Application Load Balancer indirizza il traffico direttamente alla porta appropriata in base all'intestazione dell'host. Ciò elimina la necessità di ARR mantenendo al contempo la separazione tra le applicazioni.

Prendi in considerazione una configurazione IIS multisito con tre siti, ciascuno con il proprio nome host binding:

```
<sites>
    <site name="Default Web Site" id="1">
        <bindings>
            <binding protocol="http" bindingInformation="*:8081:www.example.com" />
        </bindings>
    </site>
    <site name="InternalAPI" id="2">
```

```
        <bindings>
            <binding protocol="http" bindingInformation="*:8082:api.internal" />
        </bindings>
    </site>
    <site name="ReportingPortal" id="3">
        <bindings>
            <binding protocol="http" bindingInformation="*:8083:reports.internal" />
        </bindings>
    </site>
</sites>
```

Questi siti sono esposti alle porte 8081, 8082 e 8083 tramite i gruppi di sicurezza. EC2 L'Application Load Balancer li indirizza in base alla configurazione delle regole del listener Load Balancer.



Processo di migrazione

Per migrare questa configurazione su Elastic Beanstalk senza usare ARReb migrate, usa il comando nell'esempio seguente:

```
PS C:\migrations_workspace> eb migrate --sites "Default Web
 Site,InternalAPI,ReportingPortal"
```

Il processo di migrazione configura automaticamente l'Application Load Balancer con regole di routing basate su host che indirizzano il traffico verso il gruppo di destinazione appropriato in base all'intestazione dell'host. Ogni gruppo target inoltra il traffico alla porta corrispondente sulle istanze: EC2

1. Host header www.example.com → Target Group sulla porta 8081
2. Host header api.internal → Target Group sulla porta 8082
3. Host header reports.internal → Target Group sulla porta 8083

Configurazione SSL/TLS

Per proteggere le tue applicazioni, procedi nel seguente SSL/TLS modo:

1. Richiedi i certificati per i tuoi domini tramite AWS Certificate Manager(ACM).
2. Configura i listener HTTPS sul tuo Application Load Balancer utilizzando questi certificati.
3. Aggiorna la configurazione dell'ambiente per includere i listener HTTPS con le seguenti impostazioni delle opzioni di configurazione.

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: HTTPS
    SSLCertificateId: arn:aws:acm:region:account-id:certificate/certificate-id
    InstancePort: 80
    InstanceProtocol: HTTP
```

Con questa configurazione, la terminazione SSL avviene sul sistema di bilanciamento del carico e il traffico viene inoltrato alle istanze tramite HTTP. Ciò semplifica la gestione dei certificati mantenendo al contempo connessioni sicure con i client.

Best practice

Gruppi di sicurezza

Configura i gruppi di sicurezza per consentire il traffico in entrata solo sulle porte utilizzate dai tuoi siti IIS (8081, 8082, 8083 in questo esempio) dal gruppo di sicurezza Application Load Balancer.

Controlli dell’integrità

Configura i controlli di integrità per ogni gruppo target per garantire che il traffico venga indirizzato solo verso istanze integre. Crea endpoint per il controllo dello stato di salute per ogni applicazione, se non esistono già.

Monitoraggio

Imposta CloudWatch allarmi per monitorare separatamente lo stato e le prestazioni di ciascun gruppo target. Ciò consente di identificare problemi specifici delle singole applicazioni.

Dimensionamento

Considerate i requisiti di risorse di tutte le applicazioni quando configurate le politiche di auto scaling. Se un'applicazione ha esigenze di risorse significativamente diverse, prendi in considerazione la possibilità di migrarla in un ambiente separato.

Gestione delle directory virtuali

Il eb migrate comando preserva le strutture di directory virtuali durante la migrazione delle applicazioni IIS su Elastic Beanstalk.

Configurazione predefinita delle autorizzazioni

Durante la migrazione delle directory virtuali, eb migrate stabilisce un set di autorizzazioni di base concedendo l'accesso a: ReadAndExecute

- IIS_IUSRS
- IUSR
- Utenti autenticati

Ad esempio, si consideri una tipica struttura di directory virtuale:

```
<site name="CorporatePortal">
    <application path="/" applicationPool="CorporatePortalPool">
        <virtualDirectory path="/" physicalPath="C:\sites\portal" />
        <virtualDirectory path="/shared" physicalPath="C:\shared\content" />
        <virtualDirectory path="/reports" physicalPath="D:\reports" />
    </application>
```

```
</site>
```

Directory virtuali protette da password

Quando eb migrate incontra directory virtuali protette da password, emette avvisi e richiede un intervento manuale.

Il seguente esempio di configurazione genererà la risposta di avviso che segue l'esempio.

```
<virtualDirectory path="/secure"
                 physicalPath="C:\secure\content"
                 userName="DOMAIN\User"
                 password="[encrypted]" />
```

```
[WARNING] CorporatePortal/secure is hosted at C:\secure\content which is password-
protected and won't be copied.
```

Per mantenere la protezione tramite password, create uno script di distribuzione personalizzato come il seguente:

```
# PS C:\migrations_workspace> cat secure_vdir_config.ps1

$vdirPath = "C:\secure\content"
$siteName = "CorporatePortal"
$vdirName = "secure"
$username = "DOMAIN\User"
$password = "SecurePassword"

# Ensure directory exists
if (-not (Test-Path $vdirPath)) {
    Write-Host "Creating directory: $vdirPath"
    New-Item -Path $vdirPath -ItemType Directory -Force
}

# Configure virtual directory with credentials
Write-Host "Configuring protected virtual directory: $vdirName"
New-WebVirtualDirectory -Site $siteName -Name $vdirName `
    -PhysicalPath $vdirPath -UserName $username -Password $password

# Set additional permissions as needed
$acl = Get-Acl $vdirPath
```

```
$rule = New-Object System.Security.AccessControl.FileSystemAccessRule(
    $username, "ReadAndExecute", "ContainerInherit,ObjectInherit", "None", "Allow"
)
$acl.AddAccessRule($rule)
Set-Acl $vdirPath $acl
```

Aggiungi questo script alla tua distribuzione includendolo nel manifesto:

```
{
    "manifestVersion": 1,
    "deployments": {
        "custom": [
            {
                "name": "SecureVirtualDirectory",
                "scripts": {
                    "install": {
                        "file": "secure_vdir_config.ps1"
                    }
                }
            }
        ]
    }
}
```

Gestione delle autorizzazioni personalizzata

Il eb migrate comando fornisce una struttura per gli script di autorizzazione personalizzati per ospitare applicazioni che richiedono autorizzazioni diverse da quelle predefinite.

```
$paths = @(
    "C:\sites\portal\uploads",
    "C:\shared\content"
)

foreach ($path in $paths) {
    if (-not (Test-Path $path)) {
        Write-Host "Creating directory: $path"
        New-Item -Path $path -ItemType Directory -Force
    }

    $acl = Get-Acl $path
```

```
    # Add custom permissions
    $customRules = @(
        # Application Pool Identity - Full Control
        [System.Security.AccessControl.FileSystemAccessRule]::new(
            "IIS AppPool\CorporatePortalPool",
            "FullControl",
            "ContainerInherit,ObjectInherit",
            "None",
            "Allow"
        ),
        # Custom Service Account
        [System.Security.AccessControl.FileSystemAccessRule]::new(
            "NT SERVICE\CustomService",
            "Modify",
            "ContainerInherit,ObjectInherit",
            "None",
            "Allow"
        )
    )

    foreach ($rule in $customRules) {
        $acl.AddAccessRule($rule)
    }

    Set-Acl $path $acl
    Write-Host "Custom permissions applied to: $path"
}
```

Best practice

Segui queste best practice per pianificare, eseguire, monitorare e verificare la migrazione.

Pianificazione pre-migrazione

Documenta le autorizzazioni e i requisiti di autenticazione esistenti prima della migrazione. Prova gli script di autorizzazione personalizzati in un ambiente di sviluppo prima di distribuirli in produzione.

Gestione dei contenuti condivisi

Per le directory di contenuti condivisi, assicurati che tutte le autorizzazioni necessarie per il file system siano configurate correttamente tramite script personalizzati. Prendi in considerazione l'utilizzo di Amazon FSx for Windows File Server per requisiti di storage condiviso.

Monitoraggio e verifica

Monitora i log delle applicazioni dopo la migrazione per verificare il corretto accesso alle directory virtuali. Presta particolare attenzione alle seguenti aree:

- Accesso all'identità del pool di applicazioni
- Autorizzazioni personalizzate per gli account di servizio
- Connettività di condivisione di rete
- Errori di autenticazione

Impostazioni personalizzate del pool di applicazioni

Per impostazione predefinita, il eb migrate comando non copia le impostazioni personalizzate del pool di applicazioni. Per preservare le configurazioni personalizzate dei pool di applicazioni, seguite questa procedura per creare e applicare una sezione del manifesto personalizzata.

1. Crea un archivio dei tuoi artefatti di migrazione.

```
PS C:\migrations_workspace> eb migrate --archive
```

2. Crea uno PowerShell script personalizzato per configurare i pool di applicazioni.

```
# PS C:\migrations_workspace> cat .\migrations\latest\upload_target
\customize_application_pool_config.ps1

$configPath = "$env:windir\System32\inetsrv\config\applicationHost.config"

[xml]$config = Get-Content -Path $configPath

$newPoolXml = @"
<!-- Original IIS Configuration -->
<applicationPools>
    <add name="CustomPool"
         managedRuntimeVersion="v4.0"
         managedPipelineMode="Integrated">
        <processModel identityType="SpecificUser"
                    userName="AppPoolUser"
                    password="[encrypted]" />
        <recycling>
            <periodicRestart time="00:00:00">
                <schedule>
```

```
                    <add value="02:00:00" />
                    <add value="14:00:00" />
                </schedule>
            </periodicRestart>
        </recycling>
    </add>
</applicationPools>
"@
$newPoolXmlNode = [xml]$newPoolXml

# Find the applicationPools section
$applicationPools = $config.SelectSingleNode("//configuration/
system.applicationHost/applicationPools")

# Import the new node into the document
$importedNode = $config.ImportNode($newPoolXmlNode.DocumentElement, $true)
$applicationPools.AppendChild($importedNode)

# Save the changes
$config.Save($configPath)

Write-Host "ApplicationHost.config has been updated successfully."
```

3. Aggiorna il `aws-windows-deployment-manifest.json` file per includere lo script personalizzato.

```
{
    "manifestVersion": 1,
    "deployments": {
        ...
        "custom": [
            ...,
            {
                "name": "ModifyApplicationPoolConfig",
                "description": "Modify application pool configuration from source
 machine to remove",
                "scripts": {
                    "install": {
                        "file": "customize_application_pool_config.ps1"
                    },
                    "restart": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    },
```

```
                    "uninstall": {
                        "file": "ebmigrateScripts\\noop.ps1"
                    }
                }
            }
        ]
    }
}
```

4. Crea un ambiente con la directory di archivio aggiornata.

```
PS C:\migrations_workspace> eb migrate `
    --archive-dir '.\migrations\latest\upload_target\'
```

L'`--archive-dir`argomento dice eb migrate di usare il codice sorgente creato in precedenza, evitando la creazione di nuovi archivi.

Distribuzione delle versioni precedenti

eb migrateMantiene una cronologia delle migrazioni tramite directory con data e ora e versioni delle applicazioni in Elastic Beanstalk. Ogni migrazione crea un file zip unico che può essere distribuito se necessario.

```
PS C:\migrations_workspace> ls .\migrations\

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d----l        3/18/2025  10:34 PM                latest
d-----        3/16/2025   5:47 AM                migration_1742104049.479849
d-----        3/17/2025   9:18 PM                migration_1742246303.18056
d-----        3/17/2025   9:22 PM                migration_1742246546.565739
...
d-----        3/18/2025  10:34 PM                migration_1742337258.30742
```

Il collegamento `latest` simbolico punta sempre alla directory degli elementi di migrazione creata più di recente. Oltre ai registri delle applicazioni e degli errori pertinenti, ogni directory degli elementi di migrazione contiene anche un `upload_target.zip` file che puoi distribuire su Elastic Beanstalk.

```
PS C:\migrations_workspace> ls .\migrations\latest\
```

```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        3/18/2025  10:34 PM                upload_target
-a----        3/18/2025  10:34 PM          13137 application.log
-a----        3/18/2025  10:34 PM              0 error.log
-a----        3/18/2025  10:34 PM        1650642 upload_target.zip
```

Puoi distribuire il file utilizzando: `upload_target.zip` eb migrate

```
PS C:\migrations_workspace> eb migrate --zip .\migrations\latest\upload_target.zip
```

Risoluzione dei problemi e diagnostica

> **Prova Amazon Q Developer CLI per la risoluzione dei problemi assistita dall'intelligenza artificiale**
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Risoluzione dei problemi degli ambienti Elastic Beanstalk con Amazon Q Developer CLI nei blog. AWS

Questa sezione fornisce indicazioni per la risoluzione dei problemi più comuni che possono insorgere durante la migrazione delle applicazioni IIS su Elastic Beanstalk.

Associazione di una coppia di chiavi all'ambiente EC2

Puoi accedere in modo sicuro alle istanze Amazon Elastic Compute Cloud EC2 (Amazon) fornite per la tua applicazione Elastic Beanstalk con una coppia di chiavi Amazon. EC2 Per istruzioni sulla creazione di una coppia di chiavi, consulta Creating a Key Pair Using Amazon EC2 nella Amazon EC2 User Guide.

Specificando un nome chiave si ottiene l'eb migrateeffetto di associare l'ambiente Elastic Beanstalk alla coppia di chiavi. Per motivi di sicurezza, questa operazione non aprirà la porta 3389 nel gruppo di sicurezza delle istanze. EC2 È possibile associare gruppi EC2 di sicurezza aggiuntivi che consentono il traffico sulla porta 3389 fino a eb config dopo la migrazione iniziale.

