aws

Guia do Desenvolvedor

AWS Elastic Beanstalk



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Elastic Beanstalk: Guia do Desenvolvedor

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é AWS Elastic Beanstalk? 1
Plataformas compatíveis 2
Fluxo de trabalho de implantação de 3
Preços 3
Próximas etapas 4
Tutorial de inicialização 5
O que você vai construir 7
Etapa 1 - Crie um aplicativo 7
Etapa 2 - Implantar seu aplicativo 10
Etapa 3 - Explore o ambiente 13
Solução de problemas com logs 16
Etapa 4 - Atualize seu aplicativo 18
Etapa 5 - Dimensione seu aplicativo 19
Aumentar as configurações de capacidade 20
Verifique o aumento da capacidade 21
Limpeza 21
Próximas etapas 23
Configurando o EB CLI 24
Instale o EB CLI 25
Instalação manual 26
Instale no virtualenv 28
Instalar a EB CLI com o homebrew 29
Configurar a EB CLI 30
Ignorar arquivos que usam .ebignore 32
Usar perfis nomeados 33
Implantar um artefato, em vez da pasta do projeto 34
Definições de configuração e precedência 34
Metadados da instância 35
Usar a EB CLI com o Git 35
Associar ambientes do Elastic Beanstalk a ramificações Git 36
Implantação de alterações 36
Uso dos submódulos do Git 36
Para atribuir tags do Git à versão do aplicativo 37
Comandos da EB CLI 39

Opções comuns 40
eb abort 41
Descrição 41
Sintaxe 41
Opções 41
Saída 41
Exemplo 41
eb appversion 42
Descrição 42
Sintaxe 42
Opções 42
Usar o comando interativamente 45
Saída 45
Exemplos 45
eb clone 46
Descrição 46
Sintaxe 47
Opções 47
Saída 49
Exemplo 49
eb codesource 49
Descrição 49
Sintaxe 50
Opções 50
Saída 50
Exemplos 50
eb config 51
Descrição 51
Sintaxe 52
Opções 53
Saída 56
Exemplos 57
eb console 61
Descrição 61
Sintaxe 61
Opções 61

eb create 61
Descrição 61
Sintaxe 62
Opções 63
Saída 75
Exemplos 76
eb deploy 78
Descrição 78
Sintaxe 78
Opções 79
Saída 80
Exemplo 80
eb events 80
Descrição 80
Sintaxe 80
Opções 81
Saída 81
Exemplo 81
eb health 82
Descrição 82
Sintaxe 82
Opções 82
Saída 83
Exemplo 83
eb init 84
Descrição 84
Sintaxe 84
Opções 85
CodeBuild apoio 87
Saída 87
Exemplo 87
eb labs 88
Descrição 88
eb list 89
Descrição 89
Sintaxe 89

Opções ... 89
Saída ... 89
Exemplo 1 ... 89
Exemplo 2 ... 90
eb local ... 90
Descrição ... 90
Sintaxe ... 90
Opções ... 91
Saída ... 92
Exemplos ... 92
eb logs ... 94
Descrição ... 94
Sintaxe ... 94
Opções ... 95
Saída ... 98
Exemplos ... 98
eb migrate ... 98
Descrição ... 98
Sintaxe ... 100
Subcomandos ... 100
Opções ... 102
Saída ... 110
Exemplos ... 110
eb open ... 114
Descrição ... 114
Sintaxe ... 115
Opções ... 115
Saída ... 115
eb platform ... 115
Descrição ... 115
Usar eb platform para plataformas personalizadas ... 116
Usar eb platform para ambientes ... 123
eb printenv ... 125
Descrição ... 125
Sintaxe ... 126
Opções ... 126

Saída ... 126
Exemplo ... 126
eb restore ... 126
Descrição ... 126
Sintaxe ... 126
Opções ... 127
Saída ... 127
Exemplo ... 127
eb scale ... 128
Descrição ... 128
Sintaxe ... 128
Opções ... 128
Saída ... 128
Exemplo ... 128
eb setenv ... 129
Descrição ... 129
Sintaxe ... 129
Opções ... 129
Saída ... 129
Exemplo ... 129
eb ssh ... 130
Descrição ... 130
Sintaxe ... 131
Opções ... 131
Saída ... 132
Exemplo ... 132
eb status ... 133
Descrição ... 133
Sintaxe ... 133
Opções ... 133
Saída ... 134
Exemplo ... 134
eb swap ... 135
Descrição ... 135
Sintaxe ... 135
Opções ... 135

Saída ... 136
Exemplos ... 136
eb tags ... 136
Descrição ... 136
Sintaxe ... 137
Opções ... 137
Saída ... 139
Exemplos ... 139
eb terminate ... 140
Descrição ... 140
Sintaxe ... 141
Opções ... 141
Saída ... 142
Exemplo ... 142
eb upgrade ... 143
Descrição ... 143
Sintaxe ... 143
Opções ... 143
Saída ... 143
Exemplo ... 144
eb use ... 144
Descrição ... 144
Sintaxe ... 144
Opções ... 144
Conceitos ... 146
Aplicação ... 147
Versão da aplicação ... 147
Environment ... 147
Nível do ambiente ... 147
Configuração de ambientes ... 148
Configuração salva ... 148
Plataforma ... 148
Ambientes de servidor da web ... 148
Ambientes de operador ... 150
Considerações sobre design ... 151
Escalabilidade ... 152

Segurança ... 153
Armazenamento persistente ... 153
Tolerância a falhas ... 155
Entrega de conteúdo ... 155
Atualizações e correções de software ... 155
Conectividade ... 156
Como gerenciar aplicações do ... 157
Console de gerenciamento de aplicativos ... 158
Gerenciar versões de aplicativos ... 159
Criação de versões de aplicações ... 159
Exclusão de versões de aplicações ... 160
Ciclo de vida da versão ... 162
Atribuir tags em versões do aplicativo ... 163
Criar um pacote de origem ... 166
Criar pacote de origem a partir da linha de comando ... 167
Criar pacote de origem com o Git ... 167
Compactar arquivos no Mac OS X Finder ou Windows Explorer ... 168
Criar pacote de origem para um aplicativo .NET ... 169
Testar pacote de origem ... 170
Edifício com CodeBuild ... 170
Como criar uma aplicação do ... 171
Criar e implantar o código do aplicativo ... 171
Marcar aplicativos ... 173
Adicionar tags durante a criação do aplicativo ... 173
Gerenciar tags de um aplicativo existente ... 174
Marcar recursos ... 175
Recursos que podem ser marcados ... 176
Propagação de tags para modelos de execução ... 177
Criar ambientes ... 179
Console de gerenciamento de ambiente ... 180
Acessar o console ... 181
Painel de visão geral do ambiente ... 182
Detalhe do ambiente ... 183
Ações do ambiente ... 187
Criar ambientes ... 189
O assistente de criação de novo ambiente ... 195

Clonar um ambiente ... 219
Encerrar um ambiente ... 221
Com o AWS CLI ... 223
Com API ... 225
URL "Iniciar agora" ... 229
Compose Environments ... 233
Composição de ambientes ... 236
Implantações ... 238
Como escolher uma política de implantação ... 240
Implantar uma nova versão do aplicativo ... 242
Reimplantar a versão anterior ... 242
Outras formas de implantar seu aplicativo ... 243
Opções de implantação ... 243
Implantações azuis/verdes ... 252
Alterações de configuração ... 254
Atualizações contínuas ... 256
Atualizações imutáveis ... 261
Atualizações da plataforma ... 265
Método 1: Atualizar a versão da plataforma do ambiente ... 268
Método 2 — Executar uma Blue/Green implantação ... 270
Atualizações gerenciadas ... 272
Atualizar um ambiente legado ... 279
Migrar para 023/ AL2 AL2 ... 281
Perguntas frequentes sobre descontinuação de plataformas ... 298
Cancelar uma atualização ... 303
Reconstruir um ambiente ... 304
Reconstrução de um ambiente em execução ... 304
Reconstruir um ambiente encerrado ... 305
Tipos de ambiente ... 307
Ambiente escalável e com balanceamento de carga ... 307
Ambiente de instância única ... 308
Alterar tipo de ambiente ... 308
Ambientes de operador ... 309
Daemon SQS do ambiente de operador ... 312
Filas de mensagens não entregues ... 313
Tarefas periódicas ... 314

Use a Amazon CloudWatch para escalabilidade automática em níveis de ambiente de trabalho ... 316
Configurar ambientes de operador ... 316
Links de ambiente ... 320
Recuperação de um estado de pilha inválido ... 322
Resolvendo o erro ... 322
Por que o erro ocorre ... 323
Configurar ambientes ... 324
Recursos provisionados ... 324
Configurar usando o console ... 326
Página de configuração ... 326
Página Revisar alterações ... 328
EC2 Instâncias da Amazon ... 329
Tipos de EC2 instância da Amazon ... 331
Configuração com o console ... 332
Gerenciando grupos EC2 de segurança ... 340
Configurar com o AWS CLI ... 342
Configuração com um namespace ... 353
IMDS ... 354
Auto Scaling group (Grupo do Auto Scaling) ... 357
Modelos de inicialização ... 359
Suporte para instâncias spot ... 362
Acionadores ... 379
Ações programadas ... 381
Configuração de verificação de integridade ... 385
Load balancer ... 386
Classic Load Balancer ... 388
Application Load Balancer ... 400
Application Load Balancer compartilhado ... 420
Network Load Balancer ... 439
Configurar logs de acesso ... 451
Banco de dados ... 451
Ciclo de vida do banco de dados ... 452
Adição de uma instância de banco de dados do Amazon RDS ao seu ambiente usando o console ... 453
Conectar ao banco de dados ... 455

Configuração de uma instância de banco de dados do RDS integrada, usando o console ... 456
Configuração de uma instância de banco de dados do RDS integrada usando arquivos de configuração ... 457
Desacoplamento de uma instância de banco de dados do RDS usando o console ... 457
Desacoplamento de uma instância de banco de dados do RDS usando arquivos de configuração ... 461
Segurança ... 463
Configurar a segurança do ambiente ... 463
Namespaces de configuração de segurança do ambiente ... 466
Marcação em ambientes ... 466
Adicionar tags durante a criação do ambiente ... 467
Gerenciar tags de um ambiente existente ... 468
Variáveis de ambiente e configurações de software ... 470
Definir configurações específicas da plataforma ... 471
Configurar propriedades do ambiente (variáveis de ambiente) ... 472
Namespaces de configurações de software ... 474
Acessar propriedades de ambiente ... 476
Depuração ... 477
Visualização de logs ... 480
Notificações ... 483
Configurar notificações usando o console do Elastic Beanstalk ... 484
Configurar notificações usando opções de configuração ... 485
Configurar permissões para enviar notificações ... 487
Amazon VPC ... 489
Definir as configurações da VPC no console do Elastic Beanstalk ... 489
O namespace aws:ec2:vpc ... 492
Migração do EC2 -Classic para uma VPC ... 493
Nome de domínio ... 498
Configurar ambientes (avançado) ... 500
Opções de configuração ... 501
Precedência ... 501
Valores recomendados ... 502
Antes da criação do ambiente ... 504
Durante a criação ... 510
Depois da criação ... 517
Opções gerais ... 527

Opções específicas para a plataforma 622
Opções personalizadas 635
.Ebextensions 636
Configurações de opção 639
Servidor do Linux 641
Servidor do Windows 659
Recursos personalizados 669
Configurações salvas 698
Marcação de configurações salvas 703
`env.yaml` 705
Imagem personalizada 708
Criar uma AMI do personalizada 708
Gerenciar uma AMI personalizada 713
Limpeza de uma AMI personalizada 713
AMI baseada em plataforma descontinuada 714
Arquivos estáticos 720
Configurar arquivos estáticos usando o console 721
Configurar arquivos estáticos usando opções de configuração 722
HTTPS 723
Certificados de servidor 725
Encerrar o HTTPS no balanceador de carga 732
Encerrar HTTPS na instância 735
End-to-end criptografia 772
Passagem de TCP 776
Redirecionamento de HTTP para HTTPS 777
Plataformas 780
Glossário de plataformas 780
Modelo de responsabilidade compartilhada 784
Política de suporte de plataforma 785
Ramificações desativadas da plataforma 785
Após o período de carência de 90 dias 786
Programação de plataformas 787
Recursos de planejamento 787
Próximos lançamentos de ramificações de plataforma 788
Programação de ramificações desativadas da plataforma 788
Histórico de ramificações de plataforma retiradas 789

Histórico de servidores e sistemas operacionais ... 794
Plataformas compatíveis ... 796
Plataformas compatíveis e histórico de componentes ... 797
Plataformas Linux ... 797
Versões compatíveis do Amazon Linux ... 798
Lista de plataformas Linux do Elastic Beanstalk ... 799
Fluxo de trabalho de implantação da instância ... 800
Fluxo de trabalho de implantação de instâncias para ECS em AL2 e versões posteriores ... 803
Ferramentas de script de plataforma ... 805
Estender plataformas Linux ... 815
Buildfile e Procfile ... 815
Hooks de plataforma ... 817
Arquivos de configuração ... 820
Configuração de proxy reverso ... 820
Exemplo de aplicativo com extensões ... 824
Implantando o.NET (Windows) ... 826
QuickStart para o.NET Core no Windows ... 827
Sua AWS conta ... 828
Pré-requisitos ... 830
Etapa 1: Criar um aplicação .NET Core no Windows ... 830
Etapa 2: executar a aplicação localmente ... 831
Etapa 3: implantar sua aplicação .NET Core no Windows com a EB CLI ... 832
Etapa 4: executar a aplicação no Elastic Beanstalk ... 833
Etapa 5: limpar ... 833
AWS recursos para seu aplicativo ... 833
Próximas etapas ... 834
Implantar com o console ... 835
QuickStart para ASP.NET ... 835
Sua AWS conta ... 836
Pré-requisitos ... 838
Etapa 1: criar uma aplicação ASP.NET ... 838
Etapa 2: executar a aplicação localmente ... 839
Etapa 3: Implantar seu aplicativo ASP.NET com o AWS Toolkit for Visual Studio ... 839
Etapa 4: executar a aplicação no Elastic Beanstalk ... 840
Etapa 5: limpar ... 840
AWS recursos para seu aplicativo ... 841

Próximas etapas 841
Implantar com o console 842
Ambiente de desenvolvimento 842
Instalar um IDE 843
Instalando o AWS Toolkit for Visual Studio 843
A Plataforma .NET 843
Configurar o ambiente .NET no console do Elastic Beanstalk 844
O namespace aws:elasticbeanstalk:container:dotnet:apppool 845
Migração de versão principal 846
Manifesto de implantação 849
EC2 Lançamento rápido 868
Adicionar um banco de dados 869
Adicionar uma instância de banco de dados ao seu ambiente 870
Download de um driver 871
Conectar-se a um banco de dados 872
O AWS Toolkit for Visual Studio 873
Testar localmente 874
Criar um ambiente do Elastic Beanstalk 874
Encerrar um ambiente 883
Implante 884
Gerenciar ambientes 887
Gerenciar contas 901
Depure 902
Monitor 903
Ferramenta de implantação 905
Migrar o aplicativo local 908
Recomendações de componentes retirados 908
Ramificações retiradas do Windows 2012 R2 908
Compatibilidade com TLS 1.2 909
Implantação do.NET core (Linux) 911
QuickStart para o.NET Core no Linux 911
Sua AWS conta 912
Pré-requisitos 914
Etapa 1: criar uma aplicação .NET Core no Linux 915
Etapa 2: executar a aplicação localmente 915
Etapa 3: implantar sua aplicação .NET Core no Linux com a EB CLI 916

Etapa 4: executar a aplicação no Elastic Beanstalk ... 917
Etapa 5: limpar ... 833
AWS recursos para seu aplicativo ... 918
Próximas etapas ... 919
Implantar com o console ... 919
Ambiente de desenvolvimento ... 919
Instalar o .NET Core SDK ... 920
Instalar um IDE ... 843
Instalando o AWS Toolkit for Visual Studio ... 843
O .NET Core na plataforma Linux ... 920
Considerações sobre o .NET Core na plataforma Linux ... 920
Configurar o ambiente .NET Core no Linux ... 922
Namespace de configuração do .NET Core no Linux ... 923
Empacotar aplicativos ... 923
Procfile ... 925
Servidor de proxy ... 926
O AWS Toolkit for Visual Studio ... 927
Pré-requisitos ... 927
Criar um projeto de aplicação ... 928
Criar um ambiente do Elastic Beanstalk e implantar a aplicação ... 929
Encerrar um ambiente ... 931
Gerenciar ambientes ... 932
Monitor ... 945
Migração do Windows para o Linux ... 947
Considerações sobre a migração para a plataforma .NET Core no Linux ... 947
Implantando o Go ... 948
QuickStart para Go ... 948
Sua AWS conta ... 949
Pré-requisitos ... 951
Etapa 1: criar uma aplicação Go ... 951
Etapa 2: implantar sua aplicação Go com a EB CLI ... 952
Etapa 3: executar a aplicação no Elastic Beanstalk ... 953
Etapa 4: limpar ... 833
AWS recursos para seu aplicativo ... 954
Próximas etapas ... 955
Implantar com o console ... 955

Ambiente de desenvolvimento ... 955
Instalação do Go ... 955
Instalação do AWS SDK for Go ... 955
A Plataforma Go ... 956
Configurar o ambiente Go ... 957
Namespace de configuração do Go ... 958
Procfile ... 960
Buildfile ... 962
Configuração do proxy ... 963
Implantando o Java ... 965
QuickStart para Java ... 966
Sua AWS conta ... 967
Pré-requisitos ... 969
Etapa 1: criar uma aplicação Java ... 970
Etapa 2: executar a aplicação localmente ... 972
Etapa 3: implantar sua aplicação Java com a EB CLI ... 973
Etapa 4: executar a aplicação no Elastic Beanstalk ... 974
Etapa 5: limpar ... 974
AWS recursos para seu aplicativo ... 975
Próximas etapas ... 976
Implantar com o console ... 976
QuickStart para Java no Tomcat ... 976
Sua AWS conta ... 977
Pré-requisitos ... 979
Etapa 1: criar uma aplicação Java JSP ... 979
Etapa 2: implantar a aplicação Java JSP com a EB CLI ... 980
Etapa 3: executar a aplicação no Elastic Beanstalk ... 981
Etapa 4: limpar ... 833
AWS recursos para seu aplicativo ... 981
Próximas etapas ... 982
Implantar com o console ... 983
Ambiente de desenvolvimento ... 983
Instalar o Java Development Kit ... 983
Instalar um contêiner da web ... 983
Download de bibliotecas ... 984
Instalando o AWS SDK for Java ... 984

Instalar um editor de texto ou IDE 984
Aplicações de exemplo e tutoriais 985
Iniciar um ambiente com um aplicativo Java de exemplo 985
Próximas etapas 995
A Plataforma Tomcat 995
Configurar o grupo do seu ambiente Tomcat 997
Namespaces de configuração do Tomcat 1001
Empacotamento de arquivos WAR 1003
Estruturar a pasta do projeto 1004
Configuração do proxy 1008
Plataforma Java SE 1013
Configurar o ambiente Java SE 1015
Namespace de configuração do Java SE 1016
Buildfile 1018
Procfile 1019
Configuração do proxy 1020
Adicionar um banco de dados 1024
Fazer download do driver do JDBC 1026
Conectar a um banco de dados (plataformas Java SE) 1027
Conectar a um banco de dados (plataformas Tomcat) 1027
Solução de problemas de conexões do banco de dados 1030
Recursos 1033
Implantando o Node.js 1034
QuickStart para Node.js 1034
Sua AWS conta 1035
Pré-requisitos 1037
Etapa 1: criar uma aplicação Node.js 1038
Etapa 2: executar a aplicação localmente 1039
Etapa 3: implantar a aplicação Node.js com a EB CLI 1039
Etapa 4: executar a aplicação no Elastic Beanstalk 1040
Etapa 5: limpar 833
AWS recursos para seu aplicativo 1040
Próximas etapas 1041
Implantar com o console 1042
Ambiente de desenvolvimento 1042
Instale o Node.js. 1042

Confirmar instalação do NPM 1042
Instale o AWS SDK para Node.js 1043
Instale o gerador Express 1043
Configurar uma estrutura e um servidor Express 1044
A Plataforma Node.js 1045
Configurar o ambiente Node.js 1046
Namespace de configuração do Node.js 1049
Procfile 1053
Configurar dependências 1053
Arquivo npm Shrinkwrap 1058
Configuração do proxy 1058
Exemplos de aplicações e tutoriais 1062
Iniciar um ambiente com um aplicativo Node.js de exemplo 1062
Próximas etapas 1064
Tutorial – Express 1064
Pré-requisitos 1064
Criar um ambiente do Elastic Beanstalk 1065
Atualize o aplicativo para usar o Express 1067
Atualize o aplicativo para usar o Amazon RDS 1071
Limpeza 1076
Tutorial: Express com clustering 1076
Pré-requisitos 1077
Criar um ambiente do Elastic Beanstalk 1077
Atualize o aplicativo para usar o Express 1080
Limpeza 1094
Tutorial: Node.js com DynamoDB 1094
Pré-requisitos 1095
Criar um ambiente do Elastic Beanstalk 1095
Adicionar permissões às instâncias do seu ambiente 1098
Implante o aplicativo de exemplo 1099
Criar uma tabela do DynamoDB 1101
Atualizar os arquivos de configuração do aplicativo 1102
Configurar seu ambiente para alta disponibilidade 1104
Limpeza 1104
Próximas etapas 1105
Adicionar um banco de dados 1105

Adicionar uma instância de banco de dados ao seu ambiente 1106
Download de um driver 1107
Conectar-se a um banco de dados 1108
Recursos 1109
Implantando o PHP 1110
QuickStart para PHP 1110
Sua AWS conta 1111
Pré-requisitos 1113
Etapa 1: criar uma aplicação PHP 1113
Etapa 2: executar a aplicação localmente 1113
Etapa 3: inicializar e implantar seu aplicativo PHP 1114
Etapa 4: Navegue pelo seu aplicativo em nuvem 1114
Etapa 5: atualizar e reimplantar seu aplicativo 1114
Limpeza 833
Próximas etapas 1116
Plataforma PHP 1116
Instalando o AWS SDK for PHP 1118
Considerações sobre o PHP 8.1 no Amazon Linux 2 1118
Configurar seu ambiente PHP 1120
Namespaces para configuração 1122
Instalar as dependências 1124
Atualização do Composer 1125
Como estender o php.ini 1126
Exemplos avançados 1127
Adicionar um banco de dados 1127
Tutorial - Laravel 1131
Tutorial – CakePHP 1140
Tutorial - Symfony 1148
Tutorial – produção de HA 1155
Tutorial - HA WordPress 1165
Tutorial – Drupal de alta disponibilidade 1182
Implantando o Python 1201
QuickStart para Python 1201
Sua AWS conta 1202
Pré-requisitos 1204
Etapa 1: criar uma aplicação Python 1206

Etapa 2: executar a aplicação localmente ... 1206
Etapa 3: implantar sua aplicação Python com a EB CLI ... 1207
Etapa 4: executar a aplicação no Elastic Beanstalk ... 1208
Etapa 5: limpar ... 833
AWS recursos para seu aplicativo ... 1208
Próximas etapas ... 1209
Implantar com o console ... 1210
Ambiente de desenvolvimento ... 1210
Pré-requisitos ... 1210
Usar um ambiente virtual ... 1211
Configurar um projeto Python para o Elastic Beanstalk ... 1212
A Plataforma Python ... 1213
Configurar o ambiente Python ... 1214
Namespaces de configuração do Python ... 1216
O `python3` executável ... 1218
Procfile ... 1219
Especificação de dependências ... 1220
Tutorial – flask ... 1222
Pré-requisitos ... 1223
Configurar um ambiente Python virtual com Flask ... 1223
Criar uma aplicação Flask ... 1224
Implantar o site com a CLI do EB ... 1227
Limpeza ... 1230
Próximas etapas ... 1231
Tutorial - Django ... 1231
Pré-requisitos ... 1231
Configurar um ambiente virtual Python e instalar o Django ... 1232
Criar um projeto Django ... 1234
Configurar a aplicação Django para o Elastic Beanstalk ... 1235
Implantar o site com a CLI do EB ... 1237
Atualizar seu aplicativo ... 1240
Limpeza ... 1244
Próximas etapas ... 1245
Adicionar um banco de dados ... 1245
Adicionar uma instância de banco de dados ao seu ambiente ... 1245
Download de um driver ... 1247

Conectar-se a um banco de dados ... 1247
Recursos ... 1248
Implantando o Ruby ... 1249
Ambiente de desenvolvimento ... 1249
Instalar o Ruby ... 1250
Instalando o AWS SDK para Ruby ... 1251
Instalar um editor de texto ou IDE ... 1251
A Plataforma Ruby ... 1252
Configurar o ambiente Ruby ... 1253
Namespaces de configuração do Ruby ... 1256
Gemfile ... 1257
Procfile ... 1258
Tutorial – rails ... 1259
Pré-requisitos ... 1260
Conhecimento básico do Elastic Beanstalk ... 1260
Iniciar um ambiente do Elastic Beanstalk ... 1260
Instalar o rails e gerar um site ... 1262
Definir as configurações do rails ... 1266
Implantar o aplicativo ... 1267
Limpeza ... 1268
Próximas etapas ... 1268
Tutorial – sinatra ... 1268
Pré-requisitos ... 1268
Iniciar um ambiente do Elastic Beanstalk ... 1269
Escrever um site sinatra básico ... 1271
Implantar o aplicativo ... 1272
Limpeza ... 1273
Próximas etapas ... 1273
Adicionar um banco de dados ... 1274
Adicionar uma instância de banco de dados ao seu ambiente ... 1274
Download de adaptador ... 1275
Conectar-se a um banco de dados ... 1276
Implantação com o Docker ... 1277
Ramificações da plataforma do Docker ... 1277
Filiais de plataformas descontinuadas em execução no Amazon Linux AMI (AL1) ... 1278
Ramificação da plataforma do Docker ... 1279

QuickStart para Docker ... 1279
Configuração da imagem do Docker ... 1287
Ramificação da plataforma gerenciada pelo ECS ... 1294
Visão geral da plataforma Docker gerenciada pelo ECS ... 1294
Recursos do Amazon ECS criados pelo Elastic Beanstalk ... 1295
Arquivo `Dockerrun.aws.json` v2 ... 1296
Imagens de Docker ... 1296
Falha nas implantações de contêiner ... 1297
Estendendo as plataformas Docker baseadas em ECS ... 1297
Configuração do Docker gerenciado pelo ECS para o Elastic Beanstalk ... 1298
Tutorial - Docker gerenciado pelo ECS ... 1310
Migração para o ECS em execução em 023 AL2 ... 1320
Usar imagens de um repositório privado ... 1322
Repositório do Amazon ECR ... 1322
SSM Parameter Store ... 1324
Arquivo Dockerrun.aws.json ... 1326
Configuração de ambientes ... 1329
Configurar o software em ambientes do Docker ... 1329
Fazer referência a variáveis de ambiente em contêineres ... 1331
Usar o recurso de interpolação para variáveis de ambiente com o Docker Compose ... 1333
Geração de logs para relatórios de integridade aprimorados com o Docker Compose ... 1334
Registro em log personalizado de contêineres do Docker com o Docker Compose ... 1335
Imagens de Docker ... 1336
Configurar atualizações gerenciadas para ambientes do Docker ... 1337
Namespaces de configuração do Docker ... 1338
Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2) ... 1338
Plataformas herdadas ... 1340
(Herdado) Migrar para o Docker em execução no Amazon Linux 2 ... 1341
Contêineres Docker GlassFish (antigos) ... 1346
Ambientes de monitoramento ... 1356
Console de monitoramento ... 1356
Grafos de monitoramento ... 1357
Personalizar o console de monitoramento ... 1358
Monitorando a saúde com CLI ... 1359
Leitura da saída ... 1363
Visualização de integridade interativa ... 1365

Opções de visualização de integridade interativa 1367
Relatórios de integridade básica 1367
Cores de integridade 1368
Verificações de integridade do Elastic Load Balancing 1369
Verificações de integridade do ambiente de nível do operador e instância única 1370
Verificações adicionais 1370
CloudWatch Métricas da Amazon 1371
Monitoramento e relatórios de integridade aprimorada 1372
Agente de integridade do Elastic Beanstalk 1374
Fatores que determinam a integridade de instâncias e ambientes 1375
Personalização de regra de verificação de integridade 1378
Funções de integridade aperfeiçoada 1378
Autorização de integridade aprimorada 1379
Eventos de integridade aprimorada 1380
Comportamento dos relatórios de integridade aprimorada durante as atualizações, as implantações e a escalabilidade 1381
Habilitar integridade aprimorada 1382
Console de integridade 1385
Status e cores de integridade 1390
Métricas de instância 1393
Regras de integridade aperfeiçoada 1396
CloudWatch 1401
Usuários da API 1410
Formato de log de integridade aprimorado 1412
Notificações e solução de problemas 1416
Gerenciar alarmes 1418
Ver histórico de alterações 1420
Visualizar eventos 1422
Console 1422
Linha de comando 1422
Monitorar instâncias 1423
Visualizar logs da instância 1424
Localização do log nas EC2 instâncias da Amazon 1426
Local de log no Amazon S3 1428
Configurações de rotação de log no Linux 1428
Extensão da configuração padrão da tarefa de log 1429

Transmissão de arquivos de log para o Amazon CloudWatch Logs 1432
Integrando serviços AWS 1434
Visão geral da arquitetura 1434
CloudFront 1435
CloudTrail 1436
Informações sobre o Elastic Beanstalk em CloudTrail 1436
Noções básicas sobre entradas de arquivo de log do Elastic Beanstalk 1437
CloudWatch 1438
CloudWatch Registros 1439
Pré-requisitos para instanciar o streaming de registros para o Logs CloudWatch 1440
Como o Elastic Beanstalk configura os registros CloudWatch 1441
Transmitir registros de instâncias para CloudWatch Logs 1446
Solução de problemas da integração de CloudWatch registros 1448
Streaming de integridade do ambiente 1450
EventBridge 1453
Monitore um recurso do Elastic Beanstalk com EventBridge 1454
Eventos demonstrativos de padrões de eventos do Elastic Beanstalk 1456
Eventos demonstrativos do Elastic Beanstalk 1459
Mapeamento de campo de evento do Elastic Beanstalk 1460
AWS Config 1464
Conf AWS Config iguração 1464
Configuração AWS Config para registrar recursos do Elastic Beanstalk 1465
Visualizando detalhes de configuração do Elastic Beanstalk no console AWS Config 1466
Avaliação dos recursos do Elastic Beanstalk usando regras AWS Config 1469
DynamoDB 1470
ElastiCache 1470
Amazon EFS 1471
Arquivos de configuração 1472
Sistemas de arquivos criptografados 1473
Aplicações de exemplo 1473
Limpeza dos sistemas de arquivos 1474
IAM 1474
Perfis de instância 1475
Perfis de serviço 1480
Uso de perfis vinculados ao serviço 1488
Políticas de usuário 1500

Formato ARN ... 1508
Recursos e condições ... 1510
Controle de acesso com base em tags ... 1555
Exemplo de políticas gerenciadas ... 1559
Exemplo de políticas específicas de recursos ... 1563
Acesso ao bucket do S3 entre ambientes ... 1573
Amazon RDS ... 1575
Amazon RDS na VPC padrão ... 1577
Credenciais do Amazon RDS e Secrets Manager ... 1583
Limpar uma instância externa do Amazon RDS ... 1584
Amazon S3 ... 1584
O bucket de contas de clientes do Amazon S3 do Elastic Beanstalk ... 1584
Conteúdo do bucket de contas de clientes Amazon S3 do Elastic Beanstalk ... 1584
Excluir objetos no bucket do Amazon S3 do Elastic Beanstalk ... 1586
Excluir o bucket do Amazon S3 do Elastic Beanstalk ... 1586
Armazenamento de parâmetros do Secrets Manager e do Systems Manager ... 1588
Busque segredos para variáveis de ambiente ... 1588
Permissões obrigatórias do IAM ... 1596
Usando o Secrets Manager e o Systems Manager Parameter Store ... 1598
Solução de problemas de segredos e variáveis de ambiente ... 1600
Amazon VPC ... 1601
VPC pública ... 1603
VPC pública/privada ... 1604
VPC privada ... 1604
Bastion hosts ... 1607
Amazon RDS ... 1612
Endpoints da VPC ... 1620
Políticas de VPC endpoint ... 1624
Segurança ... 1634
Proteção de dados ... 1635
Criptografia de dados ... 1636
Privacidade entre redes ... 1637
Gerenciamento de identidade e acesso ... 1637
AWS políticas gerenciadas ... 1638
Registro em log e monitoramento ... 1652
Relatórios de integridade aprimorada ... 1652

Registros de EC2 instâncias da Amazon 1652
Notificações do ambiente 1653
CloudWatch Alarmes da Amazon 1653
AWS CloudTrail troncos 1653
AWS X-Ray depuração 1653
Validação de conformidade 1653
Resiliência 1654
Segurança da infraestrutura 1655
Modelo de responsabilidade compartilhada 1655
Práticas recomendadas de segurança 1656
Práticas recomendadas de segurança preventiva 1656
Práticas recomendadas de segurança de detecção 1657
Permissões 1659
Perfis necessários 1659
Políticas e perfis opcionais 1659
Perfil de serviço 1660
`AWSElasticBeanstalkEnhancedHealth` 1481
`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` 1481
Perfil de instância 1662
Políticas gerenciadas 1663
Criação de um perfil de EC2 instância 1663
Política de usuário 1663
Tutoriais e amostras 1665
Migração de aplicativos IIS 1668
Pré-requisitos 1669
Glossário de migração 1671
Termos do Windows, IIS e.NET 1671
Termos do Elastic Beanstalk 1676
Termos em Python 1680
Migrações básicas do IIS 1681
Explorando seu ambiente IIS 1681
Preparar-se para migração 1683
Sua primeira migração 1683
Controle da migração 1684
Monitorar o progresso 1685
Verificando a migração 1685

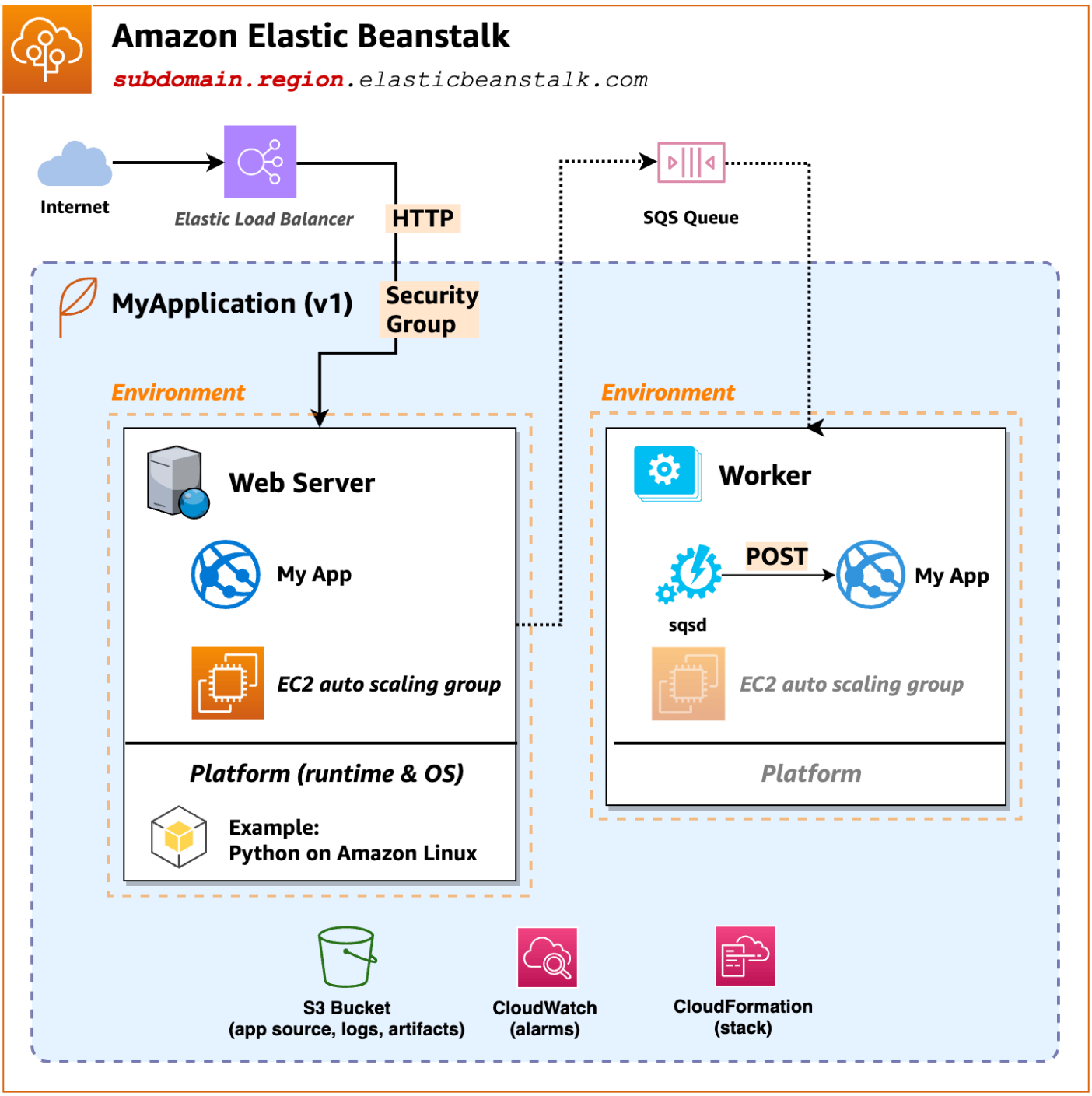
Gerenciando artefatos de migração ... 1686
Configuração de rede ... 1687
Configuração de VPC ... 1687
Implantações em vários locais com configurações de portas ... 1692
Configuração e dependências compartilhadas ... 1694
Práticas recomendadas ... 1694
Solução de problemas ... 1695
Configuração de segurança ... 1696
Configuração do perfil da instância ... 1696
Gerenciamento de funções de serviço ... 1696
Configuração do security group ... 1697
Integração de certificados SSL ... 1698
Autenticação do Windows ... 1699
Melhores práticas e solução de problemas ... 1700
Mapeamento de migração do IIS para o Elastic Beanstalk ... 1702
Sites e aplicativos do IIS no Elastic Beanstalk ... 1702
Gerenciamento virtual de diretórios e caminhos de aplicativos ... 1704
Reescrita de URL e roteamento de solicitação de aplicativo (ARR) ... 1705
Estrutura do artefato de migração ... 1705
Cenários avançados de migração ... 1707
Migrações de vários sites com Roteamento de Solicitação de Aplicativos (ARR) ... 1707
Migrações de vários sites sem ARR usando roteamento baseado em host ... 1710
Gerenciamento de diretórios virtuais ... 1714
Configurações personalizadas do pool de aplicativos ... 1718
Implantação de versões anteriores ... 1720
Solução de problemas e diagnósticos ... 1721
Associando um EC2 par de chaves ao seu ambiente ... 1721
Acessando registros ... 1722
Acessando artefatos do lado do cliente ... 1722
Monitorando a saúde do ambiente ... 1722
EC2 otimização de desempenho ... 1723
Configuração de volume do EBS ... 1723
Problemas e soluções comuns de ... 1724
Obter suporte ... 1725
Opções de migração: EB CLI versus MGN ... 1725
Quando usar cada opção de migração ... 1726

Comparação do fluxo de trabalho de migração 1727
Conclusão 1728
Solução de problemas 1729
Usar a ferramenta Systems Manager 1730
Orientações gerais 1731
Variáveis de ambiente para segredos 1732
Criação de ambiente 1733
Implantações 1734
Integridade 1734
Configuração 1735
Docker 1736
Perguntas frequentes 1736
Solução de problemas de implantações 1738
Erros de implantação 1738
Recursos 1742
Aplicações de exemplo 1743
AWS SDK para Java 1744
AWS SDK para .NET 1744
AWS Toolkit for Visual Studio 1744
AWS SDK para JavaScript em Node.js 1745
AWS SDK para PHP 1745
AWS SDK for Python (Boto) 1745
AWS SDK para Ruby 1745
Histórico do documento 1747
..... mdccliii

O que é AWS Elastic Beanstalk?

Com o Elastic Beanstalk, você pode implantar aplicativos web na nuvem AWS em uma variedade de plataformas compatíveis. Você cria e implanta seus aplicativos. O Elastic Beanstalk EC2 provisiona instâncias da Amazon, configura o balanceamento de carga, configura o monitoramento da integridade e escala dinamicamente seu ambiente.

Além dos ambientes de servidor web, o Elastic Beanstalk também fornece ambientes de trabalho que você pode usar para processar mensagens de uma fila do Amazon SQS, úteis para tarefas assíncronas ou de longa execução. Para obter mais informações, consulte Ambientes de operador do Elastic Beanstalk.



Plataformas compatíveis

O Elastic Beanstalk é compatível com aplicações desenvolvidas em Go, Java, .NET, Node.js, PHP, Python e Ruby. O Elastic Beanstalk também oferece suporte a contêineres Docker, onde você pode escolher sua própria linguagem de programação e dependências de aplicativos. Quando você

implanta seu aplicativo, o Elastic Beanstalk cria a versão da plataforma compatível selecionada e provisiona um ou AWS mais recursos, como instâncias da EC2 Amazon, AWS em sua conta para executar seu aplicativo.

Você pode interagir com o Elastic Beanstalk por meio do AWS Command Line Interface console do Elastic Beanstalk, do () ou do EB CLI AWS CLI, uma ferramenta de linha de comando de alto nível projetada especificamente para o Elastic Beanstalk.

Você pode realizar a maioria das tarefas de implantação, como alterar o tamanho da sua frota de EC2 instâncias da Amazon ou monitorar sua aplicação, diretamente da interface web (console) do Elastic Beanstalk.

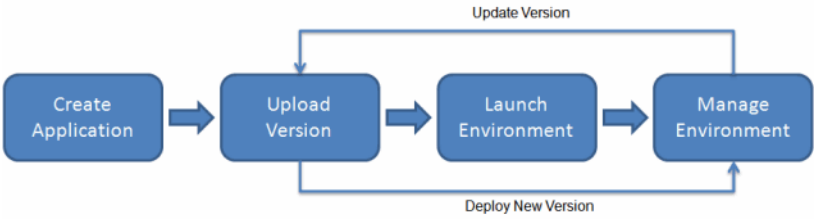
Para saber mais sobre como implantar um aplicativo web de amostra usando o Elastic Beanstalk, consulte. Saiba como começar a usar o Elastic Beanstalk

Fluxo de trabalho de implantação de

Para usar o Elastic Beanstalk, você cria um aplicativo e, em seguida, carrega seu pacote de origem do aplicativo no Elastic Beanstalk. Em seguida, você fornece informações sobre o aplicativo, e o Elastic Beanstalk inicia automaticamente um ambiente e cria e AWS configura os recursos necessários para executar seu código.

Depois de criar e implantar seu aplicativo e seu ambiente ser lançado, você pode gerenciar seu ambiente e implantar novas versões do aplicativo. As informações sobre o aplicativo, incluindo métricas, eventos e status do ambiente, são disponibilizadas por meio do console do Elastic Beanstalk e das interfaces de linha de comando. APIs

O diagrama a seguir ilustra o fluxo de trabalho do Elastic Beanstalk:



Preços

Não há custo adicional para o Elastic Beanstalk. Você paga somente pelos AWS recursos subjacentes que seu aplicativo consome. Para obter detalhes sobre definição de preço, consulte a página de detalhes do serviço Elastic Beanstalk.

Próximas etapas

Recomendamos o tutorial,Tutorial de inicialização, para começar a usar o Elastic Beanstalk. O tutorial orienta você na criação, visualização e atualização de uma amostra do aplicativo Elastic Beanstalk.

Saiba como começar a usar o Elastic Beanstalk

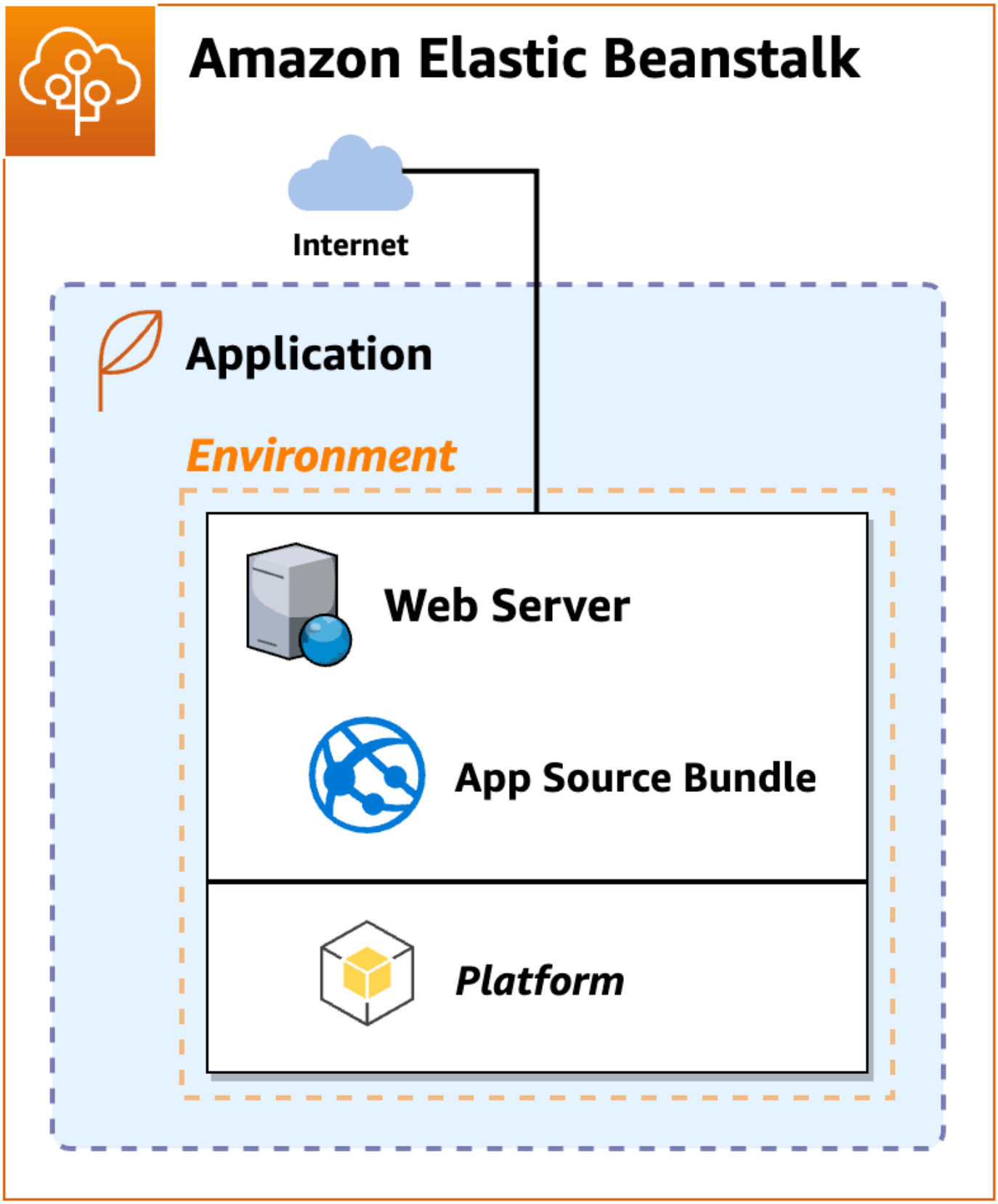
Com o Elastic Beanstalk, você pode implantar, monitorar e escalar aplicativos e serviços web. Normalmente, você desenvolverá seu código localmente e o implantará nas instâncias EC2 do servidor Amazon. Essas instâncias, também chamadas de ambientes, são executadas em plataformas que podem ser atualizadas por meio do AWS console ou da linha de comando.

Para começar, recomendamos a implantação de um aplicativo de amostra pré-criado diretamente do console. Em seguida, você pode aprender como desenvolver localmente e implantar a partir da linha de comando nothe section called “QuickStart para PHP”.

Não há custo para usar o Elastic Beanstalk, mas as taxas padrão AWS se aplicam aos recursos que você cria durante o curso deste tutorial até que você os exclua no final. As cobranças totais geralmente são inferiores a um dólar. Para obter informações sobre como minimizar as cobranças, consulte nível gratuito da AWS.

Depois de concluir este tutorial, você entenderá os fundamentos da criação, configuração, implantação, atualização e monitoramento de um aplicativo do Elastic Beanstalk com ambientes executados em instâncias da Amazon. EC2

Duração estimada: 35-45 minutos


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

O que você vai construir

Seu primeiro aplicativo do Elastic Beanstalk consistirá em um único ambiente EC2 Amazon executando a amostra de PHP em uma plataforma gerenciada por PHP.

Aplicação Elastic Beanstalk

Um aplicativo do Elastic Beanstalk é um contêiner para componentes do Elastic Beanstalk, incluindo ambientes em que o código do seu aplicativo é executado em plataformas fornecidas e gerenciadas pelo Elastic Beanstalk, ou em contêineres personalizados fornecidos por você.

Environment

Um ambiente do Elastic Beanstalk é uma AWS coleção de recursos que funcionam juntos, incluindo uma instância da Amazon. EC2 Quando você cria um ambiente, o Elastic Beanstalk provisiona os recursos necessários em sua conta. AWS

Plataforma

Uma plataforma é uma combinação de sistema operacional, tempo de execução da linguagem de programação, servidor web, servidor de aplicativos e componentes adicionais do Elastic Beanstalk. O Elastic Beanstalk fornece plataformas gerenciadas, ou você pode fornecer sua própria plataforma em um contêiner.