```
PS C:\migrations_workspace> eb migrate  `
```

```
    --keyname "my-keypair"  `
    --verbose
```

Quando crei una coppia di chiavi, Amazon EC2 memorizza una copia della tua chiave pubblica. Se non hai più bisogno di usarlo per connetterti a nessuna istanza di ambiente, puoi eliminarlo da Amazon EC2. Per maggiori dettagli, consulta la sezione Eliminazione della coppia di chiavi nella Amazon EC2 User Guide.

Per ulteriori informazioni sulla connessione alle EC2 istanze Windows Amazon, consulta Connessione a un'istanza Windows.

Accesso ai log

L'EB CLI offre una eb logs funzionalità che è possibile utilizzare per recuperare i log da un ambiente Elastic Beanstalk senza accedere alle relative istanze. EC2 Dopo l'esecuzione dieb migrate, è possibile eseguire il eb logs --zip comando che scaricherà e salverà i log nella directory. `.elasticbeanstalk\logs`

In alternativa, puoi visualizzare i log tramite la console AWS Elastic Beanstalk. Per ulteriori informazioni, consulta Visualizzazione dei log delle istanze Amazon EC2 nell'ambiente Elastic Beanstalk.

Accesso agli artefatti lato client

Il eb migrate comando memorizza i log delle applicazioni e degli errori generati all'interno delle directory degli artefatti delle migrazionimsdeploy.

```
./migrations/
### latest -> migration_20240308_123456/
### migration_20240308_123456/
    ### application.log
    ### error.log
    ### upload_target\
```

Monitoraggio dello stato dell'ambiente

Elastic Beanstalk ti aiuta a monitorare lo stato utilizzando le funzionalità avanzate di monitoraggio dello stato. È un sistema automatizzato di monitoraggio dello stato sanitario che monitora continuamente lo stato operativo delle istanze dell'applicazione, sfruttando metriche integrate come l'utilizzo della CPU, la latenza, il numero di richieste e i codici di risposta.

Il sistema di monitoraggio dello stato utilizza un approccio basato su agenti per raccogliere dati a livello di istanza e si integra con la registrazione e gli avvisi in tempo reale. Elastic Load Balancing (ELB) e Auto Scaling rispondono dinamicamente ai cambiamenti dello stato di salute, garantendo elevata disponibilità e tolleranza agli errori. Le modalità di monitoraggio avanzate, tra cui report avanzati sullo stato di salute, forniscono una visibilità granulare sul comportamento delle applicazioni, abilitando la risoluzione proattiva della risoluzione dei problemi e meccanismi di ripristino automatici.

Esegui il eb health comando EB CLI per visualizzare lo stato dell'ambiente. Sono visualizzate le seguenti informazioni:

- Stato di integrità dell'istanza
- Metriche di risposta dell'applicazione
- Utilizzo delle risorse di sistema
- Eventi di implementazione recenti

EC2 ottimizzazione delle prestazioni

Per impostazione predefinita, eb migrate seleziona il tipo di istanza c5.2xlarge per fornire un'esperienza ottimale per la prima volta con Elastic Beanstalk. Puoi ignorare questo comportamento con l'argomento: --instance-type

```
PS C:\migrations_workspace> eb migrate `
    --instance-type "t3.large"
```

Per gli ambienti di produzione, considera questi fattori quando selezioni un tipo di istanza:

- Requisiti di memoria delle tue applicazioni
- Requisiti della CPU per l'elaborazione dei carichi di lavoro
- Esigenze prestazionali
- Obiettivi di ottimizzazione dei costi

Configurazione del volume EBS

Per impostazione predefinita, Elastic Beanstalk creerà solo un volume root `C:\` block-device () per l'ambiente. Puoi trasferire volumi di snapshot di Amazon Elastic Block Store aggiuntivi con l'--ebs-snapshotsopzione:

```
PS C:\migrations_workspace> eb migrate `
    --ebs-snapshots "snap-123456789abc"
```

Per esempi di come configurare le mappature dei dispositivi a blocchi con Elastic Beanstalk, consulta l'articolo del blog Customize Ephemeral and EBS Volumes in Elastic Beanstalk Environments.

Per le applicazioni con requisiti di storage elevati, considera le seguenti opzioni:

- Utilizzo di volumi EBS per dati persistenti
- Implementazione di Amazon S3 per contenuti statici
- Utilizzo di Amazon FSx for Windows File Server per file system condivisi

Problemi e soluzioni comuni

Evento: installazione mancante di Web Deploy

Se riscontri errori relativi al mancato rilevamento di Web Deploy, installa Web Deploy 3.6 o versione successiva da Microsoft Web Platform Installer. L'esempio seguente visualizza un possibile messaggio di errore.

```
Couldn't find msdeploy.exe. Follow instructions here: https://learn.microsoft.com/en-
us/iis/install/installing-publishing-technologies/installing-and-configuring-web-deploy
```

Evento: problemi di autorizzazione durante la migrazione

Se riscontri errori relativi alle autorizzazioni, assicurati di eseguire l'EB CLI con privilegi amministrativi. L'esempio seguente visualizza un possibile messaggio di errore.