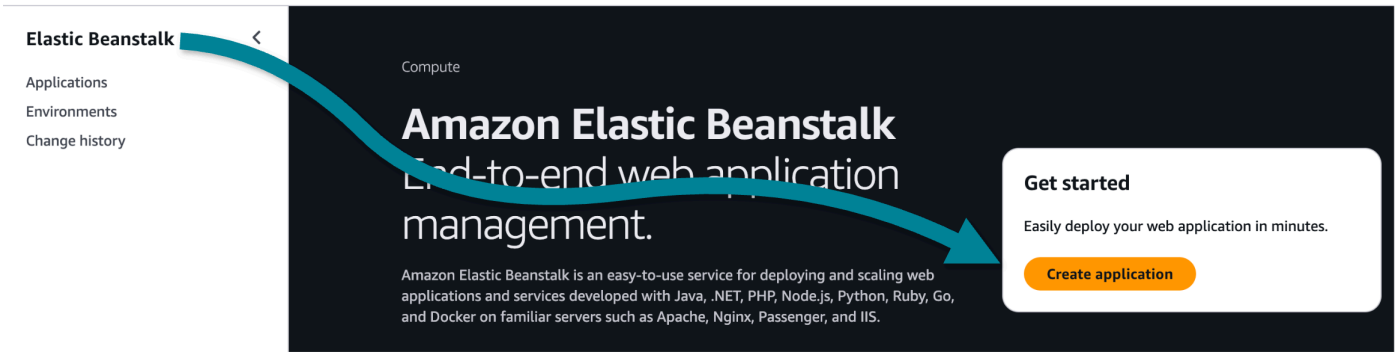
O Elastic Beanstalk é compatível com plataformas para diferentes linguagens de programação, servidores de aplicações e contêineres do Docker. Ao criar um ambiente, você deve escolher a plataforma. Você pode atualizar a plataforma, mas não pode alterar a plataforma para um ambiente.

> **Plataformas de troca**
>
> Se você precisar alterar as linguagens de programação, deverá criar e mudar para um novo ambiente em uma plataforma diferente.

Etapa 1 - Crie um aplicativo

Para criar seu exemplo de aplicação, você usará o assistente do console Create application (Criar aplicação). Ele cria uma aplicação do Elastic Beanstalk e inicia um ambiente nela.

Lembrete: um ambiente é uma coleção de AWS recursos necessários para executar o código do aplicativo.



Para criar uma aplicação

1. Abra o console do Elastic Beanstalk.
2. Selecione Criar aplicativo.
3. Em Application name (Nome da aplicação), insira **`getting-started-app`**.

O console fornece um processo de seis etapas para criar um aplicativo e configurar um ambiente. Para esse início rápido, você só precisará se concentrar nas duas primeiras etapas e, em seguida, poderá avançar para revisar e criar seu aplicativo e ambiente.

Para configurar um ambiente

1. Em Informações do ambiente, em Nome do ambiente, digite:**`gs-app-web-env`**.
2. Em Plataforma, escolha a plataforma PHP.
3. Para Código do aplicativo e predefinições, aceite os padrões (aplicativo de amostra e instância única) e escolha Avançar.

Para configurar o acesso ao serviço

Em seguida, você precisa de duas funções. Uma função de serviço permite que o Elastic Beanstalk EC2 monitore suas instâncias e atualize a plataforma do seu ambiente. Uma função de perfil de EC2 instância permite tarefas como gravar registros e interagir com outros serviços.

Para criar a função de serviço

1. Em Função de serviço, escolha Criar função.
2. Em Tipo de entidade confiável, escolha Serviços da AWS .
3. Para o caso de uso, escolha Elastic Beanstalk — Ambiente.
4. Escolha Próximo.
5. Verifique se as políticas de permissões incluem o seguinte e escolha Avançar:
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`
6. Selecione Criar perfil.
7. Volte para a guia Configurar acesso ao serviço, atualize a lista e selecione a função de serviço recém-criada.

Para criar o perfil da EC2 instância

1. Selecione Criar perfil.
2. Em Tipo de entidade confiável, escolha Serviços da AWS .
3. Para o caso de uso, escolha Elastic Beanstalk — Compute.
4. Escolha Próximo.
5. Verifique se as políticas de permissões incluem o seguinte e escolha Avançar:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
6. Selecione Criar perfil.
7. Volte para a guia Configurar acesso ao serviço, atualize a lista e selecione o perfil de EC2 instância recém-criado.

Para finalizar a configuração e a criação do seu aplicativo

1. Ignore o par de EC2 chaves.

   Mostraremos outras formas de se conectar às suas EC2 instâncias da Amazon por meio do console.

2. Escolha Ir para revisão para passar por várias etapas opcionais.

   Etapas opcionais: rede, bancos de dados, parâmetros de escalabilidade, configuração avançada para atualizações, monitoramento e registro.

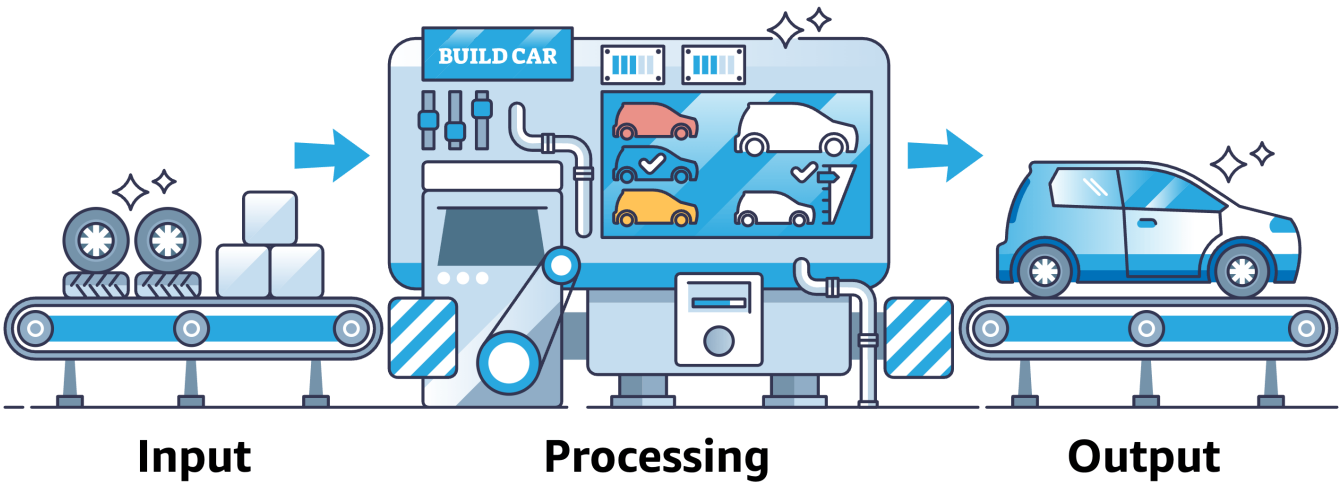
3. Na página de revisão, que mostra um resumo de suas escolhas, escolha Enviar.

> **Parabéns!**
>
> Você criou um aplicativo e configurou um ambiente! Agora você precisa aguardar a implantação dos recursos.

Etapa 2 - Implantar seu aplicativo

Quando você cria um aplicativo, o Elastic Beanstalk configura os ambientes para você. Você só precisa sentar e esperar.

A implantação inicial pode levar até cinco minutos para criar os recursos. As atualizações levarão menos tempo porque somente as alterações serão implantadas em sua pilha.



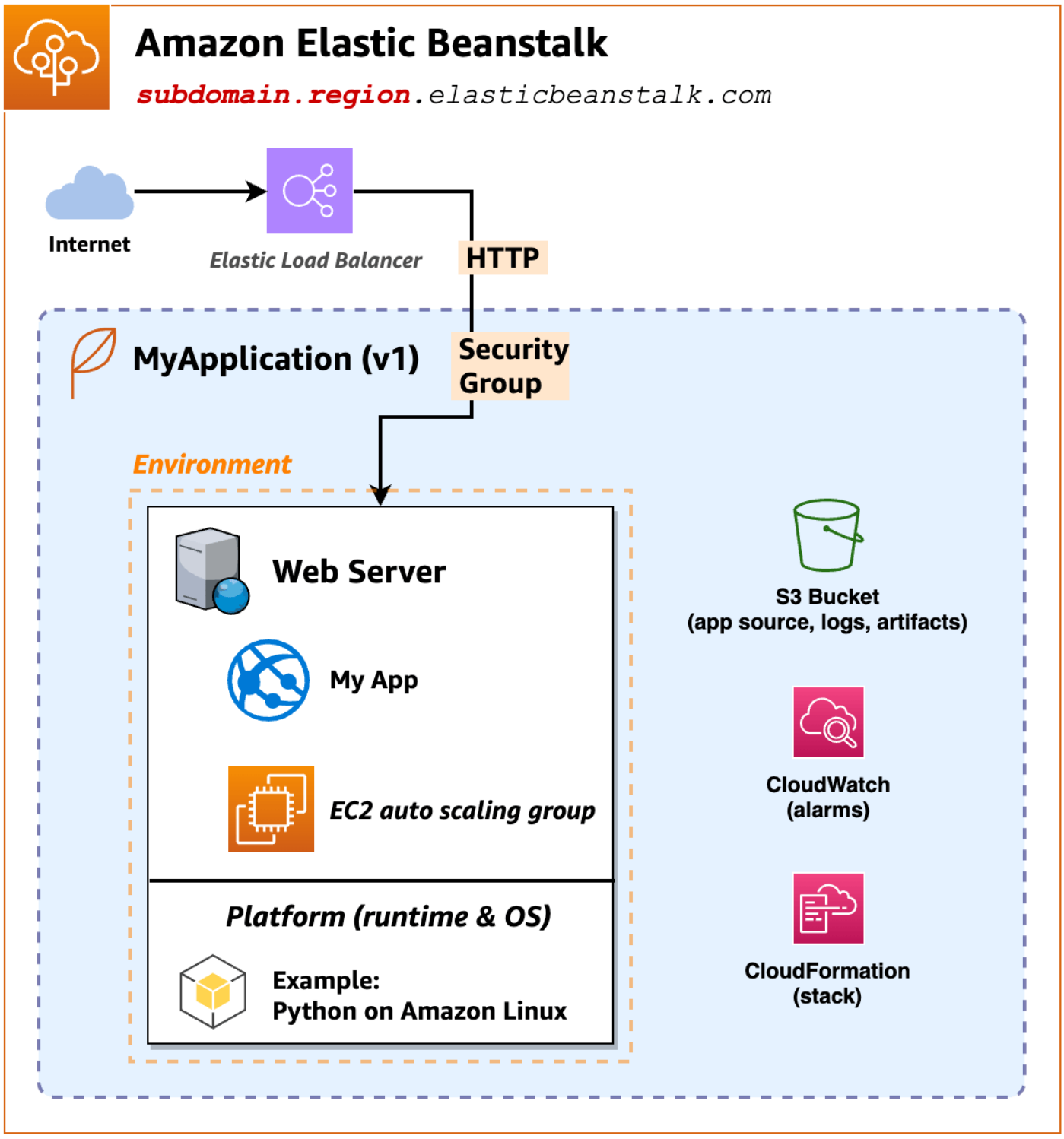
Quando você cria a aplicação demonstrativa, o Elastic Beanstalk cria os seguintes recursos da :

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você selecionou.

Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão específica de linguagem, estrutura, contêiner da web ou combinação dos mesmos. A maioria das plataformas usa o Apache ou o nginx como proxy reverso para encaminhar o tráfego da web para seu aplicativo web, fornecer ativos estáticos e gerar registros de acesso e erros. Você pode se conectar às suas EC2 instâncias da Amazon para visualizar a configuração e os registros.

- Grupo de segurança da instância — Um grupo de EC2 segurança da Amazon será criado para permitir a entrada de solicitações na porta 80, para que o tráfego de entrada em um balanceador de carga possa acessar seu aplicativo web.
- Bucket Amazon S3 — Um local de armazenamento para seu código-fonte, registros e outros artefatos.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes são criados para monitorar a carga em suas instâncias e aumentá-las ou reduzi-las conforme necessário.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para implantar os recursos em seu ambiente e fazer alterações na configuração. Você pode visualizar o modelo de definição de recursos no AWS CloudFormation console.
- Nome de domínio - Um nome de domínio que é direcionado para seu aplicativo da web no formato:`subdomain.region.elasticbeanstalk.com`.

O Elastic Beanstalk cria seu aplicativo, inicia um ambiente, cria uma versão do aplicativo e, em seguida, implanta seu código no ambiente. Durante o processo, o console acompanha o progresso e exibe o status do evento na guia Eventos.



Depois que todos os recursos forem implantados, a saúde do ambiente deve mudar para**0k**.

Sua inscrição está pronta!

Depois de ver a integridade do seu aplicativo mudar para**0k**, você pode navegar até o site do seu aplicativo web.

Etapa 3 - Explore o ambiente do Elastic Beanstalk

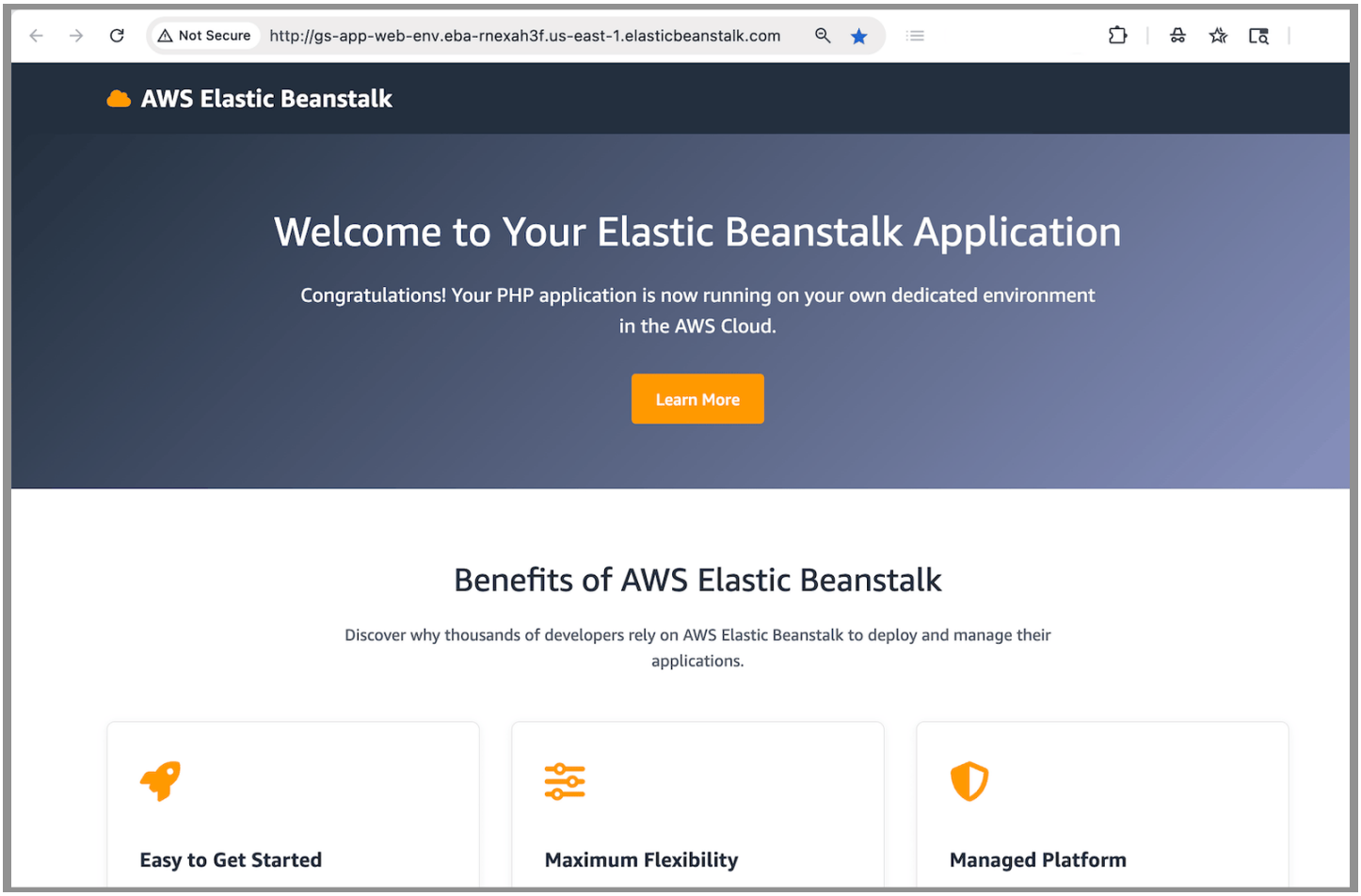
Você começará a explorar seu ambiente de aplicativos implantado na página de visão geral do ambiente no console.

Para visualizar o ambiente e seu aplicativo

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Ir para o ambiente para navegar em seu aplicativo!

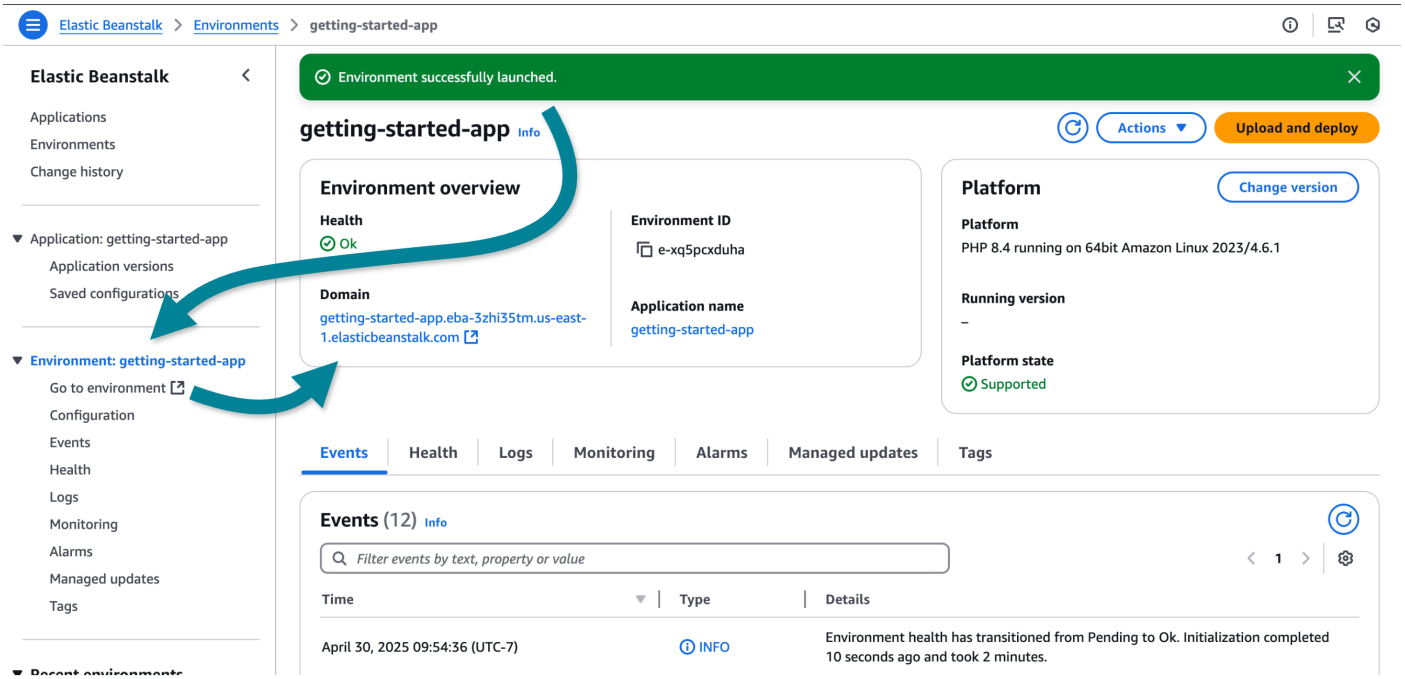
   (Você também pode escolher o link de URL listado para Domínio para navegar em seu aplicativo.)

   A conexão será HTTP (não HTTPS), então você pode ver um aviso no seu navegador.



De volta ao console do Elastic Beanstalk, a parte superior mostra a visão geral do ambiente com informações de alto nível sobre seu ambiente, incluindo nome, URL do domínio, status de integridade atual, versão em execução e a plataforma na qual o aplicativo está sendo executado. A versão e a plataforma em execução são essenciais para solucionar problemas de seu aplicativo atualmente implantado.

Depois do painel de visão geral, você verá a atividade recente do ambiente na guia Eventos.



Enquanto o Elastic Beanstalk AWS cria seus recursos e inicia seu aplicativo, o ambiente está em um estado. `Pending` Mensagens de status sobre eventos de lançamento são adicionadas continuamente à lista de eventos.

O domínio do ambiente é a URL do seu aplicativo web implantado. No painel de navegação esquerdo, Ir para o ambiente também leva você ao seu domínio. Da mesma forma, o painel de navegação esquerdo tem links que correspondem às várias guias.

Anote o link Configuração no painel de navegação esquerdo, que exibe um resumo dos valores das opções de configuração do ambiente, agrupados por categoria.

> **Definições de configuração do ambiente**
>
> Anote o link Configuração no painel de navegação esquerdo. Você pode visualizar e editar configurações detalhadas do ambiente, como funções de serviço, rede, banco de dados, escalabilidade, atualizações gerenciadas da plataforma, memória, monitoramento de integridade, implantação contínua, registro e muito mais!

As várias guias contêm informações detalhadas sobre seu ambiente:

- Eventos — Visualize uma lista atualizada de informações e mensagens de erro do serviço Elastic Beanstalk e de outros serviços para recursos em seu ambiente.
- Health — Visualize o status e as informações detalhadas de saúde das EC2 instâncias da Amazon que executam seu aplicativo.
- Registros — Recupere e baixe registros da Amazon EC2 em seu ambiente. Você pode recuperar logs completos ou as atividades recentes. Os logs recuperados ficam disponíveis por 15 minutos.
- Monitoramento — Visualize estatísticas do ambiente, como latência média e utilização da CPU.
- Alarmes — Visualize e edite alarmes configurados para métricas do ambiente.
- Atualizações gerenciadas — Veja informações sobre atualizações futuras e concluídas da plataforma gerenciada e substituição de instâncias.
- Tags — Visualize e edite pares de valores-chave que são aplicados ao seu ambiente.

> **Note**
>
> Os links no painel de navegação do console exibirão a guia correspondente.

Solução de problemas com logs

Para solucionar comportamentos inesperados ou depurar implantações, talvez você queira verificar os registros em seus ambientes.

Você pode solicitar 100 linhas de todos os arquivos de log na guia Logs no console do Elastic Beanstalk. Como alternativa, você pode se conectar diretamente à EC2 instância da Amazon e rastrear os registros em tempo real.

Para solicitar os registros (console do Elastic Beanstalk)

1. Navegue até seu ambiente no console do Elastic Beanstalk.
2. Escolha a guia Registros ou a navegação à esquerda e escolha Solicitar registros.
3. Selecione Últimas 100 linhas.
4. Depois que os registros forem criados, escolha o link Download para visualizá-los no navegador.

Nos registros, localize o registro e anote o diretório do registro de acesso do nginx.

Adicione uma política para habilitar conexões com a Amazon EC2

Antes de se conectar, você deve adicionar uma política que permita conexões com a Amazon EC2 com o Session Manager.

1. Navegue até o console do IAM.
2. Encontre e selecione a **`aws-elasticbeanstalk-ec2-role`** função.
3. Escolha Adicionar permissão e, em seguida, Anexar políticas.
4. Procure uma política padrão que comece com o seguinte texto: **`AmazonSSMManagedEC2Instance`** e adicione-a à função.

Para se conectar à sua Amazon EC2 com o Session Manager

1. Navegue até o EC2 console da Amazon.
2. Escolha Instâncias e, em seguida, selecione sua `gs-app-web-env` instância.
3. Escolha Connect e, em seguida, Session Manager.
4. Selecione Conectar.

Depois de se conectar à instância, inicie um shell bash e siga os registros:

1. Execute o comando **`bash`**.
2. Execute o comando **`cd /var/log/nginx`**.
3. Execute o comando **`tail -f access.log`**.
4. No seu navegador, acesse o URL do domínio do aplicativo. Atualizar.

> Parabéns, você está conectado!
>
> Você deve ver as entradas de registro na atualização da sua instância sempre que atualizar a página.

> O botão Connect não está funcionando?
>
> Se o botão de conexão não estiver disponível, volte para o IAM e verifique se você adicionou a política necessária à função.

Etapa 4 - Atualize seu aplicativo

Eventualmente, você desejará atualizar seu aplicativo. Você pode implantar uma nova versão do aplicativo a qualquer momento, desde que não haja outras operações de atualização em andamento no ambiente.