```
[ERROR] Access to the path 'C:\inetpub\wwwroot\web.config' is denied.
```

Evento: problemi di identità del pool di applicazioni

Se l'applicazione non si avvia a causa di problemi di identità del pool di applicazioni, crea uno script personalizzato per configurare le identità del pool di applicazioni come mostrato inthe section called “Impostazioni personalizzate del pool di applicazioni”.

Evento: errori di configurazione del certificato SSL

Se le associazioni HTTPS non funzionano, assicurati di aver specificato un ARN di certificato ACM valido utilizzando il parametro option. eb mibrate `--ssl-certificates`

Evento: timeout per la creazione dell'ambiente

Se la creazione dell'ambiente scade, controlla CloudFormation gli eventi nella Console di AWS gestione per eventuali errori specifici nella creazione delle risorse. Le cause più comuni includono problemi di configurazione del VPC o limiti del servizio.

Ottenere supporto

Se riscontri problemi che non riesci a risolvere, prima di contattare Supporto AWS raccogli le seguenti informazioni:

- ID ambiente (`eb status`)
- Registri delle applicazioni () `eb logs --zip`
- Elementi di migrazione da `.\migrations\latest\`
- Configurazione IIS di origine (output di) `eb migrate explore --verbose`
- Messaggi di errore dettagliati

Per ulteriori informazioni sulla risoluzione dei problemi di Elastic Beanstalk, consulta. Risoluzione dei problemi dell'ambiente Elastic Beanstalk

Confronto delle opzioni di migrazione: EB CLI vs. AWS Transform MGN

AWS offre diversi percorsi per la migrazione delle applicazioni Windows sul cloud. Questa sezione confronta due opzioni principali: il eb migrate comando nella CLI EB AWS Transform MGN e (MGN). Comprendere le differenze tra questi approcci ti aiuterà a scegliere la strategia di migrazione più appropriata per le tue esigenze specifiche.

Confronto delle opzioni di migrazione

Funzionalità	CLI WEB () eb migrate	AWS Transform MGN (MGN)
Obiettivo primario	Migrazione a livello di applicazione di siti Web e applicazioni IIS	Rehosting a livello di server di intere macchine (server fisici, virtuali o cloud)

Funzionalità	CLI WEB () eb migrate	AWS Transform MGN (MGN)
Ideale per	Applicazioni IIS che desideri migrare direttamente su Elastic Beanstalk con una riconfigurazione minima	Migrazioni su larga scala che coinvolgono molti server o infrastru tture complesse
Approccio alla scoperta	Individuazione a livello di applicazi one di siti, applicazioni e configura zioni IIS	Replica a livello di server di intere macchine, inclusi sistema operativo e applicazioni
Ambiente di destinazione	Crea e configura direttamente ambienti Elastic Beanstalk ottimizzati per le applicazioni Windows	Crea EC2 istanze che richiedono una configurazione aggiuntiva per funzionare con Elastic Beanstalk
Conservazione della configura zione	Conserva automaticamente le configurazioni specifiche di IIS (siti, pool di applicazioni, associazioni)	Conserva l'intera configurazione del server, che può includere component i non necessari
Modello di implementazione	Crea un ambiente Elastic Beanstalk pulito con le applicazioni distribuite utilizzando le best practice di Elastic Beanstalk	Crea una replica del server di origine che potrebbe richiedere l'ottimiz zazione per le operazioni sul cloud
Scala di migrazione	Ideale per migrazioni mirate di applicazioni specifiche	Progettato per migrazioni su larga scala di molti server
Fasi successive alla migrazione	Minimo; l'ambiente è pronto per l'uso con gli strumenti di gestione Elastic Beanstalk	Richiede passaggi aggiuntivi per l'integrazione con Elastic Beanstalk , come l'esecuzione di azioni SSM post-lancio

Quando utilizzare ciascuna opzione di migrazione

Scegli eb migrate se hai i seguenti requisiti:

- Desideri migrare applicazioni IIS specifiche anziché interi server
- Il tuo obiettivo è adottare Elastic Beanstalk come piattaforma di gestione delle applicazioni

- Vuoi sfruttare le funzionalità della piattaforma gestita di Elastic Beanstalk, come la scalabilità, l'implementazione e il monitoraggio semplificati
- Preferisci un'implementazione pulita che AWS segua le migliori pratiche per le operazioni native del cloud
- Vuoi ridurre al minimo il lavoro di configurazione successivo alla migrazione

Scegli AWS Transform MGN se hai i seguenti requisiti:

- È necessario migrare un numero elevato di server
- Avete configurazioni di server complesse che devono essere preservate esattamente
- Le applicazioni presentano problemi di compatibilità che richiedono il mantenimento dell'esatto ambiente server
- Volete «lift-and-shift» apportando modifiche minime alle vostre applicazioni
- Avete intenzione di rifattorizzare o ottimizzare le vostre applicazioni dopo la migrazione

Confronto dei flussi di migrazione

Flusso di lavoro EB CLI (eb migrate):

1. Installa l'EB CLI sul tuo server IIS di origine o su un host bastion.
2. Esegui eb migrate per scoprire le applicazioni IIS.
3. Il comando raggruppa le applicazioni e le configurazioni.
4. Un ambiente Elastic Beanstalk viene creato con risorse appropriate.
5. Le tue applicazioni vengono distribuite nel nuovo ambiente.
6. Puoi gestire immediatamente le tue applicazioni utilizzando gli strumenti Elastic Beanstalk.

AWS Transform MGN flusso di lavoro:

1. Installare l'agente AWS di replica sui server di origine.
2. Configura e testa la replica dei dati.
3. Avvia istanze di test per verificare la funzionalità.
4. Pianifica il cutover per. AWS
5. Avvia istanze di produzione.
6. Esegui azioni post-lancio per ottimizzare per il cloud.

7. Se Elastic Beanstalk è la piattaforma di destinazione, è necessaria una configurazione aggiuntiva per l'integrazione con Elastic Beanstalk.

Conclusioni

Elastic Beanstalk è la destinazione preferita per le AWS applicazioni della piattaforma Windows e offre un ambiente gestito che semplifica l'implementazione, la scalabilità e la gestione. Il eb migrate comando fornisce un percorso diretto a Elastic Beanstalk per applicazioni IIS, con rilevamento e configurazione automatici che preservano le impostazioni dell'applicazione.

Sebbene AWS Transform MGN offra potenti funzionalità per le migrazioni di server su larga scala, richiede passaggi aggiuntivi per l'integrazione con Elastic Beanstalk. Per la maggior parte delle migrazioni di applicazioni IIS in cui Elastic Beanstalk è la eb migrate piattaforma di destinazione, offre un approccio più semplificato che si allinea al modello di servizio gestito di Elastic Beanstalk.

Scegli l'approccio di migrazione più adatto alle tue esigenze specifiche, considerando fattori come la scalabilità, la complessità e l'architettura dello stato finale desiderata. AWS

Per ulteriori informazioni su AWS Transform MGN, consulta What is? AWS Transform MGN nella Guida AWS Transform MGN per l'utente.

Risoluzione dei problemi dell'ambiente Elastic Beanstalk

Questo capitolo fornisce indicazioni per la risoluzione dei problemi con l'ambiente Elastic Beanstalk. Include le voci seguenti:

- Un'introduzione a AWS Systems Manager, inclusa una procedura per eseguire runbook Elastic Beanstalk predefiniti che forniscono istruzioni e suggerimenti per la risoluzione dei problemi.
- Linee guida generali per le azioni e le risorse da utilizzare in caso di peggioramento dello stato dell'ambiente.
- Suggerimenti specifici per la risoluzione dei problemi organizzati per categoria.

Se lo stato del tuo ambiente diventa rosso, ti consigliamo di utilizzare prima AWS Systems Manager lo strumento che include runbook predefiniti per risolvere i problemi di Elastic Beanstalk. Per ulteriori informazioni, consulta la Utilizzo dello strumento Systems Manager.

> Prova Amazon Q Developer CLI per la risoluzione dei problemi assistita dall'intelligenza artificiale
>
> La CLI per sviluppatori di Amazon Q può aiutarti a risolvere rapidamente i problemi ambientali. La Q CLI fornisce soluzioni controllando lo stato dell'ambiente, esaminando gli eventi, analizzando i log e ponendo domande di chiarimento. Per ulteriori informazioni e procedure dettagliate, consulta Risoluzione dei problemi degli ambienti Elastic Beanstalk con Amazon Q Developer CLI nei blog. AWS

Argomenti

- Utilizzo dell'analisi dell'ambiente basata sull'intelligenza artificiale
- Utilizzo dei AWS Systems Manager runbook Elastic Beanstalk
- Guida generale per la risoluzione dei problemi dell'ambiente Elastic Beanstalk
- Ambienti che accedono a segreti e parametri con variabili di ambiente
- Creazione dell'ambiente e avvii di istanze
- Distribuzioni
- Integrità

- Ambienti con sistemi di bilanciamento del carico configurati a doppio stack
- Configurazione
- Risoluzione dei problemi dei container Docker
- Domande frequenti
- Errori comuni
- Errori di distribuzione

Utilizzo dell'analisi dell'ambiente basata sull'intelligenza artificiale

Elastic Beanstalk fornisce analisi basate sull'intelligenza artificiale per aiutare a identificare le cause profonde e consigliare soluzioni per i problemi di salute dell'ambiente. Puoi accedere a questa funzionalità tramite il pulsante AI Analysis della console, utilizzando l'API Elastic Beanstalk AWS CLI tramite o eb logs --analyze eseguendo con l'EB CLI. L'analisi esamina i log, gli eventi e lo stato delle istanze dell'ambiente per identificare i problemi critici e fornire consigli per la risoluzione dei problemi. step-by-step

Per ulteriori informazioni, consulta AI-powered analisi dell'ambiente.

Utilizzo dei AWS Systems Manager runbook Elastic Beanstalk

Puoi utilizzare Systems Manager per risolvere i problemi degli ambienti Elastic Beanstalk. Per aiutarti a iniziare rapidamente, Systems Manager fornisce runbook di automazione predefiniti per Elastic Beanstalk. Un runbook di automazione è un tipo di documento di Systems Manager che definisce le operazioni da eseguire sulle istanze dell'ambiente e su altre risorse AWS .

Il documento `AWSSupport-TroubleshootElasticBeanstalk` è un runbook di automazione progettato per contribuire a identificare una serie di problemi comuni che possono degradare l'ambiente Elastic Beanstalk. A tal fine, controlla i componenti del tuo ambiente, tra cui: istanze EC2, VPC, CloudFormation stack, load balancer, gruppi di Auto Scaling e configurazione di rete associata alle regole dei gruppi di sicurezza, tabelle di routing e. ACLs

Fornisce inoltre un'opzione per caricare file di log in bundle dal tuo ambiente al Supporto AWS .

Per ulteriori informazioni, consulta `AWSSupport-TroubleshootElasticBeanstalk` nella Documentazione di riferimento del runbook di Automazione AWS Systems Manager .

Utilizzo di Systems Manager per eseguire il runbook **`AWSSupport-TroubleshootElasticBeanstalk`**

> **Note**
>
> Esegui questa procedura nello stesso Regione AWS luogo in cui si trova l'ambiente Elastic Beanstalk.

1. Aprire la console AWS Systems Manager.
2. Nel riquadro di navigazione a sinistra, nella sezione Gestione delle modifiche, scegli Automazione.
3. Scegli Esegui automazione.
4. Nella scheda Proprietà di Amazon, nella casella di ricerca Documenti di automazione inserisci `AWSSupport-TroubleshootElasticBeanstalk`.
5. Seleziona la AWSSupport-TroubleshootElasticBeanstalkscheda, quindi scegli Avanti.
6. Seleziona Esegui.
7. Nella sezione Parametri di input:
   a. Dal menu a AutomationAssumeRolediscesa, seleziona l'ARN del ruolo che consente a Systems Manager di eseguire azioni per tuo conto.
   b. Per ApplicationName, inserisci il nome dell'applicazione Elastic Beanstalk.
   c. In Nome dell'ambiente, inserisci l'ambiente Elastic Beanstalk.
   d. (Facoltativo) Per S3 UploaderLink, inserisci un link se un AWS Support Engineer ti ha fornito un link S3 per la raccolta dei log.
8. Scegli Esegui.

   Se uno dei passaggi non riesce, seleziona il link sotto la colonna ID fase per il passaggio non riuscito. Viene visualizzata una pagina con i Dettagli dell'esecuzione per il passaggio. La VerificationErrorMessagesezione mostrerà un riepilogo dei passaggi che richiedono attenzione. Ad esempio, `IAMPermissionCheck` potrebbe visualizzare un messaggio di avviso. In questo caso, puoi verificare che il ruolo selezionato nel menu a AutomationAssumeRolediscesa disponga delle autorizzazioni necessarie.

Dopo avere completato con successo tutti i passaggi, l'output fornisce i passaggi per la risoluzione dei problemi e i consigli per ripristinare lo stato di integrità dell'ambiente.

Guida generale per la risoluzione dei problemi dell'ambiente Elastic Beanstalk

I messaggi di errore possono essere visualizzati nella pagina Eventi della console, nei log o nella pagina dello stato. È inoltre possibile intraprendere azioni per ripristinare un ambiente danneggiato in seguito a una modifica recente. Se lo stato dell'ambiente passa al rosso, prova a fare quanto segue:

- Richiedi un'analisi del tuo ambiente basata sull'intelligenza artificiale. Per ulteriori informazioni, consulta AI-powered analisi dell'ambiente.
- Se un'operazione sul tuo ambiente restituisce un errore contenente il testo`The stack stack_id associated with environment environment-ID is in stack-status state`, consulta la sezione Ripristino dell'ambiente Elastic Beanstalk da uno stato non valido per una guida alla risoluzione dei problemi.
- Se un'operazione sull'ambiente restituisce un errore che contiene il testo`Environment environment-name associated CloudFormation stack stack_arn does not exist`, chiudi l'ambiente e creane un altro.
- Esamina gli eventi più recenti dell'ambiente. Spesso qui vengono visualizzati i messaggi di Elastic Beanstalk relativi alla distribuzione, al caricamento e ai problemi di configurazione.
- Rivedi la cronologia delle modifiche recenti dell'ambiente. La cronologia delle modifiche elenca tutte le modifiche di configurazione apportate ai tuoi ambienti e include altre informazioni, ad esempio quale utente IAM ha apportato modifiche e quali parametri di configurazione sono stati impostati.
- Esegui il pull dei log per visualizzare le voci recenti del file di log. I log del server Web contengono informazioni sugli errori e sulle richieste in entrata.
- Connettiti a un'istanza e controlla le risorse di sistema.
- Esegui il roll back a una versione precedente e funzionante dell'applicazione.
- Annulla le modifiche di configurazione recenti o ripristina una configurazione salvata.
- Distribuisci un nuovo ambiente. Se l'ambiente sembra integro, esegui uno scambio di CNAME per instradare il traffico verso il nuovo ambiente e continuare a eseguire il debug di quello precedente.

Ambienti che accedono a segreti e parametri con variabili di ambiente

Evento: la distribuzione dell'istanza non è riuscita a ottenere uno o più segreti

Questo messaggio indica che Elastic Beanstalk non è stato in grado di recuperare uno o più segreti specificati durante la distribuzione dell'applicazione.

- Verifica che le risorse specificate dai valori ARN nella configurazione della variabile di ambiente esistano.
- Verifica che il ruolo del profilo dell'istanza Elastic Beanstalk EC2 disponga delle autorizzazioni IAM necessarie per accedere alle risorse.
- Se questo evento è stato attivato dall'`RestartAppServer`operazione, una volta risolto il problema, riprova a effettuare la chiamata per risolverlo. `RestartAppServer`
- Se l'evento è stato attivato tramite una `UpdateEnvironment` chiamata, riprova l'operazione. `UpdateEnvironment`

Per esempi di questi comandi, vedi AWS CLI esempi per Elastic Beanstalk. Per ulteriori informazioni sulle azioni API per queste operazioni, consulta l'AWS Elastic Beanstalk API Reference.

Evento: la distribuzione dell'istanza ha rilevato uno o più valori di ambiente multilinea, che non sono supportati per questa piattaforma

Le variabili multilinea non sono supportate per le piattaforme Amazon Linux 2, escluse le piattaforme Docker e Docker gestite da ECS. Per le opzioni disponibili per procedere, consulta. Valori multilinea

Evento: CreateEnvironment fallisce quando viene specificato un segreto

In caso di `CreateEnvironment` errore e si dispone di segreti come variabili di ambiente, è necessario risolvere il problema sottostante e quindi `UpdateEnvironment` utilizzarlo per completare la configurazione dell'ambiente. Non utilizzarlo`RestartAppServer`, poiché non sarà sufficiente a far apparire l'ambiente in questa situazione. Per esempi di questi comandi, vedi AWS CLI esempi per Elastic Beanstalk. Per ulteriori informazioni sulle azioni API per queste operazioni, consulta l'AWS Elastic Beanstalk API Reference.

Creazione dell'ambiente e avvii di istanze

Evento: Impossibile avviare l'ambiente

Questo evento si verifica quando Elastic Beanstalk tenta di avviare un ambiente e incontra delle operazioni non riuscite durante il percorso. Gli eventi precedenti nella pagina Events (Eventi) contribuiranno a definire la causa alla base di questo problema.

Evento: l'operazione di creazione dell'ambiente è completa, ma con timeout del comando. Prova ad aumentare la durata del timeout.

L'applicazione potrebbe richiedere molto tempo per la distribuzione se utilizzi i file di configurazione che eseguono i comandi sull'istanza, scarichi file di grandi dimensioni o installi pacchetti. Aumenta il timeout del comando per dare più tempo alla tua applicazione di avviare l'esecuzione durante le distribuzioni.

Evento: Impossibile creare le seguenti risorse: [AWSEBInstanceLaunchWaitCondition]

Questo messaggio indica che le istanze Amazon EC2 dell'ambiente non hanno comunicato a Elastic Beanstalk di essere state avviate correttamente. Questa situazione può verificarsi se le istanze non dispongono di una connessione a Internet. Se hai configurato l'ambiente per avviare istanze in una sottorete privata del VPC, verifica che la sottorete abbia un NAT per consentire alle istanze di connettersi a Elastic Beanstalk.

Evento: in questa regione è obbligatorio un ruolo di servizio. Aggiungi un'opzione di ruolo di servizio all'ambiente.

Elastic Beanstalk utilizza un ruolo di servizio per monitorare le risorse nell'ambiente e supportare aggiornamenti gestiti della piattaforma. Per ulteriori informazioni, consulta Gestione dei ruoli del servizio Elastic Beanstalk.

Distribuzioni

Problema: l'applicazione non è più disponibile durante le distribuzioni

Poiché Elastic Beanstalk utilizza un processo di aggiornamento drop-in, potrebbero verificarsi alcuni secondi di inattività. Utilizza le distribuzioni in sequenza per minimizzare l'effetto delle distribuzioni nei tuoi ambienti di produzione.

Evento: creazione della versione dell'applicazione AWS Elastic Beanstalk non riuscita

Il bundle di origine dell'applicazione può essere troppo grande o potrebbe aver raggiunto la quota della versione dell'applicazione.

Evento: l'operazione di aggiornamento dell'ambiente è completa, ma con timeout del comando. Prova ad aumentare la durata del timeout.

L'applicazione potrebbe richiedere molto tempo per la distribuzione se utilizzi i file di configurazione che eseguono i comandi sull'istanza, scarichi file di grandi dimensioni o installi pacchetti. Aumenta il timeout del comando per dare più tempo alla tua applicazione di avviare l'esecuzione durante le distribuzioni.

Integrità

Evento: l'utilizzo della CPU supera il 95,00%

Prova a eseguire più istanze oppure scegli un altro tipo di istanza.

Evento: Elastic Load Balancer awseb- *myapp* Ha zero istanze integre, i processi ELB non sono integri su istanze fuori. *X Y* , I processi ELB non sono integri in tutte le istanze.

Questi messaggi indicano che i controlli dello stato del bilanciamento del carico non riescono su una o più istanze. Se l'applicazione sembra funzionare, assicurati che l'URL del controllo dello stato dell'applicazione sia configurato correttamente. In caso contrario, prova quanto segue:

- Controlla i registri delle applicazioni (ad esempio`/var/log/web.stdout.log`) e verifica `/var/log/eb-engine.log` la presenza di errori, errori di avvio o problemi di distribuzione.
- Controlla i log del proxy per eventuali errori di connessione in `/var/log/nginx/` (nginx) o (Apache). `/var/log/httpd/`
- Verifica che l'applicazione risponda con HTTP 200 al percorso di controllo dello stato configurato e sia in ascolto sulla porta prevista.
- Verifica che i gruppi di sicurezza consentano il traffico in entrata dal sistema di bilanciamento del carico alle istanze sulla porta di controllo dello stato. Nella console EC2, controlla le regole in entrata sul gruppo di sicurezza dell'istanza per confermare che la porta di health check sia aperta al gruppo di sicurezza del load balancer.
- Controlla l'utilizzo della CPU e della memoria nelle CloudWatch metriche (,). `CPUUtilization` `EnvironmentHealth` L'esaurimento delle risorse può causare timeout per i controlli sanitari.

Per ulteriori dettagli, recupera i log completi del pacchetto o abilita lo streaming dei log. CloudWatch Puoi anche eseguire il runbook di `AWSSupport-TroubleshootElasticBeanstalk`automazione per facilitare la diagnosi del problema.

Evento: Elastic Load Balancer awseb- Impossibile trovare *myapp*

Il sistema di bilanciamento del carico dell'ambiente potrebbe essere stato rimosso. out-of-band Apporta modifiche alle risorse dell'ambiente solo con le opzioni di configurazione e l'estensibilità fornite da Elastic Beanstalk. Ricrea l'ambiente o avviane uno nuovo.

Evento: avvio dell'istanza EC2 non riuscito. In attesa dell'avvio di una nuova istanza EC2...

La disponibilità di tipi di istanze per il tuo ambiente potrebbe essere ridotta oppure potresti aver raggiunto la quota di istanze per il tuo account. Verifica il pannello di controllo dello stato dei servizi per assicurarti che il servizio Elastic Compute Cloud (Amazon EC2) sia verde oppure richiedi un aumento della quota.

Evento: nessuna delle istanze invia dati.

Elastic Beanstalk non riceve dati sanitari da nessuna delle tue istanze. Ciò può essere causato da problemi di connettività di rete, problemi relativi agli operatori sanitari o errori a livello di sistema.

- Verifica la connettività di rete. Verifica che i gruppi di sicurezza consentano l'HTTPS in uscita (porta 443). Per le istanze in sottoreti private, verifica che siano configurati un gateway NAT o un endpoint VPC. Controlla le tabelle delle rotte per le rotte in uscita corrette e assicurati che la risoluzione DNS funzioni correttamente nel tuo VPC.
- Controlla l'agente di monitoraggio dello stato. `/var/log/healthd/daemon.log`Cerca gli errori dell'agente sanitario e gli errori `/var/log/messages` a livello di sistema. Verifica che il servizio healthd sia in esecuzione connettendoti all'istanza tramite SSH ed eseguendolo. `systemctl status healthd`
- Verifica la presenza di problemi a livello di sistema come l'esaurimento delle risorse (memoria, CPU, spazio su disco), la sincronizzazione NTP/system dell'orologio e la connettività del servizio di metadati dell'istanza.

Per ulteriori dettagli, recupera i log completi del pacchetto o abilita lo streaming dei log. CloudWatch
Puoi anche eseguire il runbook di `AWSSupport-TroubleshootElasticBeanstalk`automazione per facilitare la diagnosi del problema.

Ambienti con sistemi di bilanciamento del carico configurati a doppio stack

Evento: in VPC *vpc_id* non sono configurati blocchi IPv6 CIDR. IPv6 I blocchi CIDR sono necessari per il bilanciamento del carico dualstack. Associa un blocco IPv6 CIDR al tuo VPC prima di utilizzare la modalità dualstack.

Al tuo VPC e a tutte le sottoreti devono essere associati dei blocchi IPv6 CIDR. Questo è uno dei prerequisiti del VPC da completare prima di configurare il sistema di bilanciamento del carico per il supporto dualstack. Per ulteriori informazioni sul completamento di questa attività, vedere la parte precedente di questo argomento. Prerequisiti di Amazon VPC

Evento: una o più sottoreti per VPC *vpc_id* non hanno IPv6 blocchi CIDR configurati. IPv6 I blocchi CIDR sono necessari per le sottoreti utilizzate con il sistema di bilanciamento del carico dualstack. Associa i blocchi IPv6 CIDR a tutte le sottoreti richieste prima di utilizzare la modalità dualstack.

Tutte le sottoreti del tuo VPC devono avere blocchi IPv6 CIDR associati. Questo è uno dei prerequisiti del VPC da completare prima di configurare il sistema di bilanciamento del carico per il supporto dualstack. Per ulteriori informazioni sul completamento di questa attività, vedere la parte precedente di questo argomento. Prerequisiti di Amazon VPC

Errore: l'**`IpAddressType`**opzione può essere applicata solo agli ambienti Elastic Beanstalk configurati con Application Load Balancer o Network Load Balancer.

Questo messaggio indica che l'ambiente Elastic Beanstalk potrebbe essere un ambiente a istanza singola o che potrebbe utilizzare un Classic Load Balancer. È possibile configurare solo gli ambienti configurati con Application Load Balancer o Network Load Balancer. `IpAddressType`

Configurazione

Evento: lo stack *stack_id* associato all'ambiente *environment-ID* è in stato *stack-status*

Lo CloudFormation stack sottostante dell'ambiente potrebbe trovarsi nello stato *_FAILED. È necessario correggere questo stato per continuare le operazioni di Elastic Beanstalk nell'ambiente. Per ulteriori informazioni, consulta Ripristino dell'ambiente Elastic Beanstalk da uno stato non valido.

Evento: non puoi configurare un ambiente Elastic Beanstalk con valori sia per l'opzione Target (Destinazione) che per l'opzione Application Healthcheck URL (URL controllo dello stato applicazione) di Elastic Load Balancing.

L'opzione `Target` nello spazio dei nomi `aws:elb:healthcheck` è obsoleta. Rimuovi lo spazio dei nomi dell'opzione `Target` dal tuo ambiente e riprova l'aggiornamento.

Evento: ELB non può essere collegato a più sottoreti nella stessa AZ

Questo messaggio può essere restituito se tenti di spostare un sistema di bilanciamento del carico tra sottoreti nella stessa zona di disponibilità. La modifica di sottoreti nel sistema di bilanciamento del carico ne richiede lo spostamento al di fuori delle zone di disponibilità d'origine e quindi la reintegrazione nella zona di disponibilità d'origine con le sottoreti desiderate. Durante il processo, tutte le istanze verranno migrate da una all'altra AZs, causando tempi di inattività significativi. Valuta quindi la possibilità di creare un nuovo ambiente ed esegui uno scambio di CNAME.

Risoluzione dei problemi dei container Docker

Evento: estrazione dell'immagine Docker non riuscita :latest: nome del repository non valido (), sono consentiti soltanto i caratteri [a-z0-9-_.]. Consulta i log per ulteriori dettagli.

Controlla la sintassi del file `dockerrun.aws.json` utilizzando un convalidatore JSON. Inoltre verifica il contenuto del dockerfile rispetto ai requisiti descritti in Preparazione dell'immagine Docker per la distribuzione su Elastic Beanstalk

Evento: No EXPOSE directive found in Dockerfile, abort deployment

Il `Dockerfile` o il file `dockerrun.aws.json` non dichiara la porta del container. Utilizza l'istruzione `EXPOSE` (`Dockerfile`) o il blocco `Ports` (file `dockerrun.aws.json`) per esporre una porta per il traffico in entrata.

Evento: Impossibile scaricare le credenziali di autenticazione da *`repository bucket name`*

`dockerrun.aws.json`Fornisce un bucket and/or S3 per la coppia di chiavi EC2 non valido per il file. `.dockercfg` Oppure, il profilo dell'istanza non dispone dell' GetObject autorizzazione per il bucket S3. Verifica che il file `.dockercfg` includa una coppia di chiavi EC2 e un bucket S3 validi. Concedi le autorizzazioni per l'operazione `s3:GetObject` al ruolo IAM nel profilo dell'istanza. Per i dettagli, vedi Gestione dei profili dell'istanza Elastic Beanstalk

Evento: Activity execution failed, because: WARNING: Invalid auth configuration file

Il file di autenticazione (`config.json`) non è formattato correttamente. Per informazioni, consultare Autenticazione con archivi di immagini.

Domande frequenti

Domanda: Come posso modificare l'URL della mia applicazione da myapp.us-west-2.elasticbeanstalk.com a www.myapp.com?

In un server DNS, registra un record CNAME, ad esempio **`www.mydomain.com CNAME mydomain.elasticbeanstalk.com`**.

Domanda: Come si specifica la zona di disponibilità per l'applicazione Elastic Beanstalk?

Puoi scegliere una zona di disponibilità specifica utilizzando il plug-in CLI APIs, Eclipse o Visual Studio. Per istruzioni su come usare la console Elastic Beanstalk per specificare una zona di disponibilità, consulta Auto Scaling delle istanze dell'ambiente Elastic Beanstalk.

Domanda: Come posso modificare il tipo di istanza del mio ambiente?

Per modificare il tipo di istanza dell'ambiente, vai alla pagina di configurazione dell'ambiente e scegli Edit (Modifica) nella categoria di configurazione Instances (Istanze). Quindi, seleziona un nuovo tipo di istanza e scegli Apply (Applica) per aggiornare l'ambiente. Successivamente, Elastic Beanstalk terminerà tutte le istanze in esecuzione e le sostituirà con quelle nuove.

Domanda: Come posso verificare se qualcuno ha apportato modifiche alla configurazione di un ambiente?

Per visualizzare queste informazioni, nel riquadro di navigazione della console Elastic Beanstalk scegli Change history (Cronologia delle modifiche) per visualizzare un elenco delle modifiche alla configurazione per tutti gli ambienti. Questo elenco include la data e l'ora della modifica, il parametro di configurazione modificato e il nuovo valore, nonché l'utente IAM che ha apportato la modifica. Per ulteriori informazioni, consulta Change history (Cronologia delle modifiche).

Domanda: È possibile evitare che i volumi Amazon EBS vengano eliminati quando le istanze vengono terminate?

Le istanze nell'ambiente usano Amazon EBS per lo storage, ma il volume root viene eliminato quando un'istanza viene terminata da Auto Scaling. Non è consigliabile archiviare lo stato o altri dati sulle istanze. Se necessario, puoi impedire l'eliminazione dei volumi con AWS CLI: `$ aws ec2 modify-instance-attribute -b '/dev/sdc=<vol-id>:false` come descritto nella Guida di riferimento.AWS CLI

Domanda: Come si eliminano i dati personali dall'applicazione Elastic Beanstalk?

AWS le risorse utilizzate dall'applicazione Elastic Beanstalk potrebbero archiviare informazioni personali. Quando termini un ambiente, Elastic Beanstalk termina le risorse che ha creato. Vengono terminate anche le risorse che hai aggiunto tramite i file di configurazione. Tuttavia, se hai creato AWS risorse al di fuori dell'ambiente Elastic Beanstalk e le hai associate all'applicazione, potresti dover controllare manualmente che le informazioni personali che l'applicazione potrebbe aver archiviato non vengano conservate. In questa guida per gli sviluppatori, ogni volta che parliamo della creazione di risorse aggiuntive, specifichiamo anche quando dovresti considerare la possibilità di eliminarle.

Errori comuni

In questo argomento sono elencati i messaggi di errore più comuni riscontrati quando si usa l'interfaccia a riga di comando EB con l'indicazione delle soluzioni possibili. Se si verifica un messaggio di errore non mostrato qui, utilizza i link Feedback per comunicarcelo.

ERRORE: si è verificato un errore durante la gestione del comando git. Codice di errore: 128 Errore: fatale: nome di oggetto HEAD non valido

Causa: questo messaggio di errore viene visualizzato se hai inizializzato un archivio Git ma non hai ancora eseguito il commit. L'interfaccia a riga di comando EB cerca la revisione HEAD quando la cartella di progetto contiene un archivio Git.

Soluzione: aggiungi i file nella tua cartella di progetto con l'area di gestione temporanea ed esegui il commit:

```
~/my-app$ git add .
~/my-app$ git commit -m "First commit"
```

ERRORE: questo ramo non ha un ambiente predefinito. È necessario specificare un ambiente digitando «eb status my-env-name" o impostare un ambiente predefinito digitando «eb use my-env-name».

Causa: quando si crea un nuovo ramo git, questo non è collegato a un ambiente di Elastic Beanstalk per impostazione predefinita.

Soluzione: esegui eb list per visualizzare un elenco di ambienti disponibili. Quindi esegui eb use ***env-name*** per utilizzare uno degli ambienti disponibili.

ERRORE: le piattaforme 2.0+ richiedono un ruolo di servizio. Puoi fornirne uno con l'opzione --service-role

Causa: se specifichi un nome di ambiente con eb create (ad esempio, eb create my-env), l'interfaccia a riga di comando EB non tenterà di creare automaticamente un ruolo di servizio. Se non hai il ruolo di servizio predefinito, viene visualizzato l'errore riportato sopra.

Soluzione: esegui eb create senza un nome di ambiente e segui le istruzioni per creare il ruolo di servizio predefinito.

Errori di distribuzione

La distribuzione di Elastic Beanstalk potrebbe rispondere con una risposta 404 (se l'applicazione non è riuscita ad avviarsi) o 500 (se l'applicazione fallisce durante il runtime). Per risolvere molti problemi comuni, puoi utilizzare l'EB CLI per controllare lo stato della distribuzione, visualizzarne i log, accedere alla tua istanza EC2 con SSH o aprire la pagina della Console di AWS gestione per il tuo ambiente applicativo.

Per utilizzare l'interfaccia a riga di comando EB per aiutare a risolvere i problemi di distribuzione

1. Esegui eb status per vedere lo stato della distribuzione corrente e l'integrità dell'host EC2. Esempio:

```
$ eb status --verbose

Environment details for: python_eb_app
  Application name: python_eb_app
  Region: us-west-2
  Deployed Version: app-150206_035343
  Environment ID: e-wa8u6rrmqy
  Platform: 64bit Amazon Linux 2014.09 v1.1.0 running Python 2.7
  Tier: WebServer-Standard-
  CNAME: python_eb_app.elasticbeanstalk.com
  Updated: 2015-02-06 12:00:08.557000+00:00
  Status: Ready
  Health: Green
  Running instances: 1
      i-8000528c: InService
```

> **Note**
>
> Utilizzando l'opzione `--verbose`, si ottengono informazioni sullo stato delle istanze in esecuzione. Senza di esso, eb status stamperà solo le informazioni generali sull'ambiente.

2. Esegui eb health per visualizzare le informazioni sullo stato del tuo ambiente:

```
$ eb health --refresh
 elasticBeanstalkExa-env                                Degraded
 2016-03-28 23:13:20
WebServer
    Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        2        0        2        1        0        0        0

  instance-id   status     cause
    Overall     Degraded  Incorrect application version found on 3 out of 5
 instances. Expected version "Sample Application" (deployment 1).
  i-d581497d    Degraded  Incorrect application version "v2" (deployment 2).
 Expected version "Sample Application" (deployment 1).
  i-d481497c    Degraded  Incorrect application version "v2" (deployment 2).
 Expected version "Sample Application" (deployment 1).
  i-136e00c0    Severe    Instance ELB health has not been available for 5 minutes.
  i-126e00c1    Ok
  i-8b2cf575    Ok

  instance-id   r/sec    %2xx   %3xx   %4xx   %5xx      p99      p90      p75
 p50     p10
    Overall     646.7   100.0    0.0    0.0    0.0    0.003    0.002    0.001
 0.001   0.000
  i-dac3f859    167.5    1675      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-05013a81    161.2    1612      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-04013a80    0.0         -      -      -      -        -        -       -
 -       -
  i-3ab524a1    155.9    1559      0      0      0    0.003    0.002    0.001
 0.001   0.000
  i-bf300d3c    162.1    1621      0      0      0    0.003    0.002    0.001
 0.001   0.000
```

```
 instance-id   type       az   running     load 1  load 5      user%  nice%
system%  idle%   iowait%
 i-d581497d    t2.micro   1a   25 mins       0.16     0.1        7.0    0.0
1.7   91.0       0.1
 i-d481497c    t2.micro   1a   25 mins       0.14     0.1        7.2    0.0
1.6   91.1       0.0
 i-136e00c0    t2.micro   1b   25 mins        0.0    0.01        0.0    0.0
0.0   99.9       0.1
 i-126e00c1    t2.micro   1b   25 mins       0.03    0.08        6.9    0.0
2.1   90.7       0.1
 i-8b2cf575    t2.micro   1c   1 hour        0.05    0.41        6.9    0.0
2.0   90.9       0.0

 instance-id   status     id   version              ago
      deployments
 i-d581497d    Deployed   2    v2                   9 mins
 i-d481497c    Deployed   2    v2                   7 mins
 i-136e00c0    Failed     2    v2                   5 mins
 i-126e00c1    Deployed   1    Sample Application   25 mins
 i-8b2cf575    Deployed   1    Sample Application   1 hour
```

L'esempio precedente mostra un ambiente con cinque istanze in cui la distribuzione della versione "v2" non è riuscita alla terza istanza. Dopo un'installazione non riuscita, viene ripristinata l'ultima versione riuscita che, in questo caso, è la "Sample Application" della prima della distribuzione. Per ulteriori informazioni, consulta Utilizzo dell'interfaccia a riga di comando EB per monitorare lo stato dell'ambiente.

3. Esegui eb logs per scaricare e visualizzare i log associati alla distribuzione dell'applicazione.

```
$ eb logs
```

4. Esegui eb ssh per connetterti all'istanza di EC2 in esecuzione sull'applicazione ed esaminarla direttamente. Nell'istanza l'applicazione distribuita si trova nella directory `/opt/python/current/app` e l'ambiente Python si trova in `/opt/python/run/venv/`.
5. Esegui eb console per visualizzare l'ambiente applicativo nella console di gestione AWS. È possibile utilizzare l'interfaccia Web per esaminare diversi aspetti della distribuzione, tra cui la configurazione dell'applicazione, lo stato, gli eventi, i log. Puoi anche scaricare la versione corrente o quelle precedenti delle applicazioni che hai distribuito al server.

Risorse Elastic Beanstalk

Le seguenti risorse correlate possono rivelarsi utili durante l'utilizzo di questo servizio.

- Riferimento all'API Elastic Beanstalk Una descrizione completa di tutti i SOAP e le Query. APIs Inoltre, contiene un elenco di tutti i tipi di dati SOAP.
- elastic-beanstalk-samples on GitHub — Un GitHub repository con file di configurazione di esempio di Elastic Beanstalk (.ebextensions). Il `README.md` file del repository contiene collegamenti a repository aggiuntivi con applicazioni di esempio. GitHub
- Domande tecniche frequenti su Elastic Beanstalk: principali domande poste dagli sviluppatori in merito a questo prodotto.
- AWS Elastic Beanstalk Note di rilascio: dettagli su nuove funzionalità, aggiornamenti e correzioni nelle versioni del servizio, della piattaforma, della console e dell'EB CLI di Elastic Beanstalk.

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo SDKs, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- Supporto AWS Center: l'hub per la creazione e la gestione dei casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.
- Supporto— La pagina web principale per informazioni su Supporto one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contattaci: un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .

- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Applicazioni di esempio

Di seguito sono elencati i link per il download delle applicazioni di esempio che vengono distribuite come parte di Scopri come iniziare a usare Elastic Beanstalk.

> **Note**
>
> In alcuni esempi vengono utilizzate caratteristiche che potrebbero essere state rilasciate successivamente rispetto alla versione della piattaforma che stai utilizzando. Se si verifica un errore nell'esecuzione dell'esempio, prova ad aggiornare la piattaforma a una versione corrente, come descritto in the section called “Piattaforme supportate”.

- Docker: docker.zip
- Docker Multicontainer — 2.zip docker-multicontainer-v
- Docker preconfigurato (Glassfish) — 1.zip docker-glassfish-v
- Go: go.zip
- Corretto: corretto.zip
- Tomcat: tomcat.zip
- .NET Core su Linux — .zip dotnet-core-linux
- .NET Core — dotnet-asp-windows .zip
- Node.js: nodejs.zip
- PHP: php.zip
- Python: python.zip
- Ruby: ruby.zip

AWS SDK per Java

AWS SDK per Java Fornisce un'API Java che è possibile utilizzare per creare applicazioni che utilizzano servizi di AWS infrastruttura. Con AWS SDK per Java, puoi iniziare in pochi minuti con un unico pacchetto scaricabile che include la libreria AWS Java, esempi di codice e documentazione.

AWS SDK per Java Richiede il J2SE Development Kit. Puoi scaricare il software Java più recente da http://developers.sun.com/downloads/. L'SDK richiede anche pacchetti di terze parti Apache Commons (Codec e Logging) e Saxon-HE, inclusi nella directory di terze parti dell'SDK. HTTPClient

Per ulteriori informazioni, consulta SDK AWS per Java. Per ulteriori informazioni sull'SDK, sul codice di esempio, sulla documentazione, sugli strumenti e sulle risorse aggiuntive, consulta Java on Developer Center. AWS

AWS SDK per .NET

Con AWS SDK per .NET, puoi iniziare rapidamente con un unico pacchetto scaricabile che include la AWS libreria.NET, esempi di codice e documentazione.

Per ulteriori informazioni, consulta AWS SDK for .NET. Per sapere quali versioni di .NET Framework e Visual Studio sono supportate, consulta Guida per lo sviluppatore AWS SDK per .NET.

Per ulteriori informazioni sull'SDK, sul codice di esempio, sulla documentazione, sugli strumenti e sulle risorse aggiuntive, consulta .NET on AWS Developer Center.

AWS Toolkit for Visual Studio

Con il AWS Toolkit for Visual Studio plug-in, puoi distribuire un'applicazione.NET esistente su Elastic Beanstalk. È inoltre possibile creare progetti utilizzando i AWS modelli preconfigurati con. AWS SDK per .NET

Per informazioni sui prerequisiti e sull'installazione, consulta AWS Toolkit for Visual Studio.
Per iniziare la creazione dell'applicazione Elastic Beanstalk utilizzando Visual Studio, consulta Distribuzione di applicazioni.NET Windows con Elastic Beanstalk.

AWS SDK per JavaScript in Node.js

Con l' AWS SDK per JavaScript Node.js, puoi iniziare rapidamente con un unico pacchetto scaricabile che include la libreria AWS Node.js, esempi di codice e documentazione.

Per ulteriori informazioni, consulta l'AWS SDK per Node.js. JavaScript

AWS SDK per PHP

Con AWS SDK per PHP, puoi iniziare rapidamente con un unico pacchetto scaricabile che include la libreria AWS PHP, esempi di codice e documentazione.

Per ulteriori informazioni, consulta la pagina relativa a SDK AWS per PHP. Per ulteriori informazioni sull'SDK, sul codice di esempio, sulla documentazione, sugli strumenti e sulle risorse aggiuntive, consulta il PHP on Developer Center. AWS

AWS SDK per Python (Boto)

Con AWS SDK per Python (Boto), puoi iniziare rapidamente con un unico pacchetto scaricabile che include la libreria AWS Python, esempi di codice e documentazione. Inoltre, puoi creare applicazioni Python per eliminare APIs la complessità della codifica direttamente sulle interfacce dei servizi web.

La all-in-one libreria Python è intuitiva per gli sviluppatori APIs e nasconde molte delle attività di livello inferiore associate alla programmazione per il AWS Cloud, tra cui l'autenticazione, i nuovi tentativi di richiesta e la gestione degli errori. L'SDK fornisce esempi pratici in Python su come utilizzare le librerie per creare applicazioni.

Per informazioni su Boto, codice di esempio, documentazione, strumenti e risorse aggiuntive, consulta Python AWS on Developer Center.

AWS SDK per Ruby

Con la libreria AWS Ruby, gli esempi di codice e la documentazione, è possibile creare applicazioni Ruby sfruttando al contempo la complessità della APIs codifica direttamente sulle interfacce dei servizi Web.

La all-in-one libreria Ruby è intuitiva per gli sviluppatori APIs e nasconde molte delle attività di livello inferiore associate alla programmazione per il AWS cloud, tra cui l'autenticazione, i nuovi tentativi di richiesta e la gestione degli errori. L'SDK fornisce esempi pratici in Ruby su come utilizzare le librerie per creare applicazioni.

Per informazioni sull'SDK, sul codice di esempio, sulla documentazione, sugli strumenti e sulle risorse aggiuntive, consultate Ruby on Developer Center. AWS

Contenuti archiviati per la AWS Elastic Beanstalk Developer Guide

A scopo di riferimento, questo capitolo fornisce informazioni di configurazione su alcune funzionalità legacy di Elastic Beanstalk.

Argomenti

- Piattaforme personalizzate Elastic Beanstalk (ritirate)

Piattaforme personalizzate Elastic Beanstalk (ritirate)

> **Note**
>
> Il 18 luglio 2022, Elastic Beanstalk ha impostato lo stato di tutte le filiali della piattaforma basate su Amazon Linux AMI AL1 () come ritirato. Include le piattaforme personalizzate. Elastic Beanstalk non supporta le piattaforme personalizzate. Per ulteriori informazioni sul ritiro da parte di Elastic Beanstalk dell'AMI Amazon Linux, consulta Domande frequenti sul ritiro della piattaforma.

Questo argomento rimarrà in questo documento come riferimento per tutti i clienti che hanno utilizzato la funzionalità della piattaforma personalizzata Elastic Beanstalk prima del suo ritiro. In passato, le piattaforme personalizzate Elastic Beanstalk supportavano la creazione di un'AMI da Amazon Linux AMI, RHEL 7, RHEL 6 o Ubuntu 16.04 base. AMIs Questi sistemi operativi non sono più supportati da Elastic Beanstalk. Per ulteriori informazioni sulla funzionalità delle piattaforme personalizzate, che non è più supportata, espandi il seguente argomento.

Piattaforme personalizzate

Una piattaforma personalizzata è una personalizzazione più avanzata rispetto a una immagine personalizzata in diversi modi. Una piattaforma personalizzata ti consente di sviluppare un'intera nuova piattaforma da zero, personalizzando il sistema operativo, il software aggiuntivo e gli script che vengono eseguiti da Elastic Beanstalk sulle istanze della piattaforma. Questa flessibilità ti consente di creare una piattaforma per un'applicazione che utilizza un linguaggio o altro software

dell'infrastruttura per cui Elastic Beanstalk non fornisce una piattaforma gestita. Confrontala con immagini personalizzate, in cui modifichi un'Amazon Machine Image (AMI) per l'uso con una piattaforma Elastic Beanstalk esistente ed Elastic Beanstalk continua a fornire gli script della piattaforma e a controllare lo stack del software della piattaforma. Inoltre, con le piattaforme personalizzate utilizzi un metodo automatizzato e con script per creare e gestire la personalizzazione, mentre con le immagini personalizzate apporti le modifiche manualmente su un'istanza in esecuzione.

Per creare una piattaforma personalizzata devi creare un'AMI da uno dei sistemi operativi supportati: Ubuntu, RHEL o Amazon Linux (vedi la voce `flavor` in Formato del file Platform.yaml per i numeri di versione esatti) e aggiungere ulteriori personalizzazioni. Puoi creare la tua piattaforma Elastic Beanstalk utilizzando Packer, uno strumento open source per la creazione di immagini di macchine per molte piattaforme, AMIs incluso l'uso con Amazon Elastic Compute Cloud (Amazon). EC2 Una piattaforma Elastic Beanstalk comprende un'AMI configurata per eseguire un set di software che supporta un'applicazione e metadati che possono includere opzioni di configurazione personalizzate e impostazioni di opzioni di configurazione predefinite.

Elastic Beanstalk gestisce Packer come una piattaforma integrata separata e non dovrai preoccuparti della configurazione e delle versioni di Packer.

Puoi creare una piattaforma fornendo a Elastic Beanstalk un modello Packer e gli script e i file che il modello richiama per creare un'AMI. Questi componenti sono compressi in un file di definizione della piattaforma che specifica il modello e i metadati in un archivio ZIP noto come archivio di definizione della piattaforma.

Quando crei una piattaforma personalizzata, avvii un ambiente con una singola istanza senza un IP elastico che esegue Packer. Quindi, Packer avvia un'altra istanza per creare un'immagine. Puoi riutilizzare questo ambiente per più piattaforme e più versioni di ciascuna piattaforma.

> **Note**
>
> Le piattaforme personalizzate sono specifiche della regione. AWS Se utilizzi Elastic Beanstalk in più regioni devi creare le tue piattaforme separatamente in ciascuna regione.
> In determinate circostanze, le istanze avviate da Packer non vengono eliminate e devono essere terminate manualmente. Per informazioni su come pulire manualmente queste istanze, consulta Pulizia delle istanze Packer.

Gli utenti nel tuo account possono utilizzare le piattaforme personalizzate specificando un ARN della piattaforma durante la creazione dell'ambiente. Queste ARNs vengono restituite dal eb platform create comando utilizzato per creare la piattaforma personalizzata.

Ogni volta che compili la tua piattaforma personalizzata, Elastic Beanstalk crea una nuova versione della piattaforma. Gli utenti possono specificare una piattaforma per nome per ottenerne solo la versione più recente o includere un numero di versione per ottenere una versione specifica.

Ad esempio, per distribuire la versione più recente della piattaforma personalizzata con l'ARN **`MyCustomPlatformARN`**, che potrebbe corrispondere alla versione 3.0, la riga di comando dell'interfaccia a riga di comando EB deve essere simile alla seguente:

```
eb create -p MyCustomPlatformARN
```

Per distribuire la versione 2.1, la riga di comando della CLI EB deve essere simile alla seguente:

```
eb create -p MyCustomPlatformARN --version 2.1
```

Puoi applicare tag a una versione della piattaforma personalizzata al momento della sua creazione e modificare i tag di versioni della piattaforma esistenti. Per informazioni dettagliate, vedi Tagging delle versioni della piattaforma personalizzate.

Creazione di una piattaforma personalizzata

Per creare una piattaforma personalizzata, la root della tua applicazione deve includere un file di definizione della piattaforma `platform.yaml` che definisce il tipo di generatore utilizzato per creare la piattaforma personalizzata. Il formato di questo file è descritto in Formato del file Platform.yaml. Puoi creare la tua piattaforma personalizzata da zero oppure utilizzare una delle piattaforme personalizzate di esempio come punto di partenza.

Utilizzo di una piattaforma personalizzata di esempio

Un'alternativa alla creazione di una piattaforma personalizzata consiste nell'usare uno degli esempi dell'archivio di definizione della piattaforma per eseguire il bootstrap della piattaforma personalizzata. Gli unici elementi che devi configurare negli esempi prima di poterli utilizzare sono un'AMI sorgente e una regione.

> **Note**
>
> Non utilizzare una piattaforma personalizzata di esempio non modificata in produzione. Gli esempi hanno lo scopo di mostrare alcune delle funzionalità disponibili per una piattaforma personalizzata, tuttavia non sono stati consolidati per l'uso in produzione.

NodePlatform_Ubuntu.zip

Questa piattaforma personalizzata si basa su Ubuntu 16.04 e supporta Node.js 4.4.4. Utilizziamo questa piattaforma personalizzata per gli esempi in questa sezione.

NodePlatform_RHEL.zip

Questa piattaforma personalizzata si basa su RHEL 7.2 e supporta Node.js 4.4.4.

NodePlatformAmazonLinux_.zip

Questa piattaforma personalizzata si basa su Amazon Linux 2016.09.1 e supporta Node.js 4.4.4.

TomcatPlatform_Ubuntu.zip

Questa piattaforma personalizzata si basa su Ubuntu 16.04 e supporta Tomcat 7/Java 8.

CustomPlatformNodeSampleApp_.zip

Esempio Node.js che usa express ed ejs per visualizzare una pagina Web statica.

CustomPlatform_ .zip TomcatSampleApp

Esempio Tomcat che visualizza una pagina Web statica al momento della distribuzione.

Scarica l'archivio di definizione della piattaforma di esempio: `NodePlatform_Ubuntu.zip`. Questo file contiene un file di definizione della piattaforma, un modello Packer, gli script eseguiti da Packer durante la creazione di immagini e gli script e i file di configurazione che Packer copia nell'istanza del generatore durante la creazione della piattaforma.

Example NodePlatform_Ubuntu.zip

```
|-- builder                 Contains files used by Packer to create the custom platform
|-- custom_platform.json    Packer template
|-- platform.yaml           Platform definition file
```

```
|-- ReadMe.txt              Briefly describes the sample
```

Il file di definizione della piattaforma `platform.yaml` indica a Elastic Beanstalk il nome del modello Packer `custom_platform.json`.

```
version: "1.0"

provisioner:
  type: packer
  template: custom_platform.json
  flavor: ubuntu1604
```

Il modello Packer indica a Packer come creare il file AMIs per la piattaforma, utilizzando un'AMI Ubuntu come base per l'immagine della piattaforma per i tipi di istanze HVM. La sezione `provisioners` indica a Packer di copiare nell'istanza tutti i file della cartella `builder` all'interno dell'archivio e di eseguire lo script `builder.sh` sull'istanza. Una volta completati gli script, Packer crea un'immagine dall'istanza modificata.

Elastic Beanstalk crea tre variabili di ambiente che possono essere utilizzate per etichettare in Packer: AMIs

AWS_EB_PLATFORM_ARN

L'ARN della piattaforma personalizzata.

AWS_EB_NOME_PIATTAFORMA

Il nome della piattaforma personalizzata.

AWS_EB_VERSIONE_PIATTAFORMA

La versione della piattaforma personalizzata.

Il file `custom_platform.json` di esempio utilizza queste variabili per definire i seguenti valori utilizzati negli script:

- `platform_name`, impostato da `platform.yaml`
- `platform_version`, impostato da `platform.yaml`
- `platform_arn`, impostato dallo script di compilazione principale, `builder.sh`, mostrato alla fine del file `custom_platform.json` di esempio.

Il file `custom_platform.json` contiene due proprietà di cui è necessario fornire i valori, ovvero `source_ami` e `region`. Per i dettagli sulla scelta dei valori AMI e Region corretti, consulta Aggiornamento del modello Packer nel eb-custom-platforms-samples GitHub repository.

Example custom_platform.json

```
{
  "variables": {
    "platform_name": "{{env `AWS_EB_PLATFORM_NAME`}}",
    "platform_version": "{{env `AWS_EB_PLATFORM_VERSION`}}",
    "platform_arn": "{{env `AWS_EB_PLATFORM_ARN`}}"
  },
  "builders": [
    {
      ...
      "region": "",
      "source_ami": "",
      ...
    }
  ],
  "provisioners": [
    {...},
    {
      "type": "shell",
      "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo {{ .Path }}",
      "scripts": [
        "builder/builder.sh"
      ]
    }
  ]
}
```

Gli script e altri file inclusi nell'archivio di definizione della piattaforma possono variare notevolmente a seconda delle modifiche che desideri apportare all'istanza. La piattaforma di esempio include i seguenti script:

- `00-sync-apt.sh`: impostato come commento `apt -y update`. Abbiamo commentato il comando perché richiede l'intervento dell'utente, che interrompe l'aggiornamento automatico del pacchetto. Questo potrebbe essere un problema di Ubuntu. Tuttavia, l'esecuzione di `apt -y update` è ancora consigliata come best practice. Per questo motivo, abbiamo lasciato il comando nello script di esempio a scopo di riferimento.

- `01-install-nginx.sh`: installa nginx.
- `02-setup-platform.sh`: installa `wget`, `tree` e `git`. Copia hook e configurazioni della registrazione nell'istanza e crea le seguenti directory:
  - `/etc/SampleNodePlatform`: la posizione in cui il file di configurazione del container viene caricato durante la distribuzione.
  - `/opt/elasticbeanstalk/deploy/appsource/`: la posizione in cui lo script `00-unzip.sh` carica il codice sorgente dell'applicazione durante la distribuzione. Per informazioni su questo script, consultare la sezione Strumenti di script di piattaforma per i tuoi ambienti Elastic Beanstalk.
  - `/var/app/staging/`: la posizione in cui il codice sorgente dell'applicazione viene elaborato durante la distribuzione.
  - `/var/app/current/`: la posizione in cui il codice sorgente dell'applicazione viene eseguito dopo l'elaborazione.
  - `/var/log/nginx/healthd/`: la posizione in cui l'agente di stato migliorato scrive i log.
  - `/var/nodejs`: la posizione in cui i file Node.js vengono caricati durante la distribuzione.

Utilizza l'EB CLI per creare la tua prima piattaforma personalizzata con l'archivio di definizione della piattaforma di esempio.

Per creare una piattaforma personalizzata

1. Installa l'interfaccia a riga di comando EB.
2. Crea una directory in cui estrarre la piattaforma personalizzata di esempio.

```
~$ mkdir ~/custom-platform
```

3. Estrai `NodePlatform_Ubuntu.zip` nella directory e successivamente inseriscilo nella directory estratta.

```
~$ cd ~/custom-platform
~/custom-platform$ unzip ~/NodePlatform_Ubuntu.zip
~/custom-platform$ cd NodePlatform_Ubuntu
```

4. Modifica il file `custom_platform.json` e fornisci i valori per le proprietà `source_ami` e `region`. Per ulteriori dettagli, consulta la sezione relativa all'aggiornamento del modello Packer.
5. Esegui eb platform init e segui le istruzioni per l'inizializzazione di un archivio della piattaforma.

Puoi abbreviare eb platform in ebp.

> **Note**
>
> Windows PowerShell lo utilizza ebp come alias di comando. Se stai eseguendo la CLI EB in Windows PowerShell, usa la forma lunga di questo comando:. eb platform

```
~/custom-platform$ eb platform init
```

Questo comando, inoltre, crea la directory `.elasticbeanstalk` nella directory corrente e aggiunge il file di configurazione `config.yml` alla directory. Non modificare o eliminare questo file perché Elastic Beanstalk lo utilizza durante la creazione della piattaforma personalizzata.

Per impostazione predefinita, eb platform init usa il nome della cartella corrente come nome della piattaforma personalizzata, ovvero `custom-platform` in questo esempio.

6. Esegui eb platform create per avviare un ambiente Packer e ottenere l'ARN della piattaforma personalizzata. Avrai bisogno di questo valore in un secondo momento, durante la creazione di un ambiente dalla piattaforma personalizzata.

```
~/custom-platform$ eb platform create
...
```

Per impostazione predefinita, Elastic Beanstalk crea il profilo di istanza `aws-elasticbeanstalk-custom-platform-ec2-role` per le piattaforme personalizzate. Se, invece, desideri utilizzare un profilo dell'istanza esistente, aggiungi l'opzione `-ip` *`INSTANCE_PROFILE`* al comando eb platform create.

> **Note**
>
> Packer non è in grado di creare una piattaforma personalizzata se utilizzi il profilo dell'istanza predefinito Elastic Beanstalk `aws-elasticbeanstalk-ec2-role`.

L'interfaccia a riga di comando EB mostra l'evento generato dall'ambiente Packer finché non viene completata la generazione. Puoi uscire dalla visualizzazione dell'evento premendo Ctrl+C.

7. Puoi controllare i log alla ricerca di errori utilizzando il comando eb platform logs.

```
~/custom-platform$ eb platform logs
...
```

8. Puoi controllare il processo in un secondo momento con eb platform events.

```
~/custom-platform$ eb platform events
...
```

9. Controlla lo stato della tua piattaforma con eb platform status.

```
~/custom-platform$ eb platform status
...
```

Al termine dell'operazione avrai una piattaforma che potrai utilizzare per avviare un ambiente Elastic Beanstalk.

Puoi utilizzare la piattaforma personalizzata durante la creazione di un ambiente dalla console. Per informazioni, consulta Procedura guidata per la creazione del nuovo ambiente.

Per avviare un ambiente sulla tua piattaforma personalizzata

1. Crea una directory per l'applicazione.

```
~$ mkdir custom-platform-app
~$ cd ~/custom-platform-app
```

2. Inizializza un archivio dell'applicazione.

```
~/custom-platform-app$ eb init
...
```

3. Scarica l'applicazione di esempio NodeSampleApp.zip.
4. Estrai l'applicazione di esempio.

```
~/custom-platform-app$ unzip ~/NodeSampleApp.zip
```

5. Eseguieb create -p ***CUSTOM-PLATFORM-ARN***, *CUSTOM-PLATFORM-ARN* dov'è l'ARN restituito da un eb platform create comando, per avviare un ambiente che esegue la tua piattaforma personalizzata.