A versão do aplicativo com a qual você iniciou este tutorial é chamada de Sample Application (Aplicativo de exemplo).



Para atualizar a versão do aplicativo

1. Faça o download do seguinte aplicativo de amostra em PHP:

   PHP — php-v2.zip
2. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
4. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
5. Selecione Escolher arquivo e carregue o pacote de origem da aplicação de exemplo que você baixou.

O console preenche automaticamente o rótulo da versão com um novo rótulo exclusivo, incrementando automaticamente um número inteiro à direita. Se você escolher sua própria etiqueta de versão, certifique-se de que ela seja exclusiva.

6. Escolha Implantar.

Enquanto o Elastic Beanstalk implanta seu arquivo em suas instâncias da Amazon, você pode ver o status da implantação na página de visão geral do ambiente. EC2 O status de Integridade do ambiente fica acinzentado enquanto a versão da aplicação é atualizada. Quando a implantação estiver concluída, o Elastic Beanstalk executará uma verificação de integridade da aplicação. Quando o aplicativo responde à verificação de integridade, ela é considerada íntegra e o status retorna para verde. A visão geral do ambiente exibe a nova Running Version (Versão em execução): o nome fornecido como Version label (Rótulo da versão).

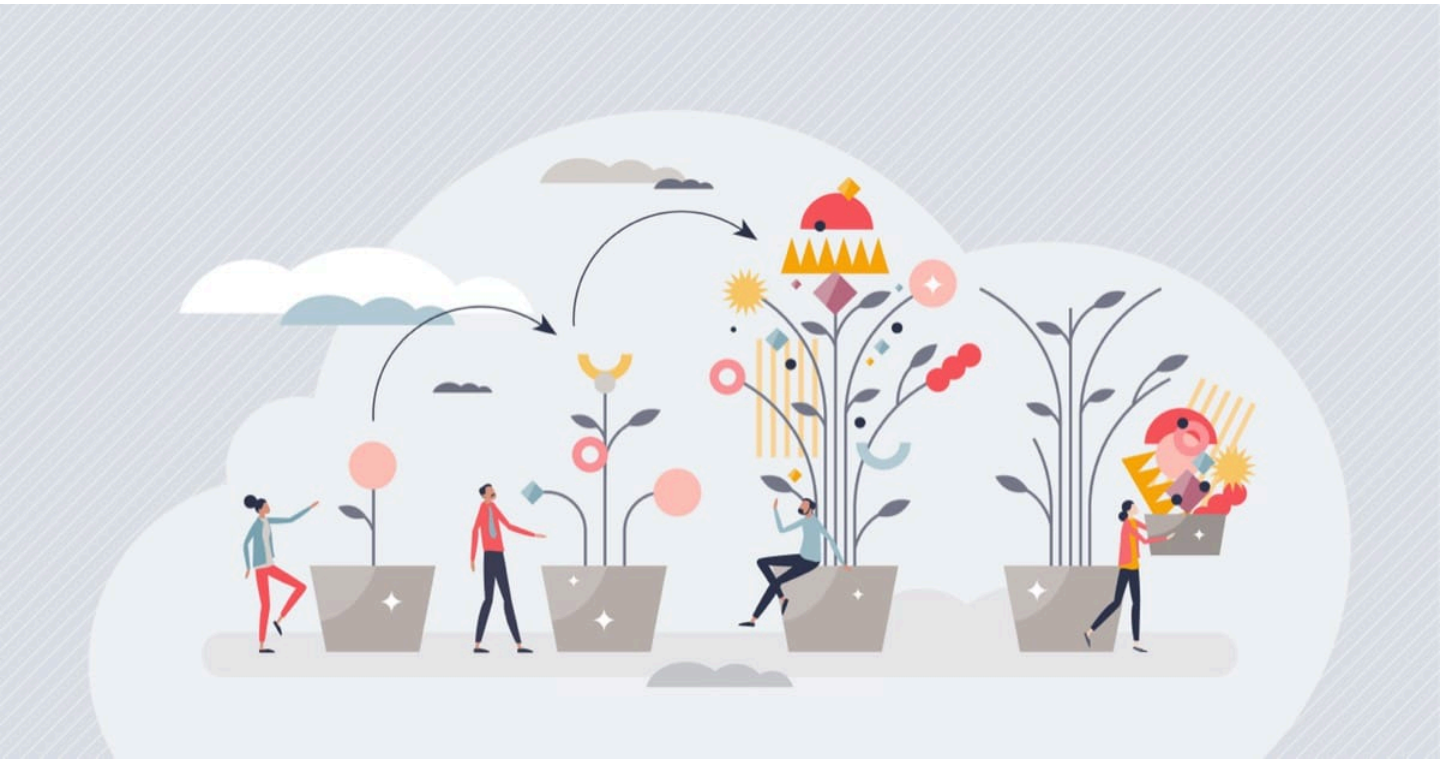
A nova versão da aplicação do Elastic Beanstalk também é carregada e adicionada à tabela de versões da aplicação. Para ver a tabela, escolha Versões do aplicativo abaixo getting-started-appno painel de navegação.

> **Atualização bem-sucedida!**
>
> Você deve ver uma mensagem “v2" atualizada depois de atualizar seu navegador.
> Se você quiser editar a fonte sozinho, descompacte, edite e, em seguida, compacte novamente o pacote de origem. No macOS, use o seguinte comando de dentro do seu diretório php com o -X para excluir atributos extras do arquivo:
> `zip -X -r ../php-v2.zip .`

Etapa 5 - Dimensione seu aplicativo

Você pode configurar o ambiente para melhor atender ao aplicativo. Por exemplo, se você tiver um aplicativo com uso intensivo de computação, poderá alterar o tipo de instância do Amazon Elastic Compute Cloud (Amazon EC2) que está executando seu aplicativo. Para aplicar alterações de configuração, o Elastic Beanstalk executa uma atualização de ambiente.

Algumas mudanças de configuração são simples e ocorrem de maneira rápida. Algumas alterações exigem a exclusão e a recriação de AWS recursos, o que pode levar alguns minutos. Quando você altera as configurações, o Elastic Beanstalk avisa sobre possíveis períodos de inatividade da aplicação.



Aumentar as configurações de capacidade

Neste exemplo de alteração de configuração, você edita as configurações de capacidade do ambiente. Você configura um ambiente escalável e com balanceamento de carga que tem entre duas e quatro EC2 instâncias da Amazon em seu grupo de Auto Scaling e, em seguida, verifica se a alteração ocorreu. O Elastic Beanstalk cria uma instância EC2 adicional da Amazon, adicionada à instância única que ele criou inicialmente. Depois, o Elastic Beanstalk associa as duas instâncias ao balanceador de carga do ambiente. Como resultado, a capacidade de resposta do aplicativo é aprimorada e sua disponibilidade é aumentada.

Como alterar a capacidade do seu ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
5. Feche a seção Instances (instâncias) para poder ver mais facilmente a seção Capacity (Capacidade). Na seção Auto Scaling Group (Grupo do Auto Scaling), altere Environment type (Tipo de ambiente) para Load balanced (Carga balanceada).
6. Na linha Instâncias, altere Min para **2** e Max para**4**.

7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

   Se você for avisado de que a atualização substituirá todas as suas instâncias atuais. Escolha Confirmar.

A atualização do ambiente pode levar alguns minutos. Você deve ver várias atualizações na lista de eventos. Assista ao evento Nova configuração implantada com sucesso no ambiente.

Verifique o aumento da capacidade

Depois que a atualização do ambiente estiver concluída e o ambiente estiver pronto, o Elastic Beanstalk iniciará automaticamente uma segunda instância para atender à sua nova configuração de capacidade mínima.

Como verificar o aumento da capacidade

1. Escolha Health na lista de guias ou no painel de navegação esquerdo.
2. Consulte a seção Integridade aprimorada da instância.

> Você acabou de aumentar a escala!
>
> Com duas EC2 instâncias da Amazon, a capacidade do seu ambiente dobrou e levou apenas alguns minutos.

Limpando seu ambiente do Elastic Beanstalk

Para garantir que você não seja cobrado por nenhum serviço que não esteja usando, exclua todas as versões do aplicativo e encerre os ambientes, o que também exclui os AWS recursos que o ambiente criou para você.



Excluir o aplicativo e todos os recursos associados

1. Excluir todas as versões do aplicativo.
    a. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
    b. No painel de navegação, escolha Aplicações e, depois, getting-started-app.
    c. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
    d. Na página Application versions (Versões do aplicativo), selecione todas as versões do aplicativo que você deseja excluir.
    e. Escolha Ações e, em seguida, escolha Excluir.
    f. Ative Delete versions from Amazon S3 (Excluir versões do Amazon S3).
    g. Escolha Delete (Excluir) e Done (Concluído).
2. Encerrar o ambiente.

a. No painel de navegação, escolha e, em seguida getting-started-app, escolha GettingStartedApp-env na lista de ambientes.
b. Escolha Actions e, em seguida, Terminate Environment.
c. Confirme que você deseja encerrar GettingStartedApp-env digitando o nome do ambiente e, em seguida, escolha Encerrar.

3. Exclua a aplicação do getting-started-app.

a. No painel de navegação, escolha o. getting-started-app
b. Escolha Ações e Excluir aplicativo.
c. Confirme que você deseja excluir getting-started-appdigitando o nome do aplicativo e escolha Excluir.

Parabéns!

Você implantou com sucesso um aplicativo de amostra AWS na nuvem, fez o upload de uma nova versão, modificou sua configuração para adicionar uma segunda instância do Auto Scaling e limpou AWS seus recursos!

Próximas etapas

Para aprender a usar a ferramenta de linha de **eb** comando para automatizar a implantação do seu código no Elastic Beanstalk, sugerimos continuar com o. the section called “QuickStart para PHP”

Em seguida, talvez você queira revisar como configurar a conexão HTTPS, consultethe section called “HTTPS”.

Configuração da interface de linha de comando do EB (EB CLI) para gerenciar o Elastic Beanstalk

O EB CLI é uma interface de linha de comando que fornece comandos interativos para criar, atualizar e monitorar ambientes em. AWS Elastic Beanstalk O projeto de código aberto EB CLI está no Github: aws/aws-elastic-beanstalk-cli

Depois de instalar a CLI do EB e configurar um diretório de projeto, é possível criar ambientes com um único comando:

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

Recomendamos instalar com o script de configuração, saiba comothe section called “Instale o EB CLI”.

O AWS CLI fornece acesso direto ao Elastic APIs Beanstalk de baixo nível. Embora poderoso, ele também é detalhado e menos preferido em relação ao EB CLI. Por exemplo, criar um ambiente com o AWS CLI requer a seguinte série de comandos:

```
$ 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"
```

Instale o EB CLI com script de configuração (recomendado)

> **Recomendamos o script do instalador**
>
> Recomendamos usar o script do instalador para configurar a CLI do EB e suas dependências e evitar possíveis conflitos com outros pacotes Python.

Pré-requisitos: Git, Python e virtualenv

Para baixar e usar o script do instalador

1. Clonar o repositório.

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

2. Instale ou atualize o EB CLI.

   macOS/Linux em Bash ou Zsh

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

   Windows em PowerShell ou janela de comando

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

3. Verifique se a EB CLI está instalada corretamente.

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

Para obter instruções completas de instalação, consulte o aws/aws-elastic-beanstalk-cli-setuprepositório em. GitHub

Instalar a CLI do EB manualmente

Você pode instalar a CLI do EB no Linux, macOS e Windows com o `pip` gerenciador de pacotes para Python, que fornece instalação, atualização e remoção de pacotes Python e suas dependências.

> **Recomendamos o script do instalador**
>
> Recomendamos usar o Instale o EB CLI para configurar a CLI do EB e evitar conflitos de dependência.

Pré-requisito - Você deve ter uma versão compatível do Python instalada. Você pode baixá-lo na página de downloads do Python no site do Python.

Para instalar o EB CLI (manualmente)

1. Execute o seguinte comando:

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

   A opção `--upgrade` informa ao `pip` para atualizar os requisitos que já estão instalados. A opção `--user` informa ao `pip` para instalar o programa em um subdiretório do diretório de usuário para evitar a modificação de bibliotecas usadas pelo seu sistema operacional.

   > **Solução de problemas**
   >
   > Caso haja problemas ao tentar instalar a CLI do EB com o `pip`, você poderá instalar a CLI do EB em um ambiente virtual para isolar a ferramenta e suas dependências, ou usar uma versão do Python diferente da que você normalmente usa.

2. Adicione o caminho para o arquivo executável da sua variável `PATH`:
   - No Linux e macOS:

     Linux: `~/.local/bin`

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

     Para modificar a variável `PATH` (Linux, Unix ou macOS):

a. Encontre o script de perfil do shell em sua pasta de usuário. Se você não tiver certeza de qual shell você tem, execute `echo $SHELL`.

```
$ ls -a ~
.  ..  .bash_logout  .bash_profile  .bashrc  Desktop  Documents  Downloads
```

- Bash: `.bash_profile`, `.profile` ou `.bash_login`.
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` ou `.login`.

b. Adicione um comando de exportação ao script de perfil. O exemplo a seguir adiciona o caminho representado por *LOCAL_PATH* à PATH variável atual.

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

c. Carregue o script de perfil descrito na primeira etapa para a sessão atual. O exemplo a seguir carrega o script de perfil representado por*PROFILE_SCRIPT*.

```
$ source ~/PROFILE_SCRIPT
```

- No Windows:

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

  Versões anteriores do Python: `%USERPROFILE%\AppData\Roaming\Python\Scripts`

  Para modificar a variável PATH (Windows):

  a. Pressione a tecla Windows e insira **`environment variables`**.

  b. Escolha Edit environment variables for your account.

  c. Selecione PATH e, em seguida, Edit (Editar).

  d. Adicione caminhos ao campo Variable value, separados por ponto e vírgula. Por exemplo: *`C:\item1\path`***`;C:\item2\path`**

  e. Selecione OK duas vezes para aplicar as novas configurações.

  f. Feche todas as janelas do prompt de comando em execução e abra novamente uma janela do prompt de comando.

3. Verifique se a EB CLI foi instalada corretamente executando eb --version.

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

A CLI do EB é atualizada regularmente para adicionar as funcionalidades compatíveis com os recursos mais recentes do Elastic Beanstalk. Para atualizar para a versão mais recente da EB CLI, execute o comando de instalação novamente.

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

Se você precisar desinstalar a EB CLI, use `pip uninstall`.

```
$ pip uninstall awsebcli
```

Instalar a EB CLI em um ambiente virtual

Você pode evitar conflitos de requisitos de versão com outros pacotes `pip` instalando a EB CLI em um ambiente virtual.

Para instalar a EB CLI em um ambiente virtual

1. Primeiro, instale `virtualenv` com`pip`.

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

2. Crie um ambiente virtual.

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

   Para usar um executável Python diferente da padrão, use a opção `-p`.

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

3. Ative o ambiente virtual.

   Linux, Unix ou macOS

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

Windows

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

4. Instale a EB CLI.

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

5. Verifique se a EB CLI está instalada corretamente.

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

Use o comando `deactivate` para sair do ambiente virtual. Sempre que iniciar uma nova sessão, execute o comando de ativação novamente.

Para atualizar para a versão mais recente, execute o comando de instalação novamente.

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

Instalar a EB CLI com o homebrew

A versão mais recente da CLI do EB geralmente fica disponível no `Homebrew` alguns dias depois de aparecer no `pip`.

> ⚠ Recomendamos o script do instalador
>
> Recomendamos usar o Instale o EB CLI para configurar a CLI do EB e evitar conflitos de dependência.

Para instalar a CLI do EB com o **`Homebrew`**

1. Verifique se você tem a versão mais recente do `Homebrew`.

```
$ brew update
```

2. Executar `brew install awsebcli`.

```
$ brew install awsebcli
```

3. Verifique se a EB CLI está instalada corretamente.

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

Configurar a EB CLI

Depois de instalar o EB CLI, você estará pronto para configurar o diretório do projeto e o EB CLI executandoeb init. O exemplo a seguir mostra as etapas de configuração ao executar eb init pela primeira vez em uma pasta de projeto chamada. eb

Para inicializar um projeto da EB CLI

1. Primeiro, a EB CLI solicita que seja selecionada uma região. Escolha sua região preferida.

```
~/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 solicitado, forneça sua chave de acesso e chave secreta para que o EB CLI possa gerenciar recursos para você. As chaves de acesso são criadas no AWS Identity and Access Management console. Se você não tiver chaves, consulte Como obter credenciais de segurança? na Referência geral da 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. Um aplicativo do Elastic Beanstalk é um recurso que contém um conjunto de versões do aplicativo (origem), ambientes e configurações salvas que estão associados a um único aplicativo web. Cada vez que você implanta o seu código-fonte no Elastic Beanstalk usando a CLI do EB, uma nova versão da aplicação é criada e adicionada à lista.

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

4. O nome do aplicativo padrão é o nome da pasta na qual você executa eb init. Insira um nome que descreva o projeto.

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

5. Selecione uma plataforma que corresponda à linguagem ou framework usados para desenvolver seu aplicativo Web. Se você ainda não começou a desenvolver um aplicativo, escolha uma plataforma de seu interesse. Você verá como iniciar um aplicativo de exemplo rapidamente e poderá alterar essa configuração mais tarde.

```
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. Selecione uma ramificação específica da plataforma.

```
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. Escolha yes (sim) para atribuir um par de chaves SSH para as instâncias no ambiente do Elastic Beanstalk. Isso permite que você se conecte diretamente a elas e solucione problemas.

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

8. Escolha um par de chaves existente ou crie um novo. Para usar eb init para criar um novo par de chaves, instale ssh-keygen em sua máquina local e disponibilize-o na linha de comando. O EB CLI registra o novo par de chaves na Amazon EC2 para você e armazena a chave privada localmente em uma pasta nomeada `.ssh` em seu diretório de usuário.

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

Agora, a instalação da EB CLI está configurada e pronta para uso.

Configuração avançada

- Ignorar arquivos que usam .ebignore
- Usar perfis nomeados
- Implantar um artefato, em vez da pasta do projeto
- Definições de configuração e precedência
- Metadados da instância

Ignorar arquivos que usam .ebignore

Você pode instruir a EB CLI a ignorar determinados arquivos no diretório do projeto por meio da adição do arquivo `.ebignore` ao diretório. Esse arquivo funciona como um `.gitignore`. Quando você implanta o diretório do projeto no Elastic Beanstalk e cria uma versão da aplicação, a CLI do EB não inclui arquivos especificados pelo `.ebignore` no pacote de origem criado.

Se não houver nenhum `.ebignore`, mas houver um `.gitignore`, a EB CLI ignorará os arquivos especificados no `.gitignore`. Se houver um `.ebignore`, a EB CLI não fará a leitura do `.gitignore`.

Quando `.ebignore` estiver presente, a EB CLI não usará comandos git para criar o pacote de origem. Isso significa que a EB CLI ignora os arquivos especificados em `.ebignore` e inclui todos os outros. Especificamente, ela inclui os arquivos de origem não confirmados.

> **Note**
>
> No Windows, a adição de um `.ebignore` faz com que a EB CLI siga os links simbólicos e inclua o arquivo vinculado durante a criação de um pacote de origem. Esse é um problema conhecido e será corrigido em uma atualização futura.

Usar perfis nomeados

Se você armazenar suas credenciais como um perfil nomeado em um arquivo `credentials` ou `config`, poderá usar a opção `--profile` para especificar explicitamente um perfil. Por exemplo, o seguinte comando cria um novo aplicativo usando o perfil `user2`.

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

Você também pode alterar o perfil padrão, definindo a variável de ambiente `AWS_EB_PROFILE`. Quando essa variável está definida, a EB CLI faz a leitura das credenciais do perfil especificado, em vez de `default` ou eb-cli.

Linux, macOS ou Unix

```
$ export AWS_EB_PROFILE=user2
```

Windows

```
> set AWS_EB_PROFILE=user2
```

Implantar um artefato, em vez da pasta do projeto

Instrua a EB CLI a implantar um arquivo ZIP ou WAR que você gera como parte de um processo de construção separado, adicionando as linhas a seguir ao `.elasticbeanstalk/config.yml` na pasta do projeto.

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

Se você configurar a CLI do EB no repositório Git e não confirmar o artefato na fonte, use a opção `--staged` para implantar a compilação mais recente.

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

Definições de configuração e precedência

O EB CLI usa uma cadeia de fornecedores para procurar AWS credenciais em vários locais diferentes, incluindo variáveis de ambiente do sistema ou do usuário e arquivos de configuração locais AWS .

A EB CLI procura credenciais e definições de configuração na seguinte ordem:

1. Opções da linha de comando: especifica um perfil nomeado usando `--profile` para substituir as configurações padrão.
2. Variáveis de ambiente: `AWS_ACCESS_KEY_ID` e `AWS_SECRET_ACCESS_KEY`.
3. O arquivo de AWS credenciais — Localizado `~/.aws/credentials` nos sistemas Linux e OS X ou `C:\Users\USERNAME\.aws\credentials` nos sistemas Windows. Este arquivo pode conter vários perfis nomeados, além de um perfil padrão.
4. O arquivo de configuração da AWS CLI: localizado `~/.aws/config` em sistemas Linux e OS X ou `C:\Users\USERNAME\.aws\config` em sistemas Windows. Esse arquivo pode conter um perfil padrão, perfis nomeados e parâmetros de configuração AWS CLI específicos para cada um.
5. Arquivo de configuração da CLI do EB legada: localizado em `~/.elasticbeanstalk/config` em sistemas Linux e OS X ou `C:\Users\USERNAME\.elasticbeanstalk\config` em sistemas Windows.
6. Credenciais do perfil da instância — Essas credenciais podem ser usadas em EC2 instâncias da Amazon com uma função de instância atribuída e são fornecidas por meio do serviço de EC2 metadados da Amazon. O perfil da instância deve ter permissão para usar o Elastic Beanstalk.

Se o arquivo de credenciais contém um perfil nomeado com o nome "eb-cli", a EB CLI preferirá esse perfil em vez do perfil padrão. Se não for encontrado nenhum perfil, ou se for encontrado um perfil sem permissão para usar o Elastic Beanstalk, a CLI do EB solicitará a entrada de chaves.

Metadados da instância

Para usar o EB CLI de uma EC2 instância da Amazon, crie uma função que tenha acesso aos recursos necessários e atribua essa função à instância quando ela for executada. Inicie a instância e instale a EB CLI usando o `pip`.

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

O `pip` é fornecido pré-instalado no Amazon Linux.

A EB CLI lê as credenciais dos metadados da instância. Para obter mais informações, consulte Conceder acesso a AWS recursos para aplicativos executados em EC2 instâncias da Amazon no Guia do usuário do IAM.

Usar a EB CLI com o Git

O EB CLI fornece uma integração com o Git para que você possa associar ramificações de código a ambientes específicos no Elastic Beanstalk para organizar suas implantações.

Para instalar o Git e inicializar o repositório Git

1. Faça download da versão mais recente do Git acessando http://git-scm.com
2. Inicialize seu repositório Git digitando o seguinte:

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

   A EB CLI agora reconhece que seu aplicativo está configurado com o Git.
3. Se você ainda não executou eb init, faça isso agora:

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

Associar ambientes do Elastic Beanstalk a ramificações Git

Você pode associar um ambiente diferente com cada ramificação do seu código. Quando você faz o check-out de uma ramificação, as alterações são implantadas no ambiente associado. Por exemplo, você pode digitar o seguinte para associar seu ambiente de produção à sua ramificação mestre e um ambiente de desenvolvimento separado à sua ramificação de desenvolvimento:

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

Implantação de alterações

Por padrão, a EB CLI implanta a última confirmação na ramificação atual, usando a mensagem e o ID de confirmação como a descrição e o rótulo da versão do aplicativo, respectivamente. Se você deseja implantar em seu ambiente sem confirmar, pode usar a opção `--staged` para implantar as alterações que foram adicionadas à área de preparação.

Para implantar alterações sem confirmação

1. Adicione arquivos novos e alterados à área de preparação:

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

2. Implante as alterações preparadas com eb deploy:

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

Se você tiver configurado a CLI do EB para implantar um artefato e não confirmar o artefato no repositório Git, use a opção `--staged` para implantar a última compilação.

Uso dos submódulos do Git

Alguns projetos de código-fonte se beneficiam dos submódulos do Git: repositórios no repositório de nível superior. Quando você implementa seu código-fonte usando eb create ou eb deploy, o EB CLI pode incluir submódulos no arquivo zip da versão do aplicativo e fazer upload deles com o resto do código-fonte.

Você pode controlar a inclusão de submódulos usando a `include_git_submodules` opção na seção `global` do arquivo de configuração EB CLI, `.elasticbeanstalk/config.yml`, na pasta do projeto.

Para incluir submódulos, defina essa opção como `true`:

```
global:
  include_git_submodules: true
```

Quando a opção `include_git_submodules` está ausente ou é definida como `false`, a EB CLI não inclui submódulos no arquivo zip carregado.

Consulte Ferramentas do Git – Submódulos para obter mais detalhes sobre os submódulos do Git.

> **Comportamento padrão**
>
> Quando você executar eb init para configurar o seu projeto, a EB CLI adiciona a opção `include_git_submodules` e a define como `true`. Isso garante que todos os submódulos que você possui em seu projeto sejam incluídos em suas implantações.
> A EB CLI nem sempre ofereceu suporte à inclusão de submódulos. Para evitar uma alteração acidental e indesejável em projetos que já existiam antes de adicionarmos o suporte ao submódulo, a EB CLI não inclui submódulos quando a opção `include_git_submodules` estiver ausente. Se você tiver um desses projetos existentes e deseja incluir submódulos em suas implantações, adicione a opção e defina-a como `true` conforme explicado nesta seção.

> **CodeCommit comportamento**
>
> A integração do Elastic Beanstalk com CodeCommit não oferece suporte a submódulos no momento. Se você habilitou sua integração com o ambiente CodeCommit, os submódulos não estão incluídos em suas implantações.

Para atribuir tags do Git à versão do aplicativo

Você pode usar uma tag do Git como rótulo de versão para identificar qual versão do aplicativo está em execução em seu ambiente. Por exemplo, digite o seguinte:

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

Referência de comandos da EB CLI

É possível usar a Interface de Linhas de Comando (CLI do EB) do Elastic Beanstalk para executar uma variedade de operações para implantar e gerenciar as aplicações e os ambientes do Elastic Beanstalk. A EB CLI integra-se ao Git para implantar o código-fonte do aplicativo que está sob controle de código-fonte Git. Para obter mais informações, consulte Configuração da interface de linha de comando do EB (EB CLI) para gerenciar o Elastic Beanstalk e Usar a EB CLI com o Git.

Comandos

- Opções comuns
- 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

Opções comuns

Você pode usar as seguintes opções com todos os comandos da EB CLI.

Nome	Descrição
`--debug`	Imprimir informações para depuração.
`-h`, `--help`	Mostrar a mensagem de ajuda. Tipo: sequência Padrão: nenhum
`--no-verify-ssl`	Ignorar verificação de certificado SSL. Use essa opção se você tiver problemas ao utilizar a CLI com um proxy.
`--profile`	Use um perfil específico do seu arquivo de AWS credenciais.
`--quiet`	Suprimir todas as saídas do comando.
`--region`	Usar a região especificada.
`-v`, `--verbose`	Exibir informações detalhadas.

eb abort

Descrição

Cancela uma atualização quando as alterações de configuração do ambiente nas instâncias ainda estão em andamento.

> **Note**
>
> Se você tiver mais de dois ambientes passando por uma atualização, será solicitado a selecionar o nome do ambiente no qual deseja reverter as alterações.

Sintaxe

eb abort

eb abort ***environment-name***

Opções

Nome	Descrição
Opções comuns	

Saída

O comando mostra uma lista de ambientes que estão sendo atualizados e solicita que você escolha a atualização que deseja anular. Se apenas um ambiente está sendo atualizado, você não precisa especificar o nome dele. Se bem-sucedido, o comando reverte as alterações de configuração do ambiente. O processo de reversão continua até que todas as instâncias no ambiente tenham a configuração de ambiente anterior ou até que o processo de reversão falhe.

Exemplo

O seguinte exemplo cancela a atualização da plataforma.

```
$ eb abort
Aborting update to environment "tmp-dev".
```

```
<list of events>
```

eb appversion

Descrição

O comando `appversion` da EB CLI gerencia as versões da aplicação do Elastic Beanstalk. Você pode criar uma nova versão de aplicação sem implantar, excluir uma versão da aplicação ou criar a política de ciclo de vida da versão da aplicação. Se você chamar o comando sem opções, ele entrará no modo interativo.

Use a opção `--create` para criar uma nova versão da aplicação.

Use a opção `--delete` para excluir uma versão do aplicativo.

Use a opção `lifecycle` para exibir ou criar a política de ciclo de vida de versão do aplicativo. Para obter mais informações, consulte the section called “Ciclo de vida da versão”.

Sintaxe

eb appversion

eb appversion [-c | --create]

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

eb appversion lifecycle [-p | --print]

Opções

Nome	Descrição Tipo: String
-uma *application-name* or --nome_do_aplicativo *application-name*	O nome do aplicativo. Se uma aplicação com o nome especific ado não for encontrada, a EB CLI criará uma versão da aplicação para uma nova aplicação. Aplicável somente com a opção `--create`. Tipo: sequência

Nome	Descrição Tipo: String
-c ou --create	Crie uma nova versão da aplicação.
-d *version-label* or --excluir *version-label*	Exclua a versão do aplicativo que está rotulada*version-label* .
`-l` *version_label* or `--label` *version_label*	Especifique um rótulo a ser usado para a versão criada pela EB CLI. Se você não usar essa opção, a EB CLI gerará um novo rótulo exclusivo. Se você fornecer um rótulo de versão, verifique se ele é exclusivo. Aplicável somente com a opção `--create`. Tipo: sequência
ciclo de vida	Chamar o editor padrão para criar uma nova política de ciclo de vida de versão do aplicativo. Use esta política para evitar atingir a cota da versão da aplicação.
lifecycle -p ou lifecycle --print	Exibir a política de ciclo de vida do aplicativo atual.

Nome	Descrição Tipo: String

`-m "`*`version_description`*` "`
 	
or
 	
`--message "`*`version_description`*`  "`
	A descrição da versão da aplicação. Está incluído entre aspas duplas.
	
	Aplicável somente com a opção `--create`.
	
	Tipo: sequência

`-p`
 	
ou
 	
`--process`
	Pré-processa e valida o manifesto de ambiente e os arquivos de configuração no pacote de origem. A validação de arquivos de configuração pode identificar problemas. Recomendamos que você faça isso antes de implantar a versão da aplicação no ambiente.
	
	Aplicável somente com a opção `--create`.

`--source codecommit/ `*`repository-name`*`/`*`branch-name`*	CodeCommit repositório e filial. Aplicável somente com a opção `--create`.
`--staged`	Use os arquivos preparados no índice git em vez da confirmação HEAD para criar a versão da aplicação. Aplicável somente com a opção `--create`.
`--timeout `*`minutes`*	O número de minutos antes de esgotar o tempo limite do comando. Aplicável somente com a opção `--create`.
Opções comuns	

Usar o comando interativamente

Se você usar o comando sem argumentos, a saída exibirá as versões da aplicação. Eles estão listados em ordem cronológica inversa, com a última versão listada primeiro. Consulte a seção Examples (Exemplos) para obter exemplos da aparência da tela. Observe que a linha de status é exibida na parte inferior. A linha de status exibe informações sensíveis ao contexto.

Pressione d para excluir uma versão do aplicativo, pressione `l` para gerenciar a política de ciclo de vida de seu aplicativo ou pressione q para sair sem fazer alterações.

> **Note**
>
> Se a versão estiver implantada em qualquer ambiente, você não poderá excluí-la.

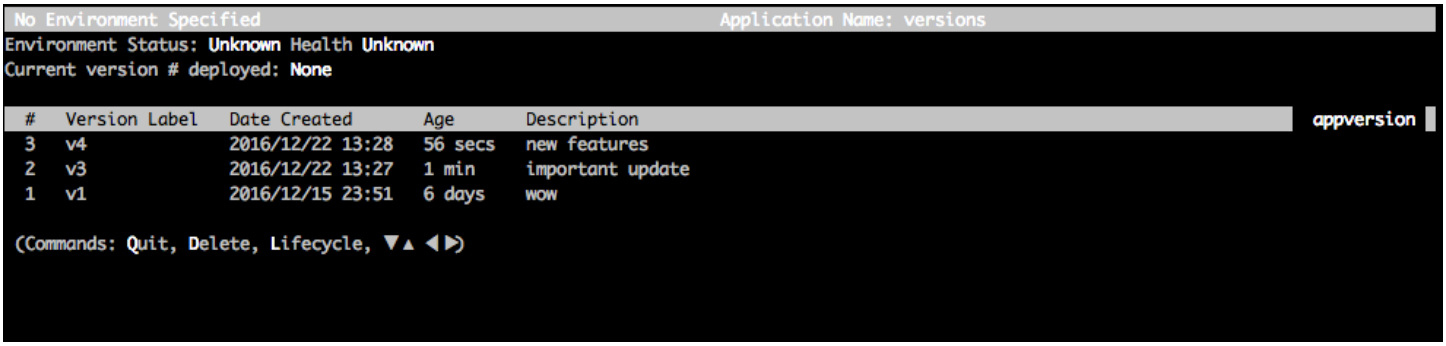
Saída

O comando com a opção `--create` exibe uma mensagem confirmando que a versão da aplicação foi excluída.

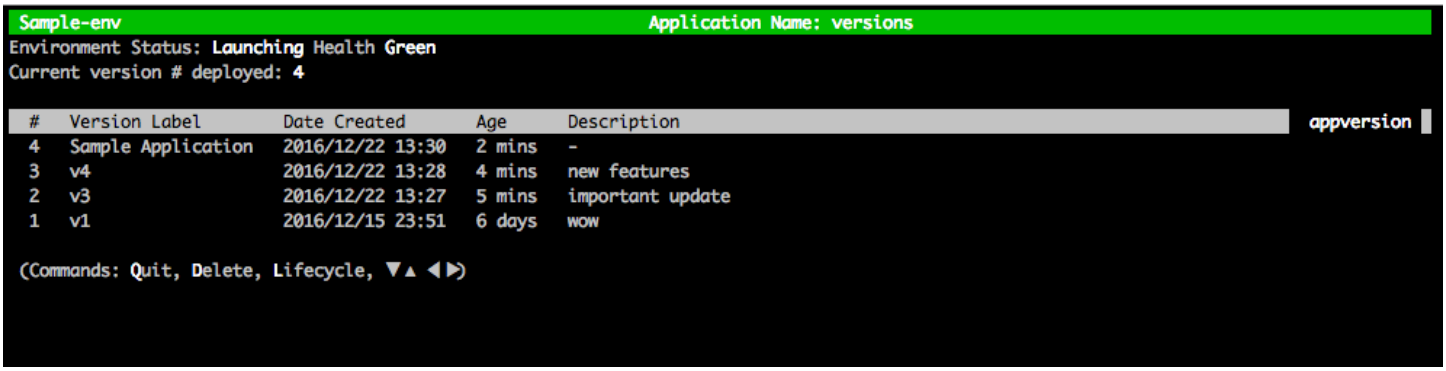
O comando com a `--delete` *`version-label`* opção exibe uma mensagem confirmando que a versão do aplicativo foi excluída.

Exemplos

O exemplo a seguir mostra a janela interativa de um aplicativo sem implantações.



O exemplo a seguir mostra a janela interativa de um aplicativo com a quarta versão, com o rótulo da versão Sample Application (Aplicativo de exemplo), implantada.



O exemplo a seguir mostra a saída de um eb appversion lifecycle -p comando, onde *ACCOUNT-ID* está o ID da conta do usuário:

```
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

Descrição

Clona um ambiente para um novo ambiente de modo que ambos têm configurações de ambiente idênticas.

> **Note**
>
> Por padrão, independentemente da versão da stack de soluções do ambiente do qual você criou o clone, o comando eb clone cria o ambiente clone com a stack de soluções mais recente. Você pode suprimir isso incluindo a opção `--exact` ao executar o comando.

> **Important**
>
> Ambientes clonados do Elastic Beanstalk não transferem os grupos de segurança para entrada, deixando o ambiente aberto para todo o tráfego da Internet. Será necessário restabelecer os grupos de segurança de entrada para o ambiente clonado.
> É possível ver recursos que não podem ser clonados verificando o status de desvio da configuração do ambiente. Para obter mais informações, consulte Detectar deriva em uma CloudFormation pilha inteira no Guia do AWS CloudFormation usuário.

Sintaxe

eb clone

eb clone ***environment-name***

Opções

Nome	Descrição
`-n` *`string`* or `--clone_name` *`string`*	Nome desejado para o ambiente clonado.
`-c` *`string`* or `--cname` *`string`*	Prefixo CNAME desejado para o ambiente clonado.

Nome	Descrição
`--envvars`	Propriedades do ambiente em uma lista separada por vírgulas com o formato *`name`* =. *`value`* Tipo: string Restrições: • Os pares de chave e valor devem ser separados por vírgulas. • As chaves e valores podem conter qualquer caractere alfabético em qualquer idioma, qualquer caractere numérico, espaço em branco, separador invisível e os seguintes símbolos: _ . : / + \ - @ • As chaves podem conter até 128 caracteres. Os valores podem conter até 256 caracteres. • As chaves e os valores diferenciam letras maiúsculas de minúsculas. • Os valores não podem coincidir com o nome do ambiente. • Os valores não podem incluir `aws:` ou `elasticbeanstalk:` . • O tamanho total de todas as propriedades do ambiente não pode exceder 4096 bytes.
`--exact`	Impede que o Elastic Beanstalk atualize a versão da pilha de soluções do novo ambiente clone para a versão mais recente disponível (para a plataforma do ambiente original).
`--scale` *`number`*	O número de instâncias a serem executadas no ambiente clone quando ele é iniciado.
`--tags` *`name`*=*`value`*	Tags para os recursos em seu ambiente em uma lista separada por vírgulas com o formato *`name`* =. *`value`*
`--timeout`	O número de minutos antes de esgotar o tempo limite do comando.

Nome	Descrição
Opções comuns	

Saída

Se tiver êxito, o comando cria um ambiente que tem as mesmas configurações que o ambiente original ou com modificações no ambiente, como especificado por qualquer opção eb clone.

Exemplo

O exemplo a seguir clona o ambiente especificado.

```
$ 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

Descrição

Configura a CLI do EB para ser implantada a partir de CodeCommit um repositório ou CodeCommit desativa a integração e carrega o pacote de origem da sua máquina local.

> **Note**
>
> Algumas AWS regiões não oferecem CodeCommit. A integração entre o Elastic CodeCommit Beanstalk não funciona nessas regiões.
> Para obter informações sobre os AWS serviços oferecidos em cada região, consulte a tabela de regiões.

Sintaxe

eb codesource

eb codesource codecommit

eb codesource local

Opções

Nome	Descrição
Opções comuns	

Saída

eb codesourcesolicita que você escolha entre CodeCommit integração e implantações padrão.

eb codesource codecommit inicia a configuração do repositório interativo para integração do CodeCommit .

eb codesource local mostra a configuração original e desabilita a integração do CodeCommit.

Exemplos

Use eb codesource codecommit para configurar a CodeCommit integração para a ramificação atual.

```
~/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
```

Use eb codesource local para desativar a CodeCommit integração da ramificação atual.

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

eb config

Descrição

Gerencia as configurações ativas e as configurações salvas do seu ambiente. Você pode usar esse comando para fazer upload, fazer download ou listar as configurações salvas do seu ambiente. Você também pode usá-lo para fazer download, exibir ou atualizar suas configurações ativas.

Se o diretório raiz contém um arquivo `platform.yaml` especificando uma plataforma personalizada, este comando também altera as configurações do construtor. Isso é feito com base nos valores definidos em `platform.yaml`.

> **Note**
>
> eb config não exibe as propriedades do ambiente. Para definir as propriedades do ambiente que você pode ler a partir da sua aplicação, use eb setenv.

Sintaxe

A seguir estão partes da sintaxe que é usada para que o comando eb config funcione com as configurações ativas do seu ambiente. Para obter exemplo específicos, consulte a seção Exemplos mais adiante neste tópico.

- eb config: exibe as configurações ativas do seu ambiente em um editor de texto que você configurou como a variável de ambiente EDITOR. Quando você salvar as alterações no arquivo e fechar o editor, o ambiente será atualizado com as configurações da opção no arquivo.

  > **Note**
  >
  > Se você não configurou uma variável de ambiente EDITOR, a EB CLI exibirá suas configurações de opção no editor padrão para arquivos YAML.

- eb config ***environment-name***: exibe e atualiza a configuração para o ambiente nomeado. A configuração é exibida em um editor de texto que você configurou ou em seus arquivos YAML do editor padrão.
- eb config save: salva as configurações ativas para o ambiente atual `.elasticbeanstalk/saved_configs/` com o nome do arquivo`[configuration-name].cfg.yml`. Por padrão, a CLI do EB salva as configurações com *`configuration-name`* base no nome do ambiente. Você pode especificar um nome de configuração diferente incluindo a opção `--cfg` com o nome de configuração desejado quando executar o comando.

  Você pode marcar a configuração salva usando a opção `--tags`.
- eb config **`--display`**: grava as configurações ativas de um ambiente em stdout em vez de um arquivo. Por padrão, isso exibe as configurações para o final.
- eb config **`--update`** ***`configuration_string`*** **`|`** ***`file_path`***— atualiza as configurações ativas do ambiente atual com as informações *`configuration_string`* especificadas no arquivo identificado por*`file_path`*.

> **Note**
>
> As opções `--display` e `--update` oferecem flexibilidade para ler e revisar as configurações de um ambiente programaticamente.

A tabela a seguir descreve a sintaxe para usar o comando eb config para funcionar com configurações salvas. Para obter exemplos, consulte a seção Exemplos mais adiante neste tópico.

- eb config get ***config-name***: faz o download da configuração salva nomeada do Amazon S3.
- eb config delete ***config-name*** : exclui a configuração salva nomeada do Amazon S3. Também o exclui localmente, se você já fez o download.
- eb config list: lista as configurações salvas que você tem no Amazon S3.
- eb config put ***filename***: faz upload da configuração salva nomeada em um bucket do Amazon S3. Eles *filename* devem ter a extensão do arquivo `.cfg.yml`. Para especificar o nome de arquivo sem um caminho, você pode salvar o arquivo na pasta `.elasticbeanstalk` ou `.elasticbeanstalk/saved_configs/` antes de executar o comando. Como alternativa, você pode especificar o *filename* fornecendo o caminho completo.

Opções

Nome	Descrição
`--cfg` *config-name*	O nome a ser usado para uma configuração salva. Essa opção funciona apenas com eb config save.

`-d`
 	
ou
 	
`--display`
	Exibe as configurações do ambiente atual (gravações em stdout).
	
	Use com a opção `--format` para especificar a saída para estar em JSON ou YAML. Se você não especificar, a saída estará no formato YAML.
	
	Essa opção só funciona se você usar o comando eb config sem nenhum dos outros subcomandos.

`-f` *format_type* ou `--format` *format_type*	Especifica o formato de exibição. Os valores válidos são JSON ou YAML. O padrão é YAML. Essa opção funciona somente com a opção `--display` .

Nome	Descrição
`--tags key1=value1[,ke`	As tags a serem adicionadas à sua configuração salva. Ao especificar tags na lista, especifique-as como pares chave=valor e separe cada uma com uma vírgula. Para obter mais informações, consulte Marcação de configurações salvas. Essa opção funciona apenas com eb config save.
`--timeout timeout`	O número de minutos antes de esgotar o tempo limite do comando.

Nome	Descrição

`-u configuration_string | file_path`
 	
ou
 	
`--update configuration_string | file_path`
	Atualiza as configurações ativas para o ambiente atual.
	
	Essa opção só funciona se você usar o comando eb config sem nenhum dos outros subcomandos.
	
	O parâmetro *configuration_string | file_path* é do tipo de string. A string fornece a lista de namespaces e opções correspondentes para adicionar, atualizar ou remover das configurações do seu ambiente. Como alternativa, a string de entrada pode representar um arquivo que contém as mesmas informações.
	
	Para especificar um nome de arquivo, a string de entrada deve seguir o formato "file://< *path*><*filename*>". Para especificar o nome do arquivo sem um *path*, salve o arquivo na pasta em que você executa o comando. Como alternativa, você pode especificar o nome de arquivo fornecendo o caminho completo.
	
	As informações de configuração devem atender às condições a seguir. Pelo menos uma das seções,OptionSettings ou OptionsToRemove, é necessária. Use OptionSettings para adicionar ou alterar opções. Use OptionsToRemove para remover opções de um namespace. Para obter exemplo específicos, consulte a seção Exemplos mais adiante neste tópico.
	
	Example
	
	Formato YAML
	
	`OptionSettings:`
	`  namespace1:`
	`    option-name-1:  option-value-1`
	`    option-name-2:  option-value-2`
	`    ...`
	`OptionsToRemove:`
	`  namespace1:`

Nome	Descrição

	option-name-1
	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",
	...
	}
	}
	}

Opções comuns	

Saída

Se o comando eb config ou eb config ***environment-name*** for executado com êxito sem subcomandos ou opções adicionadas, o comando exibirá as configurações de opção atuais no editor de texto que você configurou como a variável de ambiente EDITOR. Se você não configurou uma variável de ambiente EDITOR, a EB CLI exibirá suas configurações de opção no editor padrão para arquivos YAML.

Quando você salvar as alterações no arquivo e fechar o editor, o ambiente será atualizado com as configurações da opção no arquivo. A saída a seguir é exibida para confirmar a atualização de configuração.

```
$ 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 o comando for executado com êxito com a opção `--display`, ele exibirá as configurações do ambiente atual (gravações em stdout).

Se o comando for executado com êxito com o parâmetro `get`, o comando exibirá a localização da cópia local que você baixou.

Se o comando for executado com êxito com o parâmetro `save`, o comando exibirá a localização do arquivo salvo.

Exemplos

Esta seção descreve como alterar o editor de texto usado para visualizar e editar o arquivo de configurações de opção.

Para Linux e UNIX, o seguinte exemplo altera o editor para vim:

```
$ export EDITOR=vim
```

Para Linux e UNIX, o exemplo a seguir altera o editor para aquele que estiver instalado em `/usr/bin/kate`.

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

Para Windows, o exemplo a seguir altera o editor para Notepad++.

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

Esta seção fornece exemplos do comando eb config quando ele é executado com os parâmetros.

O exemplo a seguir exclui a configuração salva chamada `app-tmp`.

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

O exemplo a seguir faz download da configuração salva com o nome app-tmp do bucket do Amazon S3.

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

O exemplo a seguir lista os nomes das configurações salvas que são armazenadas no bucket do Amazon S3.

```
$ eb config list
```

O exemplo a seguir faz upload da cópia local da configuração salva app-tmp no bucket do Amazon S3.

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

O exemplo a seguir salva as configurações do ambiente em execução no momento. Se você não fornecer um nome a ser usado para a configuração salva, o Elastic Beanstalk fornecerá um nome ao arquivo de configuração de acordo com o nome do ambiente. Por exemplo, um ambiente chamado tmp-dev seria chamado `tmp-dev.cfg.yml`. O Elastic Beanstalk salva o arquivo na pasta `/.elasticbeanstalk/saved_configs/`.

```
$ eb config save
```

O exemplo a seguir mostra como usar a opção `--cfg` para salvar as configurações do ambiente tmp-dev em um arquivo chamado `v1-app-tmp.cfg.yml`. O Elastic Beanstalk salva o arquivo na pasta `/.elasticbeanstalk/saved_configs/`. Se você não especificar um nome de ambiente, o Elastic Beanstalk salvará as configurações do ambiente em execução no momento.

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

Esta seção fornece exemplos para o comando eb config quando ele é executado sem subcomandos.

O comando a seguir exibe as configurações de opção do ambiente atual em um editor de texto.

```
$ eb config
```

O comando a seguir exibe as configurações de opção para o ambiente my-env em um editor de texto.

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

O exemplo a seguir exibe as configurações de opções para o ambiente atual. Ele produz no formato YAML porque nenhum formato específico foi especificado com a opção `--format`.

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

O exemplo a seguir atualiza as configurações de opções para seu ambiente atual com as especificações no arquivo nomeado `example.txt`. O arquivo está no formato YAML ou JSON. A EB CLI detecta automaticamente o formato do arquivo.

- A opção Minsize é definida como 1 para o namespace `aws:autoscaling:asg`.
- O tamanho do lote para o namespace `aws:elasticbeanstalk:command` é definido como 30%.
- Ele remove a configuração da opção IdleTimeout: Nenhuma do namespace`AWSEBV2LoadBalancer.aws:elbv2:loadbalancer`.

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

Example - nome do arquivo:**`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 do arquivo:**`example.txt`** - formato JSON

```
{
    "OptionSettings": {
        "aws:elasticbeanstalk:command": {
```

```
            "BatchSize": "30",
            "BatchSizeType": "Percentage"
        },
        "aws:autoscaling:asg": {
            "MinSize": "1"
        }
    },
    "OptionsToRemove": {
        "AWSEBV2LoadBalancer.aws:elbv2:loadbalancer": {
            "IdleTimeout"
        }
    }
}
```

Os exemplos a seguir atualizam as configurações de opções para seu ambiente atual. O comando define a opção Minsize como 1 para o namespace `aws:autoscaling:asg`.