```
~/custom-platform-app$ eb create -p CUSTOM-PLATFORM-ARN
...
```

Contenuti dell'archivio di definizione della piattaforma

Un archivio di definizione della piattaforma è l'equivalente della piattaforma di un bundle di origine dell'applicazione. L'archivio di definizione della piattaforma è un file ZIP che contiene un file di definizione della piattaforma, un modello Packer e gli script e i file utilizzati dal modello Packer per creare la piattaforma.

> **Note**
>
> Se utilizzata per creare una piattaforma personalizzata, l'interfaccia a riga di comando EB crea un archivio di definizione della piattaforma dal file e dalle cartelle nell'archivio della piattaforma, perciò non devi necessariamente creare l'archivio manualmente.

Il file di definizione della piattaforma è un file in formato YAML che deve essere denominato `platform.yaml` e trovarsi nella directory root dell'archivio di definizione della piattaforma. Consultare Creazione di una piattaforma personalizzata per un elenco delle chiavi obbligatorie e facoltative supportate in un file di definizione della piattaforma.

Non devi denominare il modello Packer in un modo specifico, ma il nome del file deve corrispondere al modello dello strumento di provisioning specificato nel file di definizione della piattaforma. Per istruzioni su come creare modelli Packer, vedi la documentazione di Packer ufficiale.

Gli altri file nell'archivio di definizione della piattaforma sono script e file utilizzati dal modello per personalizzare l'istanza prima di creare un'AMI.

Hook della piattaforma personalizzata

Elastic Beanstalk usa una struttura di directory standardizzata per gli hook su piattaforme personalizzate. Si tratta di script che vengono eseguiti durante gli eventi del ciclo di vita e in risposta alle operazioni di gestione: quando le istanze nell'ambiente vengono avviate oppure quando un utente avvia una distribuzione o usa la funzionalità di riavvio del server applicazioni.

Inserisci gli script che desideri vengano attivati dagli hook in una delle sottocartelle della cartella `/opt/elasticbeanstalk/hooks/`.

> **Warning**
>
> L'utilizzo di hook della piattaforma personalizzati su piattaforme gestite non è supportato. Gli hook della piattaforma personalizzati sono progettati per piattaforme personalizzate. Sulle piattaforme gestite Elastic Beanstalk potrebbero funzionare in modo diverso o presentare alcuni problemi e il comportamento potrebbe variare tra le piattaforme. Sulle piattaforme AMI Amazon Linux (precedenti ad Amazon Linux 2) potrebbero in alcuni casi ancora funzionare in modi utile, usali con cautela.
>
> Gli hook di piattaforma personalizzati sono una caratteristica legacy delle piattaforme AMI Amazon Linux. Sulle piattaforme Amazon Linux 2, gli hook di piattaforma personalizzati nella cartella `/opt/elasticbeanstalk/hooks/` sono stati completamente interrotti. Elastic Beanstalk non li legge né li esegue. Le piattaforme Amazon Linux 2 supportano un nuovo tipo di hook di piattaforma, specificamente progettato per estendere le piattaforme gestite Elastic Beanstalk. Puoi aggiungere script e programmi personalizzati direttamente a una directory di hook nel bundle di origine dell'applicazione. Elastic Beanstalk li esegue durante varie fasi di provisioning delle istanze. Per ulteriori informazioni, espandere la sezione Hook della piattaforma in the section called “Estensione delle piattaforme Linux”.

Gli hook sono organizzati nelle seguenti cartelle:

- `appdeploy`: gli script vengono eseguiti durante la distribuzione di un'applicazione. Elastic Beanstalk esegue la distribuzione di un'applicazione durante il lancio di nuove istanze e quando un client avvia la distribuzione di una nuova versione.
- `configdeploy`: gli script vengono eseguiti quando un client esegue un aggiornamento della configurazione che influisce sulla configurazione del software sull'istanza, ad esempio, impostando le proprietà dell'ambiente o abilitando la rotazione dei log su Amazon S3.
- `restartappserver`: gli script vengono eseguiti quando un client esegue un'operazione di riavvio del server di applicazione.
- `preinit`: gli script vengono eseguiti durante il processo di bootstrap dell'istanza.
- `postinit`: gli script vengono eseguiti dopo il processo di bootstrap dell'istanza.

Le cartelle `appdeploy`, `configdeploy` e `restartappserver` contengono le sottocartelle `pre`, `enact` e `post`. In ciascuna fase di un'operazione, tutti gli script nella cartella `pre` vengono eseguiti in ordine alfabetico, quindi vengono eseguiti gli script nella cartella `enact` e infine quelli nella cartella `post`.

Quando viene lanciata un'istanza, Elastic Beanstalk esegue `preinit`, `appdeploy` e `postinit` in questo ordine. Su distribuzioni successive nelle istanze in esecuzione, Elastic Beanstalk esegue gli hook `appdeploy`. Gli hook `configdeploy` vengono eseguiti quando un utente aggiorna le impostazioni di configurazione del software sull'istanza. Gli hook `restartappserver` vengono eseguiti solo quando l'utente inizia il riavvio del server dell'applicazione.

Quando riscontrano errori, i tuoi script possono uscire con uno stato diverso da zero e scrivere in `stderr` per far fallire l'operazione. Il messaggio che scrivi in `stderr` apparirà nell'evento generato quando l'operazione ha esito negativo. Elastic Beanstalk è inoltre in grado di acquisire queste informazioni nel file di log `/var/log/eb-activity.log`. Se non desideri che l'operazione abbia esito negativo, restituisci 0. I messaggi che scrivi in `stderr` o `stdout` vengono visualizzati nei log di distribuzione, ma non nel flusso di eventi, a meno che l'operazione non abbia esito negativo.

Pulizia delle istanze Packer

In determinate circostanze, ad esempio nel caso di interruzione del processo del generatore Packer prima della conclusione, le istanze avviate da Packer non vengono eliminate. Queste istanze non fanno parte dell'ambiente Elastic Beanstalk e possono essere visualizzate e terminate solo utilizzando il servizio Amazon. EC2

Per eliminare manualmente queste istanze

1. Apri la EC2 console Amazon.
2. Assicurati di trovarti nella stessa AWS regione in cui hai creato l'istanza con Packer.
3. In Risorse, scegli *N* Running Instances, dove *N* indica il numero di istanze in esecuzione.
4. Fare clic nella casella di testo di query.
5. Selezionare il tag Name (Nome).
6. Immettere packer.

   L'aspetto della query deve essere simile a: tag:Name: packer
7. Selezionare le istanze che corrispondono alla query.

8. Se Instance State (Stato istanza) è running (In esecuzione), scegliere Actions (Operazioni), Instance State (Stato istanza), Stop (Arresta), quindi Actions (Operazioni), Instance State (Stato istanza), Terminate (Termina).

Formato del file Platform.yaml

Il file `platform.yaml` presenta il formato seguente.

```
version: "version-number"

provisioner:
   type: provisioner-type
   template: provisioner-template
   flavor: provisioner-flavor

metadata:
   maintainer: metadata-maintainer
   description: metadata-description
   operating_system_name: metadata-operating_system_name
   operating_system_version: metadata-operating_system_version
   programming_language_name: metadata-programming_language_name
   programming_language_version: metadata-programming_language_version
   framework_name: metadata-framework_name
   framework_version: metadata-framework_version

option_definitions:
   - namespace: option-def-namespace
     option_name: option-def-option_name
     description: option-def-description
     default_value: option-def-default_value

option_settings:
   - namespace: "option-setting-namespace"
     option_name: "option-setting-option_name"
     value: "option-setting-value"
```

Sostituisci i segnaposto con i questi valori:

`version-number`

Obbligatorio. La versione della definizione YAML. Deve essere **`1.0`**.

provisioner-type

Obbligatorio. Il tipo di generatore utilizzato per creare la piattaforma personalizzata. Deve essere **packer**.

provisioner-template

Obbligatorio. Il file JSON contenente le impostazioni per. *provisioner-type*

provisioner-flavor

Opzionale. Il sistema operativo di base utilizzato per l'AMI. Una delle seguenti:

amazon (predefinito)

Amazon Linux. Se non è specificato, la versione più recente di Amazon Linux al momento della creazione della piattaforma.

Amazon Linux 2 non è un sistema operativo supportato.

ubuntu1604

Ubuntu 16.04 LTS

rhel7

RHEL 7

rhel6

RHEL 6

metadata-maintainer

Opzionale. Informazioni di contatto per la persona che possiede la piattaforma (100 caratteri).

metadata-description

Opzionale. Descrizione della piattaforma (2.000 caratteri).

metadata-operating_system_name

Opzionale. Nome del sistema operativo della piattaforma (50 caratteri). Questo valore è disponibile quando si filtra l'output per l'ListPlatformVersionsAPI.

metadata-operating_system_version

Opzionale. Versione del sistema operativo della piattaforma (20 caratteri).

metadata-programming_language_name

Opzionale. Linguaggio di programmazione supportato dalla piattaforma (50 caratteri)

metadata-programming_language_version

Opzionale. Versione del linguaggio della piattaforma (20 caratteri).

metadata-framework_name

Opzionale. Nome del framework Web utilizzato dalla piattaforma (50 caratteri).

metadata-framework_version

Opzionale. Versione del framework Web della piattaforma (20 caratteri).

option-def-namespace

Opzionale. Uno spazio dei nomi in `aws:elasticbeanstalk:container:custom` (100 caratteri)

option-def-option_name

Opzionale. Il nome dell'opzione (100 caratteri). È possibile definire fino a un massimo di 50 opzioni di configurazione personalizzate che la piattaforma fornisce agli utenti.

option-def-description

Opzionale. Descrizione dell'opzione (1.024 caratteri).

option-def-default_value

Opzionale. Il valore predefinito utilizzato quando l'utente non ne specifica uno.

L'esempio seguente crea l'opzione **`NPM_START`**.

```
options_definitions:
 -  namespace: "aws:elasticbeanstalk:container:custom:application"
    option_name: "NPM_START"
    description: "Default application startup command"
    default_value: "node application.js"
```

option-setting-namespace

Opzionale. Spazio dei nomi dell'opzione.

option-setting-option_name

Opzionale. Nome dell'opzione. Puoi specificare fino a 50 opzioni fornite da Elastic Beanstalk.

option-setting-value

Opzionale. Il valore utilizzato quando l'utente non ne specifica uno.

L'esempio seguente crea l'opzione **TEST**.

```
option_settings:
 - namespace: "aws:elasticbeanstalk:application:environment"
   option_name: "TEST"
   value: "This is a test"
```

Tagging delle versioni della piattaforma personalizzate

Puoi applicare tag alle versioni AWS Elastic Beanstalk personalizzate della tua piattaforma. I tag sono coppie chiave-valore associate AWS alle risorse. Per informazioni sul tagging delle risorse, sui casi d'uso, sui vincoli delle chiavi e dei valori di tag e sui tipi di risorse supportati di Elastic Beanstalk, consulta Tagging delle risorse dell'applicazione Elastic Beanstalk.

Puoi specificare i tag al momento della creazione di una versione della piattaforma personalizzata. In una versione della piattaforma personalizzata esistente, puoi aggiungere o rimuovere i tag e aggiornare i valori dei tag esistenti. Puoi aggiungere fino a 50 tag per ogni versione della piattaforma personalizzata.

Aggiunta di tag durante la creazione della versione della piattaforma personalizzata

Se utilizzi la CLI EB per creare una versione della piattaforma personalizzata, utilizza l'opzione `--tags` con eb platform create per aggiungere i tag.

```
~/workspace/my-app$ eb platform create --tags mytag1=value1,mytag2=value2
```

Con AWS CLI o con altri client basati su API, aggiungi tag utilizzando il `--tags` parametro sul comando. create-platform-version

```
$ aws elasticbeanstalk create-platform-version \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
```

```
      --platform-name my-platform --platform-version 1.0.0 --platform-definition-bundle
 S3Bucket=amzn-s3-demo-bucket,S3Key=sample.zip
```

Gestione dei tag di una versione della piattaforma personalizzata esistente

Puoi aggiungere, aggiornare ed eliminare tag in una versione della piattaforma personalizzata Elastic Beanstalk esistente.

Se utilizzi la CLI EB per aggiornare la versione della piattaforma personalizzata, utilizza eb tags per aggiungere, aggiornare, eliminare o elencare i tag.

Ad esempio, il comando seguente elenca i tag in una versione della piattaforma personalizzata.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-
account-id:platform/my-platform/1.0.0"
```

Il comando seguente aggiorna il tag `mytag1` ed elimina il tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2 \
      --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-
platform/1.0.0"
```

Per un elenco completo delle opzioni e per altri esempi, consulta `eb tags`.

Con lo AWS CLI o altri client basati su API, utilizzate il list-tags-for-resource comando per elencare i tag di una versione personalizzata della piattaforma.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-platform/1.0.0"
```

Utilizza il comando update-tags-for-resource per aggiungere, aggiornare o eliminare i tag in una versione della piattaforma personalizzata.

```
$ aws elasticbeanstalk update-tags-for-resource \
      --tags-to-add Key=mytag1,Value=newvalue --tags-to-remove mytag2 \
      --resource-arn "arn:aws:elasticbeanstalk:us-east-2:my-account-id:platform/my-
platform/1.0.0"
```

Specifica i tag da aggiungere e i tag da aggiornare nel parametro `--tags-to-add` di update-tags-for-resource. È stato aggiunto un tag non esistente e il valore di un tag esistente è stato aggiornato.