> **Note**
>
> Esses exemplos são específicos para o Windows PowerShell. Eles escapam das ocorrências literais do caractere de aspas duplas ('), precedendo-o com um caractere barra (\). Diferentes sistemas operacionais e ambientes de linha de comando podem ter sequências de escape diferentes. Por esse motivo, recomendamos usar a opção de arquivo mostrada nos exemplos anteriores. Especificar as opções de configuração em um arquivo não requer caracteres de escape e é consistente em diferentes sistemas operacionais.

O exemplo a seguir está no formato JSON. A EB CLI detecta se o formato está em JSON ou YAML.

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

O exemplo a seguir está no formato YAML. Para inserir a string YAML no formato correto, o comando inclui espaçamento e end-of-line retornos necessários em um arquivo YAML.

- Termine cada linha com a chave “enter” ou “return”.
- Inicie a segunda linha com dois espaços e inicie a terceira linha com quatro espaços.

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

```
>>  aws:autoscaling:asg:
>>    MinSize: \"1\"'
```

eb console

Descrição

Abre um navegador para exibir o painel de configuração do ambiente no console de gerenciamento do Elastic Beanstalk.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também exibirá a configuração do ambiente do compilador, conforme especificado no `platform.yaml`, no console de gerenciamento do Elastic Beanstalk.

Sintaxe

eb console

eb console ***environment-name***

Opções

Nome	Descrição
Opções comuns	

eb create

Descrição

Cria um novo ambiente e implanta uma versão do aplicativo nele.

> **Note**
>
> - Para usar o eb create em uma aplicação .NET, é necessário criar um pacote de implantação conforme descrito em Criar pacote de origem para um aplicativo .NET e definir a configuração da CLI para implantar o pacote como um artefato conforme descrito em Implantar um artefato, em vez da pasta do projeto.

- A criação de ambientes com a CLI do EB exige uma função de serviço. É possível criar uma função de serviço criando um ambiente no console do Elastic Beanstalk. Se você não tiver uma função de serviço, a CLI do EB tentará criar uma ao executar `eb create`.

Você pode implantar a versão do aplicativo de algumas origens:

- Por padrão: a partir do código-fonte da aplicação no diretório local do projeto.
- Usando a opção `--version`: a partir de uma versão da aplicação que já existe em sua aplicação.
- Quando o diretório do projeto não tem código de aplicação, ou ao usar a opção `--sample`: a partir de uma aplicação de amostra específica para a plataforma do seu ambiente.

Sintaxe

eb create

eb create ***environment-name***

O nome de um ambiente deve ter entre 4 e 40 caracteres de comprimento. Ele pode conter somente letras, números e hifens (-). Um nome de ambiente não pode começar nem terminar com um hífen.

Se você incluir um nome de ambiente no comando, a EB CLI não pedirá que você faça seleções ou crie uma função de serviço.

Se você executar o comando sem um argumento de nome de ambiente, ele será executado em um fluxo interativo e solicitará que você insira ou selecione valores para algumas configurações. Neste fluxo interativo, caso você esteja implementando um aplicativo de amostra, a EB CLI também pergunta se você deseja fazer download desse aplicativo de amostra no seu diretório local do projeto. Ao fazer download, você pode usar a EB CLI mais tarde com o novo ambiente para executar as operações que exigem o código da aplicação, como eb deploy.

Alguns prompts de fluxo interativo são exibidos somente sob determinadas condições. Por exemplo, se você optar por usar um Application Load Balancer e sua conta tiver pelo menos um Application Load Balancer compartilhável, o Elastic Beanstalk exibirá um prompt perguntando se você deseja usar um balanceador de carga compartilhado. Esse prompt não será exibido se não houver nenhum balanceador de carga da aplicação compartilhável em sua conta.

Opções

Nenhuma dessas opções é obrigatória. Se você executar eb create sem nenhuma opção, a EB CLI solicita que você insira ou selecione um valor para cada configuração.

Nome	Descrição

`-d`
 	
ou
 	
`--branch_default`
	Definir o ambiente como o ambiente padrão para o repositório atual.

`--cfg` *`config-name`*	Use as configurações de plataforma de uma configuração salva em `.elasticbeanstalk/saved_configs/` ou no bucket do Amazon S3. Especifique o nome do arquivo apenas, sem a extensão `.cfg.yml`.
`-c` *`subdomain-name`* or `--cname` *`subdomain-name`*	O nome do subdomínio para prefixar a entrada de CNAME DNS que é roteada para seu website. Tipo: sequência Padrão: o nome do ambiente

`-db`
 	
ou
 	
`--database`
	Anexa um banco de dados ao ambiente. Se você executar eb create com a opção `--database` , mas sem as opções `--database.username` e `--database.password` , a EB CLI solicitará o nome de usuário principal e a senha do banco de dados.

`-db.engine` *`engine`* or `--database.engine` *`engine`*	O tipo de mecanismo de banco de dados. Se você executar eb create com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção `--database` . Tipo: sequência

Nome	Descrição
	Valores válidos: `mysql`, `oracle-se1` , `postgres`, `sqlserver-ex` , `sqlserver-web` , `sqlserver-se`
`-db.i` *`instance_type`* or `--database.instance` *`instance_type`*	O tipo de EC2 instância da Amazon a ser usada para o banco de dados. Se você executar eb create com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção `--databas e` . Tipo: string Valores válidos: O Amazon RDS é compatível com um conjunto padrão de instâncias de banco de dados. Para selecionar uma instância de banco de dados apropriada para seu mecanismo de banco de dados, você deve levar em consideração algumas considerações específicas. Para obter mais informações, consulte Classes de instâncias de banco de dados no Guia do usuário do Amazon RDS.
`-db.pass` *`password`* or `--database.password` *`password`*	A senha do banco de dados. Se você executar eb create com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção `--database` .

Nome	Descrição
`-db.size` *`number_of_gigabytes`* or `--database.size` *`number_of_gigabytes`*	O número de gigabytes (GB) para alocar ao armazenamento de banco de dados. Se você executar eb create com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado. Esse será o caso mesmo que você não tenha executado o comando com a opção `--database` . Tipo: número Valores válidos: • MySQL: `5` a `1024`. O padrão é `5`. • Postgres: `5` a `1024`. O padrão é `5`. • Oracle: `10` a `1024`. O padrão é `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`*	O nome de usuário para o banco de dados. Se você executar eb create com essa opção, a EB CLI iniciará o ambiente com um banco de dados anexado, mesmo que você não tenha executado o comando com a opção `--database` . Se você executar eb create com a opção `--database` , mas sem as opções `--database.username` e `--database.password` , a EB CLI solicitará o nome de usuário e a senha principais do banco de dados.
`-db.version` *`version`* or `--database.version` *`version`*	Usado para especificar a versão do mecanismo de banco de dados. Se esse sinalizador estiver presente, o ambiente será iniciado com um banco de dados com o número de versão especificado, mesmo que o sinalizador `--database` não esteja presente.

Nome	Descrição
`--elb-type` *type*	Novo tipo de load balancer. Tipo: sequência Valores válidos: `classic`, `application` , `network` Padrão: `application`

`-es`
 	
ou
 	
`--enable-spot`
	Habilite solicitações de instância spot para seu ambiente. Para obter mais informações, consulte Auto Scaling group (Grupo do Auto Scaling).
	
	Opções relacionadas:
	
	• `--instance-types`
	• `--on-demand-base-capacity`
	• `--on-demand-above-base-capacity`
	• `--spot-max-price`

`--env-group-suffix` *groupname*	Nome do grupo para anexar ao nome do ambiente. Somente para uso com ambientes do Compose.
`--envvars`	Propriedades do ambiente em uma lista separada por vírgulas com o formato *name* =. *value* Para obter os limites, consulte Configurar propriedades do ambiente (variáveis de ambiente).
`-ip` *profile_name* or `--instance_profile` *profile_name*	O perfil da instância com a função do IAM com as credenciais de segurança temporárias que seu aplicativo precisa para acessar AWS os recursos.

Nome	Descrição

`-it`
 	
or
 	
`--instance-types` *`type1`*`[,`*`type2`* `...]`
	Uma lista separada por vírgulas dos tipos de EC2 instância da Amazon que você deseja que seu ambiente use. Se você não especificar essa opção, o Elastic Beanstalk fornecerá tipos de instância padrão.
	
	Para obter mais informações, consulte EC2 Instâncias da Amazon e Auto Scaling group (Grupo do Auto Scaling).
	
	Important
	A EB CLI aplica somente essa opção a instâncias spot. A menos que esta opção seja usada com a opção `--enable-spot` , a EB CLI a ignorará. Para especificar um tipo de instância para uma instância sob demanda, use a opção `--intance-type` (sem "s").

`-i`
 	
or
 	
`--instance_type`
	O tipo de EC2 instância da Amazon que você deseja que seu ambiente use. Se você não especificar essa opção, o Elastic Beanstalk fornecerá um tipo de instância padrão.
	
	Para obter mais informações, consulte EC2 Instâncias da Amazon.
	
	Important
	A EB CLI aplica essa opção somente a instâncias sob demanda. Não use essa opção com a opção `--enable-spot` , pois a EB CLI a ignora quando você faz isso. Para especificar tipos de instância para uma instância spot, use a opção `--intance-types` (sem "s").

Nome	Descrição
`-k` *key_name* or `--keyname` *key_name*	O nome do par de EC2 chaves da Amazon a ser usado com o cliente Secure Shell (SSH) para fazer login com segurança nas EC2 instâncias da Amazon que estão executando seu aplicativo Elastic Beanstalk. Se você incluir essa opção com o comando eb create, o valor fornecido substituirá qualquer nome de chave especific ado com eb init. Valores válidos: um nome de chave existente registrado na Amazon EC2
`-im` *number-of-instances* or `--min-instances` *number-of-instances*	O número mínimo de EC2 instâncias da Amazon que você precisa que seu ambiente tenha. Tipo: número (inteiro) Padrão: 1 Valores válidos: 1 a `10000`
`-ix` *number-of-instances* or `--max-instances` *number-of-instances*	O número máximo de EC2 instâncias da Amazon que você permite que seu ambiente tenha. Tipo: número (inteiro) Padrão: `4` Valores válidos: 1 a `10000`
`--modules` *component-a component-b*	Uma lista de ambientes de componente para criar. Somente para uso com ambientes do Compose.

Nome	Descrição

`-sb`
 	
ou
 	
`--on-demand-base-capacity`
	O número mínimo de instâncias sob demanda que o grupo de Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente é dimension ado.
	
	Esta opção só pode ser especificada com a opção `--enable-spot` . Para obter mais informações, consulte Auto Scaling group (Grupo do Auto Scaling).
	
	Tipo: número (inteiro)
	
	Padrão: `0`
	
	Valores válidos: `0` para `--max-instances` (quando ausente: opção `MaxSize` no namespace `aws:autoscaling:asg`)

`-sp`
 	
ou
 	
`--on-demand-above-base-capacity`
	A porcentagem de instâncias sob demanda como parte da capacidade adicional que o grupo de Auto Scaling provisiona além do número de instâncias especificado pela opção `--on-demand-base-capacity` .
	
	Esta opção só pode ser especificada com a opção `--enable-spot` . Para obter mais detalhes, consulte Auto Scaling group (Grupo do Auto Scaling).
	
	Tipo: número (inteiro)
	
	Padrão: `0` para um ambiente de uma única instância; `70` para um ambiente com balanceamento de carga
	
	Valores válidos: `0` a `100`

Nome	Descrição
`-p` *platform-version* or `--platform` *platform-version*	A versão da plataforma a ser usada. É possível especificar uma plataforma, uma plataforma e a versão, a ramificação de uma plataforma, o nome de uma pilha de solução ou o ARN de uma pilha de solução. Por exemplo: • php, PHP, `node.js`: a versão mais recente da plataforma especificada • `php-7.2`, `"PHP 7.2"`: a versão recomendada da plataforma PHP 7.2 (geralmente a mais recente) • `"PHP 7.2 running on 64bit Amazon Linux"`: a versão da plataforma PHP especificada por esse nome de pilha da solução • `"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"`: a versão da plataforma PHP especificada por esse nome de pilha da solução • `"arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"` : a versão da plataforma PHP especificada por esse ARN de pilha da solução Use `eb platform list` para obter uma lista de configurações disponíveis. Se você especificar a opção `--platform` , ela substituirá o valor que foi fornecido `eb init`.

`-pr`
 	
ou
 	
`--process`
	Pré-processa e valida o manifesto de ambiente e os arquivos de configuração no pacote de origem. A validação dos arquivos de configuração pode identificar problemas antes de implantar a versão do aplicativo em um ambiente.

Nome	Descrição
`-r` *region* or `--region` *region*	A AWS região em que você deseja implantar o aplicativo. Para obter a lista de valores que você pode especificar para essa opção, consulte Endpoints e cotas do AWS Elastic Beanstalk no Referência geral da AWS.
`--sample`	Implante o aplicativo de exemplo no novo ambiente, em vez do código no seu repositório.
`--scale` *number-of-instances*	Iniciar com o número especificado de instâncias
`--service-role` *servicerole*	Atribua ao ambiente uma função de serviço diferente do padrão. **Note** Não insira um ARN. Insira apenas o nome da função. O Elastic Beanstalk insere um prefixo no nome da função com os valores corretos para criar internamente o ARN resultante.

Nome	Descrição
`-ls` *load-balancer* or `--shared-lb` *load-balancer*	Configure o ambiente para usar um load balancer compartilhado. Forneça o nome ou o ARN de um balanceador de carga compartilhável em sua conta: um Application Load Balancer criado explicitamente, não um criado por outro ambiente do Elastic Beanstalk. Para obter mais informações, consulte Application Load Balancer compartilhado. Exemplos de parâmetros: • `FrontEndLB` : o nome de um balanceador de carga. • `arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/app/FrontEndLB/0dbf78d8ad96abbc` : ARN de um Application Load Balancer. Você pode especificar essa opção somente com `--elb-type application` . Se você especificar essa opção e não especificar `--shared-lb` , o Elastic Beanstalk criará um balanceador de carga dedicado para o ambiente.
`-lp` *port* or `--shared-lb-port` *port*	A porta de listener padrão do balanceador de carga compartilhado para este ambiente. O Elastic Beanstalk adiciona uma regra de listener que encaminha todo o tráfego desse listener para o processo de ambiente padrão. Para obter mais informações, consulte Application Load Balancer compartilhado. Tipo: número (inteiro) Padrão: `80` Valores válidos: qualquer número inteiro que represente uma porta de listener do load balancer compartilhado.

Nome	Descrição
`--single`	Crie o ambiente com uma única EC2 instância da Amazon e sem um balanceador de carga. **Warning** Um ambiente de única instância não está pronto para produção. Se a instância tornar-se instável durante a implantação, ou o Elastic Beanstalk encerrar e reiniciar a instância durante uma atualização de configuração, sua aplicação poderá ficar indisponível durante um determina do período. Use ambientes de única instância para desenvolvimento, teste ou preparação. Use ambientes com balanceamento de carga para produção.

`-sm`
 	
ou
 	
`--spot-max-price`
	O preço máximo por hora, em USD, que você está disposto a pagar por uma instância spot.
	
	Esta opção só pode ser especificada com a opção `--enable-spot` . Para obter mais detalhes, consulte Auto Scaling group (Grupo do Auto Scaling).
	
	Tipo: número (flutuante)
	
	Padrão: Preço sob demanda, para cada tipo de instância. O valor da opção neste caso é `null`.
	
	Valores válidos: `0.001` a `20.0`
	
	Para recomendações sobre as opções de preço máximo para instâncias spot, consulte o histórico de preços de instâncias spot no Guia EC2 do usuário da Amazon.

Nome	Descrição
`--tags` *key1=value1[,key2=va*	Marque os recursos no seu ambiente. As tags são especificadas como uma lista de `key=value` separados por vírgulas. Para obter mais informações, consulte Marcação em ambientes.

`-t worker`
 	
ou
 	
`--tier worker`
	Crie um ambiente de operador. Omita essa opção para criar um ambiente de servidor web.

`--timeout` *minutes*	O número definido de minutos antes do limite de tempo do comando.
`--version` *version_label*	Especifica a versão do aplicativo que você deseja implantar no ambiente, em vez do código-fonte do aplicativo no diretório do projeto local. Tipo: sequência Valores válidos: um rótulo de versão do aplicativo existente
`--vpc`	Configure uma VPC em seu ambiente. Quando você incluir essa opção, a EB CLI solicitará que você insira todas as configurações obrigatórias antes de iniciar o ambiente.
`--vpc.dbsubnets` *subnet1,subnet2*	Especifica sub-redes para instâncias de banco de dados em uma VPC. Obrigatório quando `--vpc.id` for especificado.
`--vpc.ec2subnets` *subnet1,subnet2*	Especifica sub-redes para EC2 instâncias da Amazon em uma VPC. Obrigatório quando `--vpc.id` for especificado.

Nome	Descrição
`--vpc.elbpublic`	Inicia o balanceador de carga do Elastic Load Balancing em uma sub-rede pública em sua VPC. Não é possível especificar essa opção com `--tier worker` e `--single`.
`--vpc.elbsubnets` *subnet1,subnet2*	Especifica sub-redes para o balanceador de carga do Elastic Load Balancing em uma VPC. Não é possível especificar essa opção com `--tier worker` e `--single`.
`--vpc.id` *ID*	Inicia seu ambiente na VPC especificada.
`--vpc.publicip`	Lança suas EC2 instâncias da Amazon em uma sub-rede pública em sua VPC. Não é possível especificar essa opção com `--tier worker` .
`--vpc.securitygroups` *securitygroup1,securitygroup2*	Especifica o grupo IDs de segurança. Obrigatório quando `--vpc.id` for especificado.
Opções comuns	

Saída

Se for bem-sucedido, o comando apresentará prompts com perguntas e retornará o status da operação de criação. Se houver problemas durante a inicialização, você poderá usar a operação eb events para obter mais detalhes.

Se você ativou o CodeBuild suporte em seu aplicativo, eb create exibirá informações de CodeBuild como seu código é criado. Para obter informações sobre o CodeBuild suporte no Elastic Beanstalk, consulte. Usando o EB CLI com AWS CodeBuild

Exemplos

O exemplo a seguir cria um ambiente no modo interativo.

```
$ 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:
...
```

O exemplo a seguir também cria um ambiente no modo interativo. Neste exemplo, o diretório do projeto não tem código de aplicativo. O comando implanta um aplicativo de amostra e faz o download deste no diretório local do projeto.

```
$ 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:
...
```

O seguinte comando cria um ambiente sem exibir prompts.

```
$ 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:
...
```

O seguinte comando cria um ambiente em uma VPC personalizada.

```
$ 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

Descrição

Implanta o pacote de origem do aplicativo, que se encontra no diretório do projeto inicializado, no aplicativo em execução.

Se o git for instalado, a EB CLI usará o comando `git archive` para criar um arquivo `.zip` do conteúdo do comando `git commit` mais recente.

No entanto, quando `.ebignore` está presente no diretório do projeto, a CLI do EB não usa semântica e comandos git para criar o pacote de origem. Isso significa que a EB CLI ignora os arquivos especificados em `.ebignore` e inclui todos os outros. Especificamente, ela inclui os arquivos de origem não confirmados.

> **Note**
>
> Você pode configurar a EB CLI para implantar um artefato do seu processo de compilação, em vez de criar um arquivo ZIP da pasta do projeto. Para mais detalhes, consulte Implantar um artefato, em vez da pasta do projeto.

Sintaxe

eb deploy

eb deploy ***environment-name***

Opções

Nome	Descrição
`-l` *version_label* or `--label` *version_label*	Especifique um rótulo a ser usado para a versão criada pela EB CLI. Se o rótulo já tiver sido usado, a EB CLI reimplanta a versão anterior com esse rótulo. Tipo: sequência
`--env-group-suffix` *groupname*	Nome do grupo para anexar ao nome do ambiente. Somente para uso com ambientes do Compose.

`-m "`*version_description*` "`
 	
or
 	
`--message "`*version_description*` "`
	A descrição da versão do aplicativo, entre aspas duplas.
	
	Tipo: String

`--modules` *component-a component-b*	Lista de componentes para atualizar. Somente para uso com ambientes do Compose.

`-p`
 	
ou
 	
`--process`
	Pré-processa e valida o manifesto de ambiente e os arquivos de configuração no pacote de origem. A validação dos arquivos de configuração pode identificar problemas antes de implantar a versão do aplicativo em um ambiente.

`--source codecommit/` *repository-name*`/`*branch-name*	CodeCommit repositório e filial.
`--staged`	Implanta arquivos preparados no índice GIT, em vez da confirmação HEAD.
`--timeout` *minutes*	O número de minutos antes de esgotar o tempo limite do comando.

Nome	Descrição
`--version` *version_label*	Uma versão do aplicativo existente para implantar. Tipo: sequência
Opções comuns	

Saída

Se for bem-sucedido, o comando retornará o status da operação `deploy`.

Se você ativou o CodeBuild suporte em seu aplicativo, eb deploy exibirá informações de CodeBuild como seu código é criado. Para obter informações sobre CodeBuild suporte no Elastic Beanstalk, consulte. Usando o EB CLI com AWS CodeBuild

Exemplo

O exemplo a seguir implanta o aplicativo atual.

```
$ 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

Descrição

Retorna os eventos mais recentes do ambiente.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também retornará os eventos mais recentes para o ambiente do criador.

Sintaxe

eb events

eb events ***environment-name***

Opções

Nome	Descrição

`-f`
 	
ou
 	
`--follow`
	Transmite eventos por stream. Para cancelar, pressione CTRL +C.

Saída

Se for bem-sucedido, o comando retornará os eventos recentes.

Exemplo

O exemplo a seguir retorna os eventos mais recentes.

```
$ 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

Descrição

Retorna a integridade mais recente do ambiente.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também retornará a integridade mais recente do ambiente do compilador.

Sintaxe

eb health

eb health ***environment-name***

Opções

Nome	Descrição

`-r`
 	
ou
 	
`--refresh`
	Mostra informações de integridade interativamente e atualiza a cada dez segundos à medida que novas informações são relatadas.

`--mono`	Não exibe cor na saída.

Saída

Se for bem-sucedido, o comando retornará a integridade recente.

Exemplo

O exemplo a seguir retorna as informações de integridade mais recentes para um 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

Descrição

Define valores padrão para aplicações do Elastic Beanstalk criados com a CLI do EB pedindo que você responda a uma série de perguntas.

> **Note**
>
> Os valores definidos com eb init se aplicam ao diretório e ao repositório atuais no computador atual.
> O comando cria um aplicativo do Elastic Beanstalk em sua conta. Para criar um ambiente do Elastic Beanstalk, execute eb create após a execução de eb init.

Sintaxe

eb init

eb init *application-name*

Opções

Se você executar eb init sem especificar a opção `--platform`, a CLI do EB solicita que você insira um valor para cada configuração.

> **Note**
>
> Para usar eb init para criar um novo par de chaves, instale `ssh-keygen` em sua máquina local e disponibilize-o na linha de comando.

Nome	Descrição	

`-i`
 	
`--interactive`
	Força a EB CLI a pedir que você forneça um valor para cada opção de comando eb init.
	
	> **Note**
	> O comando `init` pede que você forneça valores para as opções do comando eb init que não têm um valor (padrão). Após a primeira execução do comando eb init em um diretório, a EB CLI pode não perguntar sobre nenhuma opção de comando. Portanto, use a opção `--interactive` quando desejar alterar uma configuração previamente definida.	

`-k` *`keyname`* `--keyname` *`keyname`*	O nome do par de EC2 chaves da Amazon a ser usado com o cliente Secure Shell (SSH) para fazer login com segurança nas EC2 instâncias da Amazon que executam seu aplicativo Elastic Beanstalk.	
`--modules` *`folder-1 folder-2`*	Lista de diretórios filho para inicializar. Somente para uso com ambientes do Compose.	
`-p` *`platform-version`*	A versão da plataforma a ser usada. É possível especificar uma plataforma, uma plataforma e a	

Nome	Descrição	
`--platform` *`platform-version`*	versão, a ramificação de uma plataforma, o nome de uma pilha de solução ou o ARN de uma pilha de solução. Por exemplo: • `php`, `PHP`, `node.js`: a versão mais recente da plataforma especificada • `php-7.2`, `"PHP 7.2"`: a versão recomendada da plataforma PHP 7.2 (geralmente a mais recente) • `"PHP 7.2 running on 64bit Amazon Linux"`: a versão da plataforma PHP especificada por esse nome de pilha da solução • `"64bit Amazon Linux 2017.09 v2.6.3 running PHP 7.1"`: a versão da plataforma PHP especificada por esse nome de pilha da solução • `"arn:aws:elasticbeanstalk:us-east-2::platform/PHP 7.1 running on 64bit Amazon Linux/2.6.3"` : a versão da plataforma PHP especificada por esse ARN de pilha da solução Use `eb platform list` para obter uma lista de configurações disponíveis. Especifique a opção `--platform` para ignorar a configuração interativa. **Note** Quando você especificar essa opção, a EB CLI não solicitará valores para as demais opções. Em vez disso, ele usa valores padrão para cada opção. É possível definir todas as opções para as quais não deseja usar valores padrão.	

Nome	Descrição	
`--source codecommit/ repository-name/branch-name`	CodeCommit repositório e filial.	
`--tags key1=value1`	Marque seu aplicativo. As tags são especificadas como uma lista de `key=value` separados por vírgulas. Consulte mais detalhes em Marcar aplicativos.	
Opções comuns		

CodeBuild apoio

Se você executar eb init em uma pasta com um arquivo buildspec.yml o Elastic Beanstalk analisará o arquivo para uma entrada eb_codebuild_settings com opções específicas do Elastic Beanstalk. Para obter informações sobre o CodeBuild suporte no Elastic Beanstalk, consulte. Usando o EB CLI com AWS CodeBuild

Saída

Se for bem-sucedido, o comando orientará você na configuração de uma nova aplicação do Elastic Beanstalk por meio de uma série de prompts.

Exemplo

A solicitação de exemplo a seguir inicializa a EB CLI e pede que você insira informações sobre o aplicativo. Substitua o *placeholder* texto por seus próprios valores.

```
$ 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

Descrição

Subcomandos de eb labs suporte work-in-progress ou funcionalidade experimental. Esses comandos podem ser removidos ou retrabalhados em futuras versões da EB CLI, sem garantia de compatibilidade.

Para obter uma lista dos subcomandos e descrições disponíveis, execute eb labs --help.

eb list

Descrição

Lista todos os ambientes no aplicativo atual ou todos os ambientes em todos os aplicativos, conforme especificado pela opção `--all`.

Se o diretório raiz contém um arquivo `platform.yaml` especificando uma plataforma personalizada, este comando também lista o ambiente do compilador.

Sintaxe

eb list

Opções

Nome	Descrição

`-a`
ou
`--all`
	Lista todos os ambientes de todos os aplicativos.

`-v`
ou
`--verbose`
	Fornece informações mais detalhadas sobre todos os ambientes , incluindo instâncias.

Opções comuns	

Saída

Se for bem-sucedido, o comando retornará uma lista de nomes de ambiente na qual seu ambiente atual é marcado com um asterisco (*).

Exemplo 1

O exemplo a seguir relaciona os seus ambientes e indica que tmp-dev é o seu ambiente padrão.

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

Exemplo 2

O exemplo a seguir relaciona seus ambientes com detalhes adicionais.

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

eb local

Descrição

Use eb local run para executar os contêineres de seu aplicativo localmente no Docker. Verifique o status do contêiner do aplicativo com eb local status. Abra o aplicativo em um navegador web com eb local open. Recupere a localização de logs do aplicativo com eb local logs.

eb local setenv e eb local printenv permitem que você defina e visualize variáveis de ambiente que são fornecidas aos contêineres do Docker que você executa localmente com eb local run.

É necessário executar todos os comandos eb local no diretório de projeto de um aplicativo do Docker que foi inicializado como um repositório da EB CLI usando eb init.

> **Note**
>
> Use eb local em um computador local executando Linux ou macOS. O comando não oferece suporte ao Windows.
> Antes de usar o comando no macOS, instale o Docker para Mac e verifique se o boot2docker não está instalado (ou se não está no caminho de execução). O comando eb local tenta usar o boot2docker se estiver presente, mas não funciona bem com ele no macOS.

Sintaxe

eb local run

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Opções

eb local run

Nome	Descrição
`--envvars key1=value1,key2=value2`	Define variáveis do ambiente que a EB CLI passará para os contêineres do Docker local. Em ambientes de vários contêineres, todas as variáveis são passadas para todos os contêineres.
`--port hostport`	Mapeia uma porta no host para a porta exposta no contêiner. Se você não especificar essa opção, a EB CLI usa a mesma porta no host e no contêiner. Essa opção funciona somente com aplicativos da plataforma do Docker. Ela não se aplica à plataforma Docker de vários contêineres.
Opções comuns	

eb local status

eb local open

eb local logs

eb local setenv

eb local printenv

Nome	Descrição
Opções comuns	

Saída

eb local run

Mensagens de status do Docker. Permanece ativa enquanto o aplicativo estiver em execução. Pressione Ctrl+C para interromper o aplicativo.

eb local status

O status de cada contêiner usado pelo aplicativo, em execução ou não.

eb local open

Abre o aplicativo em um navegador web e sai.

eb local logs

A localização dos logs gerados no diretório do projeto por aplicativos em execução localmente em eb local run.

eb local setenv

Nenhum

eb local printenv

O nome e os valores de variáveis de ambiente definidas com eb local setenv.

Exemplos

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

Visualize o status de seus contêineres locais:

```
~/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

Veja o caminho do log do projeto atual:

```
~/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

Defina as variáveis do ambiente que serão utilizadas com eb local run.

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

Imprima as variáveis do ambiente definidas com eb local setenv.

```
~/project$ eb local printenv
```

```
Environment Variables:
PARAM1=value
```

eb logs

Descrição

O eb logs comando tem duas finalidades distintas: ativar ou desativar o streaming de CloudWatch registros para o Logs e recuperar registros de instâncias ou registros de registros de CloudWatch registros. Com a opção `--cloudwatch-logs` (`-cw`), o comando habilita ou desabilita o streaming de logs. Sem essa opção, ele recupera logs.

Ao recuperar logs, especifique a opção `--all`, `--zip` ou `--stream` para recuperar logs completos. Se você não especificar nenhuma dessas opções, o Elastic Beanstalk recuperará logs finais.

O comando processa logs do ambiente especificado ou do ambiente padrão. Os logs relevantes variam por tipo de contêiner. Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também processará logs do ambiente do compilador.

Para obter mais informações, consulte the section called “CloudWatch Registros”.

Sintaxe

Para ativar ou desativar o streaming de CloudWatch registros para o Logs:

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

Para recuperar logs de instâncias:

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

Para recuperar logs de integridade do ambiente:

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

Opções

Nome	Descrição

`-cw [enable | disable]`
 	
or
 	
`--cloudwatch-logs [enable | disable]`
	Ativa ou desativa o streaming de CloudWatch registros para o Logs. Se nenhum argumento for fornecido, o streaming de logs será habilitado. Se a opção `--cloudwatch-log-source` (`-cls`) não for especificada, o streaming de logs de instância será habilitado ou desabilitado.

`-cls instance | environment-health | all`
 	
or
 	
`--cloudwatch-log-source instance | environment-health | all`
	Especifica a origem dos registros ao trabalhar com o CloudWatch Logs. Com a forma de ativação ou desativação do comando, esses são os registros para os quais ativar ou desativar o streaming de CloudWatch registros. Com a forma de recuperação do comando, esses são os registros a serem recuperados do CloudWatch Logs.
	
	Valores válidos:
	
	• Com o `--cloudwatch-logs` (enable ou disable) – `instance | environment-health | all`
	• Sem `--cloudwatch-logs` (retrieve) - `instance | environment-health`
	
	Significados dos valores:
	
	• `instance` (padrão) – logs de instâncias
	• `environment-health` – logs de integridade do ambiente (compatível somente quando a integridade avançada está habilitada no ambiente)
	• `all` - as duas origens do log

`-a`
 	
ou
 	
`--all`
	Recupera logs completos e salva-os no diretório `.elasticbeanstalk/logs`.

Nome	Descrição

`-z`
 	
ou
 	
`--zip`
	Recupera logs completos, compacta-os em um arquivo `.zip` e, em seguida, salva o arquivo no diretório `.elasticbeanstalk/logs` .

`--stream`	Transmite (saídas contínuas) logs completos. Com essa opção, o comando continua a ser executado até que você o interrompa (pressione **`Ctrl+C`**).
`-i` *`instance-id`* ou `--instance` *`instance-id`*	Recupera apenas os logs de uma instância especificada.

Nome	Descrição

`-g log-group`
 	
or
 	
`--log-group log-group`
	Especifica o grupo de CloudWatch registros de registros do qual os registros serão recuperados. A opção é válida somente quando o streaming de registros da instância para CloudWatch Logs está ativado.
	
	Se o streaming de log de instância estiver habilitado, e você não especificar a opção `--log-group` , o grupo de logs padrão será um dos seguintes:
	
	• Amazon Linux 2: `/aws/elasticbeanstalk/ environment-name /var/log/eb-engine.log`
	• Plataformas Windows – `/aws/elasticbeanstalk/ environment-name /EBDeploy-Log`
	• Amazon Linux AMI (AL1) — `/aws/elasticbeanstalk/ environment-name /var/log/eb-activity.log`
	
	Note
	Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.
	
	Para obter informações sobre o grupo de logs correspondente a cada arquivo de log, visite Como o Elastic Beanstalk configura os registros CloudWatch .

Opções comuns	

Saída

Por padrão, exibe os logs diretamente no terminal. Usa um programa de paginação para exibir a saída. Pressione **`Q`** ou **`q`** para sair.

Com `--stream`, mostra os logs existentes no terminal e continua em execução. Pressione **`Ctrl+C`** para sair.

Com `--all` e `--zip`, salva os logs em arquivos locais e exibe o local do arquivo.

Exemplos

O exemplo a seguir permite o streaming de registros de instâncias para CloudWatch Logs.

```
$ 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.
```

O exemplo a seguir recupera os logs de instância em um arquivo `.zip`.

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

eb migrate

Descrição

Migra sites e aplicativos do Internet Information Services (IIS) de um servidor Windows para o Elastic Beanstalk. O comando empacota seus aplicativos, preserva suas configurações e os implanta em um novo ambiente do Elastic Beanstalk.

Para obter mais informações sobre como migrar seus sites e aplicativos do IIS, consulteMigração de aplicativos IIS.

> **Note**
>
> Antes de usar esse comando, verifique se o sistema atende aos seguintes requisitos:
>
> - Serviços de Informações da Internet (IIS) versão 7.0 ou posterior
> - Web Deploy 3.6 ou posterior instalado
> - Privilégios administrativos no servidor Windows
> - AWS credenciais configuradas com as permissões apropriadas
> - Seu servidor de origem tem acesso de saída à Internet aos AWS serviços.

As etapas a seguir resumem o processo de migração:

1. Descubra os sites do IIS e suas configurações.
2. Conteúdo e configuração do aplicativo Package.
3. Crie o ambiente e o aplicativo do Elastic Beanstalk.
4. Implante o aplicativo com configurações preservadas.

O comando cria artefatos de migração em um diretório estruturado, conforme mostrado na lista a seguir:

```
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
```

Use eb migrate cleanup para gerenciar esses artefatos.

Sintaxe

eb migrate ***[options]***

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

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

Quando executado sem argumentos, eb migrate opera no modo não interativo. Para executá-lo no modo interativo, executeeb migrate --interactive.

O comando do modo interativo solicita as seguintes informações:

- Seleção de sites do IIS para migrar
- Nomes do ambiente e do aplicativo
- Seleção da versão da plataforma
- Tipo de instância e outras opções de configuração

Subcomandos

explore

O eb migrate explore subcomando examina seu servidor IIS e lista os sites disponíveis.

Use esse comando para exibir as seguintes informações:

- Exibir todos os sites do IIS no servidor
- Com `--verbose`, inspecione a configuração detalhada, incluindo:
  - Ligações e portas do site
  - Pools de aplicativos
  - Diretórios virtuais e seus caminhos físicos
  - Configurações de Autenticação

```
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:
...
```

Limpeza

O eb migrate cleanup subcomando gerencia os artefatos de migração com as seguintes ações:

- Preservando a migração bem-sucedida mais recente em `./migrations/latest`
- Removendo diretórios de migração mais antigos
- Manutenção de arquivos de configuração críticos

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

Use `--force` para ignorar as solicitações de confirmação durante a limpeza.

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

Opções

Nenhuma dessas opções é obrigatória. Se você executar eb migrate sem nenhuma opção, a CLI do EB será executada no modo não interativo. Comeb migrate --interactive, o EB CLI solicita que você insira ou selecione um valor para as configurações necessárias.

Nome	Descrição
`-a` *`application-name`* or `--application-name` *`application-name`*	Nome do novo aplicativo Elastic Beanstalk. Tipo: String Padrão: EBMigrated App
`--archive` *`directory-or-zip`*	O diretório ou arquivo ZIP contendo o código-fonte gerado anteriormente peloeb migrate --archive-only. Use essa opção para implantar um pacote de migração criado anteriormente. Exemplo: `--archive .\migrations\latest\upload_target` ou `--archive .\migrations\latest\upload_target.zip`

`-ao`
 	
or
 	
`--archive-only`
	Crie somente o diretório de arquivamento de destino sem implantação.
	
	O diretório resultante pode ser implantado manualmente usando eb migrate a `archive` opção oueb deploy.

`-c` *`subdomain-name`* or `--cname` *`subdomain-name`*	O nome do subdomínio para prefixar a entrada DNS CNAME do seu aplicativo migrado. Tipo: String

Nome	Descrição
	Padrão: o nome do ambiente

`-cf`
 	
or
 	
`--copy-firewall-config`
	Copie a configuração do firewall do servidor de origem para o destino de todas as portas HTTP com associações ativas.
	
	Cria regras de grupo de segurança correspondentes em AWS.

`-es` *`snapshot-id`* [*`snapshot-id`* ...] or `--ebs-snapshots` *`snapshot-id`* [*`snapshot-id`* ...]	Lista separada por vírgulas do snapshot do Amazon EBS IDs para associar ao ambiente. Example: `--ebs-snapshots snap-1234567890abcdef0, snap-0987654321fedcba1`
`--encrypt-ebs-volumes`	Aplique criptografia para todos os novos volumes do Amazon EBS. **Important** Essa é uma configuração de toda a conta que afeta toda a criação futura de volumes do Amazon EBS.
`-e` *`environment-name`* or `--environment-name` *`environment-name`*	Nome do novo ambiente do Elastic Beanstalk. Tipo: String Padrão: EBMigrated Env Restrições: devem ter entre 4 e 40 caracteres. Só pode conter letras, números e hífens. Não pode começar ou terminar com um hífen (-).

Nome	Descrição
`--force`	Ignore as solicitações de confirmação durante as operações. Quando usado com o cleanup subcomando, remove artefatos de migração sem confirmação.
`-ip` *`profile-name`* or `--instance-profile` *`profile-name`*	Perfil de instância para associar às EC2 instâncias Amazon do ambiente. Se não for especificado, cria um perfil de instância padrão com permissões para acessar os recursos do Elastic Beanstalk. Para obter mais informações, consulte the section called “Perfil de instância”.
`-i` *`instance-type`* or `--instance-type` *`instance-type`*	O tipo de EC2 instância da Amazon para seu ambiente do Elastic Beanstalk. Tipo: String Padrão: c5.2xlarge Para ver os tipos de instância disponíveis, consulte os tipos de EC2 instância da Amazon no Guia EC2 do usuário da Amazon.

`-in`
 	
or
 	
`--interactive`
	Forçar o modo interativo para o processo de migração.
	
	Solicita valores de configuração mesmo quando os padrões estão disponíveis.

Nome	Descrição

`-k key-name`
 	
or
 	
`--keyname key-name`
	Amazon EC2 key pair para permitir o acesso RDP às instâncias do ambiente.
	
	Útil para investigar problemas no nível da instância que não são visíveis nos registros.
	
	Valores válidos: um nome de par de chaves existente registrado na Amazon EC2

`-p platform-version`
 	
or
 	
`--platform platform-version`
	Tempo de execução da plataforma Elastic Beanstalk para o ambiente. Se não for especificado, detectado automaticamente a partir da versão host do Windows Server.
	
	Example: `"64bit Windows Server 2016 v2.16.2 running IIS 10.0"`
	
	Para obter uma lista das versões disponíveis da plataforma, use a lista de plataformas web.

Nome	Descrição
`--remote`	Indica a execução da migração no modo remoto. Essa opção permite a execução a partir de um bastion host, que se conecta ao servidor de destino que contém o aplicativo e as configurações a serem migradas para o Elastic Beanstalk. Executando a partir do servidor bastion, `eb migrate` descobre configurações, organiza a lógica de migração no host bastion e, em seguida, implanta seu aplicativo em um novo ambiente do Elastic Beanstalk. Essa opção elimina a necessidade de instalar o EB CLI e o Python no servidor Windows que você precisa migrar. Em vez disso, você instala o Python e o EB CLI em um bastion host, onde executa o comando com a eb migrate opção. `--remote` Use a `--target-ip` opção para especificar o host com as configurações do IIS a serem migradas. Deve ser usado com `--target-ip --username` , `--password` e.
`--target-ip` *`ip-address`*	Endereço IP público da máquina Windows remota que contém os servidores IIS a serem migrados. Necessário ao usar`--remote`. Só pode ser especificado ao usar`--remote`.
`--username` *`username`*	Nome de usuário do perfil de usuário para acessar a máquina Windows remota que contém os servidores IIS a serem migrados. Necessário ao usar`--remote`. Só pode ser especificado ao usar`--remote`.

Nome	Descrição
`--password` *password*	Senha do perfil do usuário para acessar a máquina Windows remota que contém os servidores IIS a serem migrados. Necessário ao usar`--remote`. Só pode ser especificado ao usar`--remote`.
`-sr` *role-name* or `--service-role` *role-name*	Função de serviço do IAM para o Elastic Beanstalk gerenciar serviços relacionados. AWS Se não for especificado, cria uma função de serviço padrão com as permissões necessárias. Para obter mais informações, consulte the section called “Perfil de serviço”. **Note** Especifique somente o nome da função, não o ARN completo. O Elastic Beanstalk cria automatic amente o ARN completo.
`-s` *site-names* or `--sites` *site-names*	Lista separada por vírgula dos sites do IIS a serem migrados. Se não for especificado, migra todos os sites disponíveis no servidor. Example: `--sites "Default Web Site,Intr anet,API"`
`--ssl-certificates` *certificate-arn* [,*certifica te-arn* ...]	Lista separada por vírgulas do certificado SSL do ACM ARNs para associar ao Application Load Balancer. Obrigatório ao migrar sites com ligações HTTPS. Example: `--ssl-certificates arn:aws:a cm:region:account:certificate/certif icate-id`

Nome	Descrição

`-t key1=value1[,key2=value2 ..`
 	
or
 	
`--tags key1=value1[,key2=va`
	Lista de `key=value` pares separados por vírgulas para marcar novos recursos em seu ambiente: ambiente, aplicativo Elastic Beanstalk, versão do aplicativo.
	
	Para obter mais informações, consulte Marcação em ambientes.

`--verbose`	Mostre informações detalhadas durante o processo de migração. Quando usado com o explore subcomando, exibe detalhes abrangentes da configuração do site.

Nome	Descrição

`-vpc` *`config-file-or-string`*
 	
or
 	
`--vpc-config` *`config-file-or-string`*
	Configuração de VPC para o ambiente, especificada como um caminho de arquivo JSON ou uma string JSON.
	
	A configuração deve incluir:
	
	`{`
	`"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`: (Obrigatório) Identificador de VPC
	• `publicip`: se deve atribuir público IPs às instâncias
	• `elbscheme` : esquema de balanceamento de carga (público ou privado)
	• `ec2subnets` : Lista de sub-redes IDs para instâncias EC2
	• `securitygroups` : grupo de segurança separado por vírgula IDs
	• `elbsubnets` : Lista de sub-redes IDs para o balanceador de carga
	
	Important
	A migração ignorará todas as configurações de VPC existentes do ambiente de origem quando você especificar o *`--vpc-config`* parâmetro. Quando você usa esse parâmetro , a migração usará somente as configurações de VPC especificadas no arquivo de configura

Nome	Descrição
	ção que você está transmitindo. O uso desse parâmetro substitui o comportamento padrão de descobrir a configuração da VPC da instância de origem ou usar a VPC padrão.
Opções comuns	

Saída

O comando fornece atualizações de status durante todo o processo de migração:

1. Detecção de configuração de VPC (ao executar em uma EC2 instância)
2. Progresso da geração do pacote de origem para cada site
3. Status de criação do ambiente
4. Progresso da implantação

Se for bem-sucedido, exibe os detalhes do novo ambiente, incluindo:

- Nome e ID da do ambiente
- Nome da aplicação
- Região
- Versão da plataforma
- Ambiente CNAME

Para problemas durante a migração, use os eb healthcomandos eb eventse para obter informações detalhadas.

Exemplos

Uso básico

Migração básica no modo interativo:

```
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
```

Migração de sites específicos com configuração personalizada:

```
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
```

Criação de arquivo de migração sem implantação:

```
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.
```

Exemplos de configuração avançada

Migração com configuração de VPC personalizada usando um arquivo 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"
```

Migração de sites com certificados SSL e cabeçalhos de 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
```

Migração com configuração de snapshot do 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
```

Exemplos de configuração de segurança

Lidando com sites com regras de firewall complexas:

```
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
```

Migração com funções personalizadas do IAM:

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

Exemplos de execução remota

Migrando aplicativos IIS de um servidor 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
```