> **Note**
>
> Per utilizzare alcuni AWS CLI comandi e CLI di EB con una versione della piattaforma personalizzata Elastic Beanstalk, è necessario l'ARN della versione personalizzata della piattaforma. È possibile recuperare l'ARN utilizzando il seguente comando.
>
> ```
> $ aws elasticbeanstalk list-platform-versions
> ```
>
> Utilizza l'opzione `--filters` per filtrare l'output per il nome della piattaforma personalizzata.

Cronologia dei documenti

La tabella seguente descrive le importanti modifiche alla AWS Elastic Beanstalk Developer Guide dall'aprile 2024.

Modifica	Descrizione	Data
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkWorkerTier`	Autorizzazioni aggiornate nella policy AWS gestita per aggiungere autorizzazioni Amazon Nova e AWS Marketplace per l'analisi dell'ambiente basata sull'intelligenza artificiale.	29 aprile 2026
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkWebTier`	Autorizzazioni aggiornate nella policy AWS gestita per aggiungere autorizzazioni Amazon Nova e AWS Marketplace per l'analisi dell'ambiente basata sull'intelligenza artificiale.	29 aprile 2026
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkMulticontainerDocker`	Autorizzazioni aggiornate nella policy AWS gestita per aggiungere autorizzazioni Amazon Nova e AWS Marketplace per l'analisi dell'ambiente basata sull'intelligenza artificiale.	29 aprile 2026
Argomento rivisto: i registri di distribuzione sono ora disponibili sulle piattaforme Windows Server	I registri di distribuzione sono ora supportati nelle versioni della piattaforma Windows Server.	22 aprile 2026

Argomento rivisto: l'analisi dell'ambiente basata sull'intelligenza artificiale è ora disponibile sulle piattaforme Windows Server	L'analisi dell'ambiente basata sull'intelligenza artificiale è ora supportata nelle versioni della piattaforma Windows Server.	22 aprile 2026
Argomento rivisto: prerequisiti per l'analisi dell'ambiente basata sull'intelligenza artificiale	È stato aggiunto il requisito di inviare il modulo con i dettagli del caso d'uso di Anthropic in Amazon Bedrock prima di utilizzare l'analisi dell'ambiente basata sull'intelligenza artificiale.	24 marzo 2026
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy`	Autorizzazioni aggiornate nella policy AWS gestita per supportare gli aggiornamenti gestiti quando è abilitata l'etichettatura delle risorse con i modelli di avvio.	17 marzo 2026
Argomento rivisto: eb logs comando con opzione `--analyze`	È stata aggiunta `--analyze` l'opzione al eb logs comando per richiedere l'analisi basata sull'intelligenza artificiale dei log di ambiente, degli eventi e dello stato dell'istanza.	13 marzo 2026
Nuovo argomento: Visualizzazione dei registri di distribuzione	Elastic Beanstalk aggiunge il supporto per visualizzare i log di distribuzione per l'ambiente direttamente dalla console.	11 marzo 2026
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkWorkerTier`	Autorizzazioni aggiornate nella politica AWS gestita per l'analisi dell'ambiente basata sull'intelligenza artificiale.	11 marzo 2026

AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkWebTier`	Autorizzazioni aggiornate nella politica AWS gestita per l'analisi dell'ambiente basata sull'intelligenza artificiale.	11 marzo 2026
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkMulticontainerDocker`	Autorizzazioni aggiornate nella politica AWS gestita per l'analisi dell'ambiente basata sull'intelligenza artificiale.	11 marzo 2026
Nuovo argomento: analisi dell'ambiente basata sull'intelligenza artificiale	Elastic Beanstalk aggiunge analisi basate sull'intelligenza artificiale per aiutare a identificare le cause principali e fornire soluzioni per problemi di salute ambientale utilizzando Amazon Bedrock.	26 febbraio 2026
Nuovo argomento: CI/CD integrazione con GitHub Actions	È stata aggiunta documentazione per l'implementazione delle CI/CD pipeline con Elastic Beanstalk utilizzando Actions. GitHub	23 febbraio 2026
AWS aggiornamenti delle policy gestiti per `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy`	Autorizzazioni aggiornate nella politica AWS gestita per supportare l'etichettatura delle risorse in ambienti a istanza singola.	27 gennaio 2026
Argomento rivisto: estrazione di chiavi JSON per i segreti Gestione dei segreti AWS	Elastic Beanstalk aggiunge il supporto per estrarre Gestione dei segreti AWS chiavi JSON specifiche dai segreti aggiungendo due punti e il nome della chiave all'ARN segreto.	13 gennaio 2026

AWS aggiornamenti delle policy gestiti per `AdministratorAccess-AWSElasticBeanstalk`	Autorizzazioni aggiornate nella politica AWS gestita.	12 novembre 2025
Argomento rivisto: Distribuzione di applicazioni Elastic Beanstalk in.NET utilizzando strumenti di distribuzione AWS	Argomento aggiornato per rimuovere il contenuto obsoleto di awsdeploy.exe e aggiungere la documentazione per gli attuali strumenti di distribuzione delle applicazioni.NET. AWS	3 ottobre 2025
Nuovo argomento: Configurazione dei sistemi di bilanciamento del carico Elastic Beanstalk dual-stack	Elastic Beanstalk aggiunge il supporto all'ambiente per IPv4 servire IPv6 sia i protocolli che i protocolli fornendo l'opzione di configurazione dual-stack per Application e Network Load Balancer.	9 settembre 2025
Nuovo argomento: CI/CD risorse per Elastic Beanstalk	È stato aggiunto un argomento con risorse per l'implementazione delle CI/CD pipeline con Elastic Beanstalk.	22 agosto 2025
L'argomento Piattaforme personalizzate è stato spostato nel capitolo Contenuti archiviati	L'argomento Piattaforme personalizzate ritirate è stato spostato nel capitolo Contenuti archiviati.	16 luglio 2025
Argomento rivisto: manifesto di distribuzione multisito di IIS	Documentazione aggiornata per l'esecuzione di più applicazioni e applicazioni ASP.NET Core con un manifesto di distribuzione.	11 luglio 2025

Nuovo argomento: QuickStart per Docker Compose	Nuovo argomento: QuickStart per la distribuzione di un'applicazione Docker Compose multi-container su Elastic Beanstalk.	8 luglio 2025
Argomento rivisto: eb migrate comando con opzione `--remote`	`--remote`Opzione aggiunta al eb migrate comando per la migrazione remota delle applicazioni IIS.	25 giugno 2025
Nuovo argomento: riferimento allo schema del manifesto di distribuzione di Windows	È stata aggiunta la documentazione di riferimento allo schema per il manifesto di distribuzione di Windows.	11 giugno 2025
Argomento rivisto: Che cos'è AWS Elastic Beanstalk?	Nuovo diagramma di architettura e introduzione sintetica per maggiore chiarezza.	5 giugno 2025
Argomento rivisto: Configurazione dell'EB CLI	Istruzioni di installazione condensate e mirate.	5 giugno 2025
Argomento rivisto: PHP Quickstart	È stato aggiunto un passaggio per aggiornare e ridistribuire l'applicazione e sono state migliorate alcune istruzioni.	5 giugno 2025
Argomento rivisto: Tutorial introduttivo	Il tutorial introduttivo è stato semplificato per offrire maggiore chiarezza e velocizzare l'onboarding.	5 giugno 2025
Contenuti riorganizzati	Struttura dei contenuti riorganizzata per evidenziare le informazioni essenziali e migliorare la reperibilità.	5 giugno 2025

Nuovo argomento: gestione dei gruppi di sicurezza EC2	Elastic Beanstalk aggiunge il supporto per l'esclusione dell'ambiente dal gruppo di sicurezza EC2 predefinito.	30 aprile 2025
Nuovo argomento: migrazione delle applicazioni IIS su Elastic Beanstalk	Elastic Beanstalk aggiunge funzionalità per la migrazione delle applicazioni Windows IIS. AWS Elastic Beanstalk	18 aprile 2025
Nuovo argomento: Utilizzo di Elastic Gestione dei segreti AWSAWS Systems Manager Beanstalk con e Parameter Store	Elastic Beanstalk aggiunge il supporto Gestione dei segreti AWS ai AWS Systems Manager segreti di riferimento e ai parametri Parameter Store con variabili di ambiente.	31 marzo 2025
AWS aggiornamenti delle policy gestiti per. `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`	Autorizzazioni aggiornate nella politica AWS gestita.	27 febbraio 2025
Le configurazioni di avvio sono diventate obsolete a favore dei modelli di avvio	Ho rivisto questo argomento per spiegare in che modo gli ambienti Elastic Beanstalk sono influenzati dalla deprecazione delle configurazioni di avvio da parte di Amazon EC2 Auto Scaling a favore dei modelli di avvio. Abbiamo inoltre rivisto altri contenuti relativi alle configurazioni di lancio e ai modelli di lancio.	23 gennaio 2025

Nuovo argomento: Utilizzo di EC2 Fast Launch con i rami della piattaforma Windows	Le versioni della piattaforma Elastic Beanstalk Windows includono AMI di base con EC2 Fast Launch abilitato.	22 gennaio 2025
Nuovo argomento: strategia di allocazione delle istanze Spot	Elastic Beanstalk aggiunge il supporto per la configurazione della strategia di allocazio ne Spot durante la creazione dell'ambiente.	15 gennaio 2025
`AdministratorAccess-AWSElasticBeanstalk` AWS policy gestita	Autorizzazioni aggiornate nella politica gestita. AWS	11 dicembre 2024
Raccomandazioni per gli ambienti Graviton arm64 first wave, capitolo sui contenuti archiviati	I consigli EB per gli ambienti Graviton arm64 di prima generazione sono stati spostati nel capitolo sui contenuti archiviati	5 ottobre 2024
Modelli di avvio	Nuovo argomento: Avvia i modelli. Questo argomento spiega come Amazon EC2 Auto Scaling sta eliminando gradualmente le configurazioni di avvio a favore dei modelli di lancio.	1 ottobre 2024
QuickStartfor Python	Nuovo argomento: QuickStart per Python	24 settembre 2024
Autorizzazioni richieste per i bucket Amazon S3 per policy restrittive sugli endpoint VPC	Nuovo argomento: Autorizza zioni richieste per i bucket Amazon S3 per policy restritti ve sugli endpoint VPC	18 settembre 2024

QuickStart per Java su Tomcat	Nuovo argomento: QuickStart per l'applicazione Java JSP in esecuzione su Tomcat	12 settembre 2024
QuickStart per Java	Nuovo argomento: QuickStart per Java	12 settembre 2024
Utilizzo dei repository privati di Amazon EC2 Systems Manager for Docker	Argomento ripristinato: Utilizzo dei repository privati di Amazon EC2 Systems Manager for Docker. Sostituis ce l'argomento Utilizzo dei repository privati Gestione dei segreti AWS for Docker fino a quando non è supportata l'integrazione Elastic Beanstalk /SSM.	8 settembre 2024
EB CLI 2.6 (ritirato) al capitolo Contenuti archiviati	Spostato EB CLI 2.6 (ritirato) nel capitolo Contenuti archiviat i	15 agosto 2024
API EB (ritirata) nel capitolo Contenuti archiviati	API EB spostata (ritirata) nel capitolo Contenuti archiviati	15 agosto 2024
Distribuzione di applicazioni Elastic Beanstalk da contenito ri Docker	Aggiornato e riorganizzato: distribuzione di applicazioni Elastic Beanstalk dai contenito ri Docker	15 agosto 2024
Contenuti archiviati	Nuovo capitolo: contenuti archiviati	15 agosto 2024
QuickStart per Windows ASP.NET	Nuovo argomento: QuickStart per Windows ASP.NET	5 luglio 2024
QuickStart per.NET Core su Windows	Nuovo argomento: QuickStart per.NET Core su Windows	28 giugno 2024

QuickStart per Docker	Nuovo argomento: QuickStart per Docker	19 giugno 2024
Impedire l'accesso ai bucket Amazon S3 tra ambienti	Nuovo argomento: Prevenzione dell'accesso ai bucket Amazon S3 tra ambienti	12 giugno 2024
QuickStart per.NET Core su Linux	Nuovo argomento: QuickStart per.NET Core su Windows	28 maggio 2024
QuickStart per PHP	Nuovo argomento: QuickStart per PHP	10 maggio 2024
QuickStart per Node.js	Nuovo argomento: QuickStart per Node.js	5 maggio 2024
QuickStart per Go	Nuovo argomento: QuickStart per Go	5 maggio 2024
Pianificazione del rilascio della piattaforma Elastic Beanstalk	È stato aggiunto un nuovo argomento che include un programma di. Prossime versioni di Platform Branch Spostato Ritiro della pianificazione dei rami della piattaforma e Cronologia dei rami di piattaforma ritirati in questo argomento.	1º maggio 2024
`AWSElasticBeanstalkRoleCore` AWS policy gestita	Autorizzazioni aggiornate nella politica AWS gestita.	30 aprile 2024
`AWSElasticBeanstalkManagedUpdatesServiceRolePolicy` AWS policy gestita	Autorizzazioni aggiornate nella politica AWS gestita.	30 aprile 2024

AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy AWS policy gestita	Autorizzazioni aggiornate nella politica AWS gestita.	30 aprile 2024
AWSElasticBeanstalkMaintenance AWS policy gestita	Autorizzazioni aggiornate nella politica AWS gestita.	30 aprile 2024
AWSElasticBeanstalkInternalMaintenanceRolePolicy AWS policy gestita	Autorizzazioni aggiornate nella politica AWS gestita.	30 aprile 2024

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.