Migração remota com seleção de local específico:

```
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

Descrição

Abre o URL público do site no navegador padrão.

Sintaxe

eb open

eb open ***environment-name***

Opções

Nome	Descrição
Opções comuns	

Saída

O comando eb open não gera saída. Em vez disso, ele abre o aplicativo em uma janela do navegador.

eb platform

Descrição

Este comando dá suporte a dois espaços de trabalho diferentes:

Plataforma

Use este espaço de trabalho para gerenciar plataformas personalizadas.

Ambiente

Use este espaço de trabalho para selecionar uma plataforma padrão ou mostrar informações sobre a plataforma atual.

O Elastic Beanstalk fornece o atalho ebp para eb platform.

> **Note**
>
> O Windows PowerShell usa ebp como alias de comando. Se você estiver executando o EB CLI no Windows PowerShell, use a forma longa desse comando —. eb platform

Usar eb platform para plataformas personalizadas

Lista as versões da plataforma atual e permite que você gerencie plataformas personalizadas.

Sintaxe

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***]

Opções

Nome	Descrição
`create [version] [options]`	Criar uma nova versão da plataforma. Saiba mais.
`delete version [options]`	Excluir uma versão da plataforma. Saiba mais.
`events [version] [options]`	Exibir os eventos de uma versão da plataforma. Saiba mais.
`init [platform] [options]`	Inicializar um repositório de plataforma. Saiba mais.
`list [options]`	Listar as versões da plataforma atual. Saiba mais.

Nome	Descrição
`logs [version] [options]`	Exibir logs do ambiente do compilador para uma versão da plataforma. Saiba mais.
`status [version] [options]`	Exibir o status de uma versão da plataforma. Saiba mais.
`use [platform] [options]`	Selecionar uma plataforma diferente a partir da qual novas versões são criadas. Saiba mais.
Opções comuns	

Opções comuns

Todos os comandos eb platform incluem as seguintes opções comuns.

Nome	Descrição

`-h`
OU
`--help`
	Mostra uma mensagem de ajuda e sai.

`--debug`	Mostra saída de depuração adicional.
`--quiet`	Suprime toda a saída.

`-v`
OU
`--verbose`
	Mostra saída adicional.

`--profile PROFILE`	Usa o especificado *PROFILE* de suas credenciais.
`-r REGION` OU	Use a região*REGION*.

Nome	Descrição
`--region` *REGION*	
`--no-verify-ssl`	Não verifique os certificados AWS SSL.

Eb platform create

Cria uma nova versão da plataforma e retorna o nome de região da Amazon (ARN) da nova versão. Se não houver um ambiente de compilador em execução na região atual, o comando iniciará uma. As opções de incremento *version* e (`-M`,`-m`, e`-p`) são mutuamente exclusivas.

Opções

Nome	Descrição
version	Se *version* não for especificado, cria uma nova versão com base na plataforma mais recente com a versão do patch (N em N.n.n) incrementada.

`-M`
OU
`--major-increment`
	Incrementa o número de versão principal (o N em N.n.n).

`-m`
OU
`--minor-increment`
	Incrementa o número de versão secundária (o N em n.N.n).

`-p`
OU
`--patch-increment`
	Incrementa o número de versão de patch (o N em n.n.N).

`-i` *INSTANCE_TYPE* OU	Use *INSTANCE_TYPE* como tipo de instância, como**`t1.micro`**.

Nome	Descrição
--tipo de instância *INSTANCE_TYPE*	
`-ip` *INSTANCE_PROFILE* OU `--instance-profile` *INSTANCE_PROFILE*	Use *INSTANCE_PROFILE* como perfil de instância ao criar AMIs uma plataforma personalizada. Se a opção `-ip` não estiver especificada, criará o perfil de instância `aws-elasticbeanstalk-custom-platform-ec2-role` e o usará para a plataforma personalizada.
`--tags` *key1*`=`*value1*`[,`*ke*	As tags são especificadas como uma lista de `key=value` separados por vírgulas.
`--timeout` *minutes*	O número definido de minutos antes do limite de tempo do comando.
`--vpc.id` *VPC_ID*	O ID da VPC na qual o Packer compila.
`--vpc.subnets` *VPC_SUBNETS*	As sub-redes VPC nas quais o Packer compila.
`--vpc.publicip`	Associa o público IPs às EC2 instâncias lançadas.

Eb platform delete

Excluir uma versão da plataforma. A versão não será excluída se um ambiente a estiver usando.

Opções

Nome	Descrição
version	A versão para excluir. Este valor é obrigatório.
`--cleanup`	Remover todas as versões de plataforma no estado `Failed`.
`--all-platforms`	Se `--cleanup` for especificado, remove todas as versões de plataforma no estado `Failed` para todas as plataformas.

Nome	Descrição
`--force`	Não requer confirmação ao excluir uma versão.

Eb platform events

Exibir os eventos de uma versão da plataforma. Se *`version`* for especificado, exiba os eventos dessa versão, caso contrário, exiba os eventos da versão atual.

Opções

Nome	Descrição
`version`	A versão para a qual os eventos são exibidos. Este valor é obrigatório.

`-f`
OU
`--follow`
	Continuar a exibir os eventos à medida que ocorrem.

Eb platform init

Inicializar um repositório de plataforma.

Opções

Nome	Descrição
`platform`	O nome da plataforma para inicializar. Este valor é obrigatório, a menos que `-i` (modo interativo) esteja habilitado.

`-i`
OU
`--interactive`
	Usar o modo interativo.

Nome	Descrição
`-k` *KEYNAME* OU `--keyname` *KEYNAME*	O nome da EC2 chave padrão.

Você pode executar este comando em um diretório previamente inicializado, embora não possa alterar o tipo de espaço de trabalho se executado em um diretório previamente inicializado.

Para reinicializar com opções diferentes, use a opção `-i`.

Eb platform list

Listar as versões da Plataforma associada a um espaço de trabalho (diretório) ou uma região.

O comando retorna resultados diferentes dependendo do tipo de espaço de trabalho no qual você o executa da seguinte forma:

- Em um workspace da Plataforma (um diretório inicializado por `eb platform init`), o comando retornará uma lista de todas as versões de Plataforma da Plataforma personalizada definida no workspace. Adicione a opção `--verbose` ou `--all-platforms` para obter uma lista de todas as versões de Plataforma de todas as Plataformas personalizadas que sua conta tiver na região associada ao workspace.
- Em um workspace da aplicação (um diretório inicializado por `eb init`), o comando retornará uma lista de todas as versões de plataforma, tanto para plataformas gerenciadas pelo Elastic Beanstalk quanto para as plataformas personalizadas da sua conta. A lista usa nomes curtos de versão da Plataforma, e algumas variáveis da versão da Plataforma podem ser combinadas. Adicione a opção `--verbose` para obter uma lista detalhada com nomes completos e todas as variáveis listadas separadamente.
- Em um diretório não inicializado, o comando só funciona com a opção `--region`. Ele retorna uma lista de todas as versões de plataforma gerenciada pelo Elastic Beanstalk compatíveis na região. A lista usa nomes curtos de versão da Plataforma, e algumas variáveis da versão da Plataforma podem ser combinadas. Adicione a opção `--verbose` para obter uma lista detalhada com nomes completos e todas as variáveis listadas separadamente.

Opções

Nome	Descrição

`-a`
 	
OU
 	
`--all-platforms`
	Válido somente em um workspace inicializado (um diretório inicializado por `eb platform init` ou `eb init`). Lista as versões das Plataformas de todas as Plataformas associadas à sua conta.

`-s` *`STATUS`* OU `--status` *`STATUS`*	Liste somente as plataformas que correspondem a*`STATUS`*: • Ready • Reprovada • Deleting (Excluindo) • Creating (Criando)

Eb platform logs

Exibir logs do ambiente do compilador para uma versão da plataforma.

Opções

Nome	Descrição
`version`	A versão da plataforma cujos logs são exibidos. Se omitido, exibe logs da versão atual.
`--stream`	Transmita os registros de implantação que foram configurados com CloudWatch.

Eb platform status

Exibir o status de uma versão da plataforma.

Opções

Nome	Descrição
version	A versão da plataforma cujo status é recuperado. Se omitida, exibe o status da versão atual.

Eb platform use

Selecionar uma plataforma diferente a partir da qual novas versões são criadas.

Opções

Nome	Descrição
platform	Especifica *platform* como a versão ativa desse espaço de trabalho. Este valor é obrigatório.

Usar eb platform para ambientes

Lista as plataformas compatíveis e permite que você defina a plataforma padrão e a versão da plataforma a serem usadas ao iniciar um ambiente. Use eb platform list para ver uma lista de todas as plataformas compatíveis. Use eb platform select para alterar a plataforma de seu projeto. Use eb platform show para ver a plataforma selecionada de seu projeto.

Sintaxe

eb platform list

eb platform select

eb platform show

Opções

Nome	Descrição
`list`	Listar a versão da plataforma atual.

Nome	Descrição
`select`	Selecionar a plataforma padrão.
`show`	Mostrar informações sobre a plataforma atual.

Exemplo 1

O exemplo a seguir lista os nomes de todas as configurações para todas as plataformas compatíveis com o 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
```

Exemplo 2

O exemplo a seguir solicita que você escolha a plataforma em uma lista de plataformas e a versão que você deseja implantar para a plataforma especificada.

```
$ 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)
```

Exemplo 3

O exemplo a seguir mostra informações sobre a plataforma padrão atual.

```
$ 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

Descrição

Imprime todas as propriedades de ambiente na janela de comando.

Sintaxe

eb printenv

eb printenv ***environment-name***

Opções

Nome	Descrição
Opções comuns	

Saída

Se for bem-sucedido, o comando retornará o status da operação `printenv`.

Exemplo

O exemplo a seguir imprime as propriedades de ambiente do ambiente especificado.

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

eb restore

Descrição

Reconstrói um ambiente encerrado, criando um novo ambiente com os mesmos nome, ID e configuração. O nome do ambiente, o nome de domínio e a versão do aplicativo devem estar disponíveis para uso para que a reconstrução seja bem-sucedida.

Sintaxe

eb restore

eb restore ***environment_id***

Opções

Nome	Descrição
Opções comuns	

Saída

A EB CLI exibe uma lista de ambientes encerrados que estão disponíveis para restauração.

Exemplo

```
$ 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

Descrição

Escala o ambiente para sempre executar um número específico de instâncias, definindo o número mínimo e máximo de instâncias como o número especificado.

Sintaxe

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

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

Opções

Nome	Descrição
--timeout	O número de minutos antes de esgotar o tempo limite do comando.
Opções comuns	

Saída

Se for bem-sucedido, o comando atualizará o número mínimo e máximo de instâncias a executar para o número especificado.

Exemplo

O exemplo a seguir define o número de instâncias como 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

Descrição

Define as propriedades do ambiente para o ambiente padrão.

Sintaxe

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

Você pode incluir quantas propriedades quiser, mas o tamanho total de todas as propriedades não pode exceder 4096 bytes. Você pode excluir uma variável deixando o valor em branco. Para obter os limites, consulte Configurar propriedades do ambiente (variáveis de ambiente).

> **Note**
>
> Se o `value` contiver um caractere especial, você deverá escapar esse caractere colocando o caractere \ antes dele.

Opções

Nome	Descrição
--timeout	O número de minutos antes de esgotar o tempo limite do comando.
Opções comuns	

Saída

Se for bem-sucedido, o comando exibe que a atualização do ambiente foi bem-sucedida.

Exemplo

O exemplo a seguir define a variável de 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.
```

O comando a seguir define várias propriedades de ambiente. Ele adiciona a propriedade de ambiente chamada `foo` e define seu valor para `bar`, altera o valor da propriedade `JDBC_CONNECTION_STRING` e exclui as propriedades `PARAM4` e `PARAM5`.

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

eb ssh

Descrição

> **Note**
>
> Esse comando não funciona com ambientes que executam instâncias do Windows Server.

Conecte-se a uma EC2 instância Linux da Amazon em seu ambiente usando o Secure Shell (SSH). Se um ambiente tiver várias instâncias em execução, a EB CLI solicitará que você especifique uma delas para se conectar. Para usar esse comando, o SSH deve ser instalado na máquina local e estar disponível na linha de comando. Os arquivos de chave privada devem estar localizados em uma pasta `.ssh` nomeada em seu diretório de usuário, e as EC2 instâncias em seu ambiente devem ter endereços IP públicos.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também se conectará a instâncias do ambiente personalizado.

> **Chaves SSH**
>
> Se você ainda não configurou o SSH, pode usar a EB CLI para criar uma chave ao executar eb init. Se você já executou o eb init, execute-o novamente com a opção `--interactive` e selecione Yes (Sim) e Create New Keypair (Criar novo par de chaves) quando for solicitada a configuração do SSH. As chaves criadas durante esse processo serão armazenadas na pasta correta pela EB CLI.

Esse comando abre temporariamente a porta 22 no security group do seu ambiente para o tráfego de entrada de 0.0.0.0/0 (todos os endereços IP), caso não haja regras em vigor para a porta 22. Se você tiver configurado o security group do ambiente para abrir a porta 22 para um intervalo restrito de CIDR, objetivando maior segurança, a EB CLI respeitará essa configuração, ignorando quaisquer alterações feitas no security group. Para substituir esse comportamento e forçar a EB CLI a abrir a porta 22 para todo o tráfego de entrada, use a opção `--force`.

Consulte EC2 grupos de segurança para informações sobre como configurar o grupo de segurança do ambiente.

Sintaxe

eb ssh

eb ssh ***environment-name***

Opções

Nome	Descrição

`-i`
 	
ou
 	
`--instance`
	Especifica o ID da instância à qual você se conecta. Recomendamos o uso dessa opção.

`-n`
 	
ou
 	
`--number`
	Especifique a instância para conexão por número.

`-o`
 	
ou
 	
`--keep_open`
	Deixe a porta 22 aberta no security group depois que a sessão SSH for encerrada.

`--command`	Execute um comando shell na instância especificada, em vez de iniciar uma sessão SSH.

Nome	Descrição
`--custom`	Especifique um comando SSH para usar, em vez de 'ssh -i keyfile'. Não inclua o usuário remoto e o nome do host.
`--setup`	Altere o par de chaves atribuído às instâncias do ambiente (requer a substituição das instâncias).
`--force`	Abra a porta 22 para o tráfego de entrada de 0.0.0.0/0 no security group do ambiente, mesmo que o security group já esteja configurado para SSH. Use essa opção se o security group do ambiente estiver configurado para abrir a porta 22 para um intervalo restrito de CIDR que não inclui o endereço IP do qual você está tentando se conectar.
`--timeout` *`minutes`*	O número definido de minutos antes do limite de tempo do comando. Pode ser usado com o argumento `--setup`.
Opções comuns	

Saída

Se for bem-sucedido, o comando abrirá uma conexão SSH com a instância.

Exemplo

O exemplo a seguir conecta você ao ambiente especificado.

```
$ 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

Descrição

Fornece informações sobre o status do ambiente.

Se o diretório raiz contiver um arquivo `platform.yaml` especificando uma plataforma personalizada, esse comando também fornecerá informações sobre o ambiente do compilador.

Sintaxe

eb status

eb status ***environment-name***

Opções

Nome	Descrição

`-v`
 	
ou
 	
`--verbose`
	Fornece mais informações sobre instâncias individuais, como o status delas com o load balancer do Elastic Load Balancing.

Nome	Descrição
Opções comuns	

Saída

Se tiver êxito, o comando retornará as informações a seguir sobre o ambiente:

- Nome do ambiente
- Application name
- Versão do aplicativo implantada
- ID do ambiente
- Plataforma
- Nível do ambiente
- CNAME
- Hora em que o ambiente foi atualizado pela última vez
- Status
- Integridade

Se você usa o modo detalhado, o EB CLI também fornece o número de instâncias da Amazon em execução. EC2

Exemplo

O exemplo a seguir mostra o status do 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

Descrição

Troca o CNAME do ambiente pelo CNAME de outro ambiente (por exemplo, para evitar tempo de inatividade quando você atualizar a versão do aplicativo).

> **Note**
>
> Se você tiver mais de dois ambientes, será solicitado a selecionar o nome do ambiente que está atualmente usando o CNAME desejado em uma lista de ambientes. Para suprimir isso, especifique o nome do ambiente a usar incluindo a opção `-n` quando executar o comando.

Sintaxe

eb swap

eb swap ***environment-name***

> **Note**
>
> *`environment-name`*É o ambiente para o qual você deseja um CNAME diferente. Se você não especificar *`environment-name`* como parâmetro da linha de comando ao executareb swap, o EB CLI atualizará o CNAME do ambiente padrão.

Opções

Nome	Descrição

`-n`
 	
or
 	
`--destination_name`
	Especifica o nome do ambiente com o qual você deseja trocar CNAMEs. Se você executar eb swap sem essa opção, a EB CLI solicitará que seja feita a seleção em uma lista de ambientes.

Nome	Descrição
Opções comuns	

Saída

Se for bem-sucedido, o comando retornará o status da operação `swap`.

Exemplos

O exemplo a seguir troca o ambiente tmp-dev pelo 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'.
```

O exemplo a seguir troca o ambiente tmp-dev pelo ambiente live-env, mas não solicita que você informe ou selecione um valor para as configurações.

```
$ 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

Descrição

Adicione, exclua, atualize e liste tags de um recurso do Elastic Beanstalk.

Para obter detalhes sobre a marcação de recursos no Elastic Beanstalk, consulte Marcar recursos da aplicação do Elastic Beanstalk.

Sintaxe

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* ...]

Você pode combinar as opções de subcomando `--add`, `--update` e `--delete` em um único comando. Pelo menos uma delas é necessária. Você não pode combinar nenhuma dessas três opções de subcomando com `--list`.

Sem nenhum argumento adicional, todos esses comandos listam ou modificam tags do ambiente padrão no aplicativo do diretório atual. Com um *environment-name* argumento, os comandos listam ou modificam as tags desse ambiente. Com a opção `--resource`, os comandos listam ou modificam as tags de qualquer recurso do Elastic Beanstalk: uma aplicação, um ambiente, uma versão da aplicação, uma configuração salva ou uma versão da plataforma personalizada. Especifique o recurso pelo seu nome de recurso da Amazon (ARN).

Opções

Nenhuma dessas opções é obrigatória. Se você executar eb create sem opções, será solicitado a inserir ou selecionar um valor para cada configuração.

Nome	Descrição

`-l`
 	
ou
 	
`--list`
	Listar todas as tags que estão sendo aplicadas no recurso atualmente.

`-a` *key1*=*value1*[,*key2*=*value2*] ou	Aplique novas tags no recurso. Especifique as tags como uma lista de `key=value` separados por vírgulas. Não é possível especificar chaves de tags existentes.

Nome	Descrição
`--add key1=value1[,key2=val`	Valores válidos. Veja Marcar recursos.

`-u key1=value1[,key2=value2`
 	
ou
 	
`--updat e key1=value1[,key2=value2 .`
	Atualize os valores de tags de recursos existentes. Especifique as tags como uma lista de `key=value` separados por vírgulas. Você precisa especificar chaves de tags existentes.
	
	Valores válidos. Veja Marcar recursos.

`-d key1[,key2 ...]`
 	
ou
 	
`--delete key1[,key2 ...]`
	Exclua as tags de recursos existentes. Especifique as tags como uma lista de chaves separados por vírgulas. Você precisa especificar chaves de tags existentes.
	
	Valores válidos. Veja Marcar recursos.

`-r region`
 	
or
 	
`--region region`
	O Região da AWS em que seu recurso existe.
	
	Padronizado: a região padrão configurada.
	
	Para obter a lista de valores que você pode especificar para essa opção, consulte Endpoints e cotas do AWS Elastic Beanstalk no Referência geral da AWS.

`--resource ARN`	O ARN do recurso para o qual o comando modifica ou lista tags. Se não for especificado, o comando refere-se ao ambiente (padrão ou especificado) no aplicativo do diretório atual. Valores válidos: consulte um dos subtópicos de Marcar recursos específico do recurso de interesse. Esses tópicos mostram como o ARN do recurso é construído e explicam como obter uma lista dos recursos existentes para seu aplicativo ou conta. ARNs

Saída

A opção de subcomando de `--list` exibe uma lista das tags de recursos. A saída mostra as tags que o Elastic Beanstalk aplica por padrão e suas tags personalizadas.

```
$ 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
```

As opções de subcomando `--add`, `--update` e `--delete`, quando bem-sucedidas, não têm nenhuma saída. Você pode adicionar a opção `--verbose` para ver saída detalhada da atividade do comando.

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

Key                                 Value

mytag                               tag value
```

Exemplos

O comando a seguir adiciona uma tag com êxito com a chave `tag1` e o valor `value1` ao ambiente padrão do aplicativo e, ao mesmo tempo, exclui a tag `tag2`.

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

O comando a seguir adiciona uma tag com êxito a uma configuração salva em um aplicativo.

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

O comando a seguir falha porque ele tenta atualizar uma tag não existente.

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

  tag3
```

O comando a seguir falha porque ele tenta atualizar e excluir a mesma chave.

```
$ 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

Descrição

Encerra o ambiente em execução para que você não incorra em cobranças por recursos não AWS utilizados.

O uso da opção `--all` exclui o aplicativo para o qual o diretório atual foi inicializado usando eb init. O comando termina todos os ambientes na aplicação. Termina também todos as versões da aplicação e as configurações salvas da aplicação e depois exclui a aplicação.

Se o diretório raiz contém um arquivo `platform.yaml` que especifica uma plataforma personalizada, esse comando encerra o ambiente personalizado em execução.

> **Note**
>
> Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Se você tiver dados de um ambiente que deseja manter, defina a política de exclusão de banco de dados como `Retain` antes de encerrar o ambiente. Isso mantém o banco de dados operacional fora do Elastic Beanstalk. Depois disso, qualquer ambiente do Elastic Beanstalk deve se conectar a ele como um banco de dados externo. Se você quiser fazer backup dos dados sem manter o banco de dados operacional, defina a política de exclusão para fazer um snapshot do banco de dados antes

de encerrar o ambiente. Para obter mais informações, consulte Ciclo de vida do banco de dados no capítulo Configuração de ambientes deste guia.

> **Important**
>
> Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.
>
> Para obter mais informações, consulte Proteção contra um registro de delegação pendente no Route 53 no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em Proteções de domínio aprimoradas para solicitações da CloudFront Amazon no AWS blog de segurança.

Sintaxe

eb terminate

eb terminate ***environment-name***

Opções

Nome	Descrição
`--all`	Encerra todos os ambientes no aplicativo, bem como as versões e as configurações salvas do aplicativo, e em seguida exclui o aplicativo.
`--force`	Termina o ambiente sem pedir confirmação.
`--ignore-links`	Termina o ambiente, mesmo se houver ambientes dependentes vinculados a ele. Consulte Ambientes do Compose.
`--timeout`	O número de minutos antes de esgotar o tempo limite do comando.

Saída

Se for bem-sucedido, o comando retornará o status da operação `terminate`.

Exemplo

A solicitação de exemplo a seguir encerra o 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

Descrição

Atualiza a plataforma de seu ambiente para a versão mais recente da plataforma na qual ele está sendo executado no momento.

Se o diretório raiz contém um arquivo `platform.yaml` especificando uma plataforma personalizada, este comando atualiza o ambiente para a versão mais recente da plataforma personalizada em que está em execução no momento.

Sintaxe

eb upgrade

eb upgrade ***environment-name***

Opções

Nome	Descrição
`--force`	Atualiza sem a necessidade de que você confirme o nome do ambiente antes de iniciar o processo de atualização.
`--noroll`	Atualiza todas as instâncias sem usar as atualizações contínuas para manter algumas instâncias em serviço durante a atualizaç ão.
Opções comuns	

Saída

O comando mostra uma visão geral da alteração e solicita que você confirme a atualização digitando o nome do ambiente. Em caso de êxito, seu ambiente é atualizado e iniciado com a versão mais recente da plataforma.

Exemplo

O exemplo a seguir atualiza a versão da plataforma atual do ambiente especificado para a versão mais recente disponível da plataforma.

```
$ 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

Descrição

Define o ambiente especificado como o padrão.

Ao usar o Git, eb use define o ambiente padrão para a ramificação. Execute esse comando uma vez em cada ramificação que você deseje implantar no Elastic Beanstalk.

Sintaxe

eb use ***environment-name***

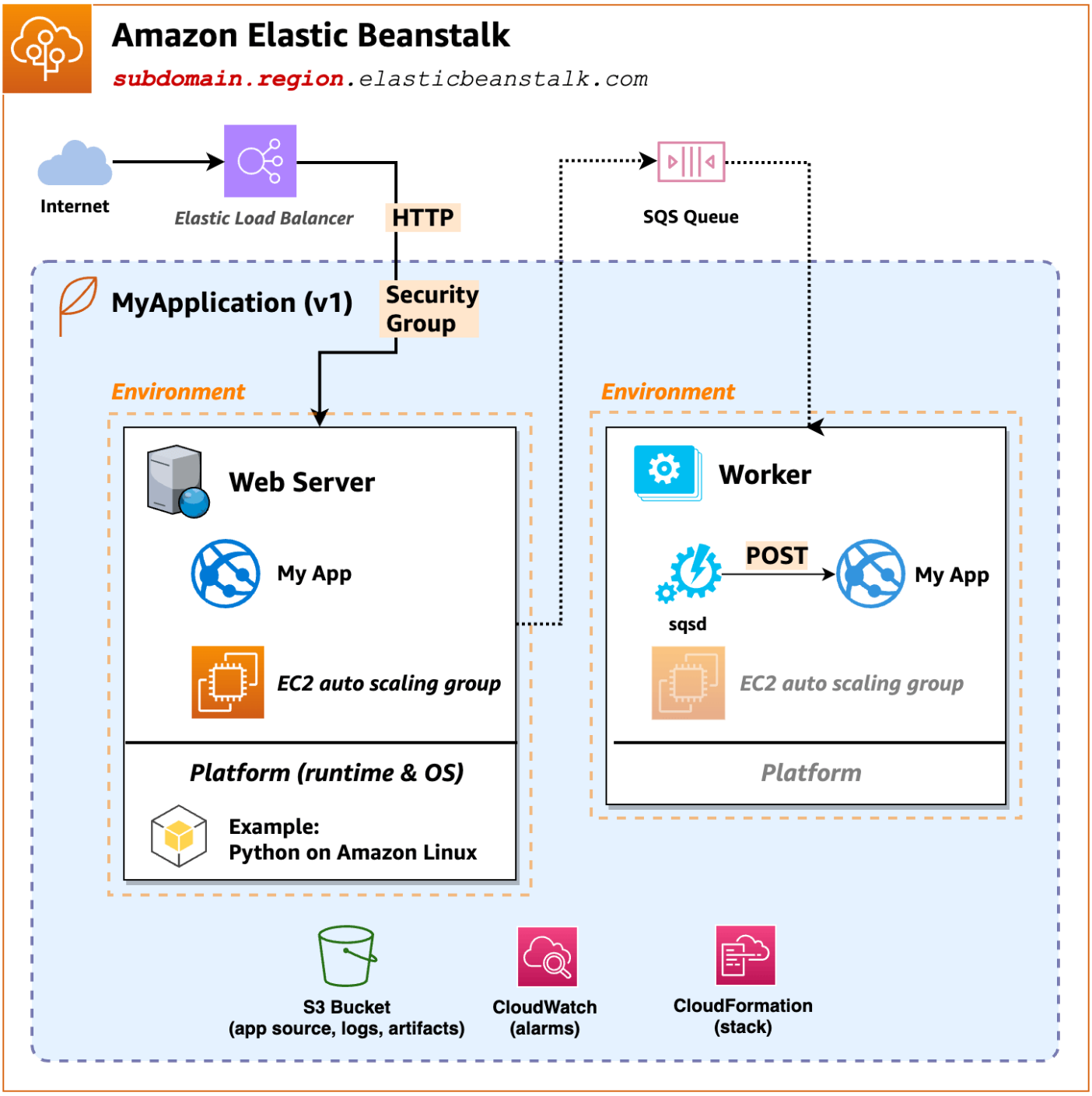
Opções

Nome	Descrição
`--source codecommit/` *repository-name*`/`*branch-name*	CodeCommit repositório e filial.
`-r` *region* `--region` *region*	Altere a região na qual você cria ambientes.

Nome	Descrição
Opções comuns	

Entendendo conceitos no Elastic Beanstalk

Familiarizar-se com os conceitos e os termos ajudará você a obter a compreensão necessária para implantar seus aplicativos com o Elastic Beanstalk.



Aplicação

Um aplicativo do Elastic Beanstalk é um contêiner para componentes do Elastic Beanstalk, incluindo ambientes, versões e configurações de ambiente. Em um aplicativo do Elastic Beanstalk, você gerencia todos os recursos relevantes para executar seu código.

Versão da aplicação

No Elastic Beanstalk, uma versão da aplicação se refere a uma iteração rotulada específica do código implantável de uma aplicação Web. Uma versão do aplicativo aponta para um objeto do Amazon Simple Storage Service (Amazon S3) que contém o código implantável, como um arquivo WAR Java.

Uma versão do aplicativo faz parte de um aplicativo. Os aplicativos podem ter várias versões e cada uma delas é única. Em um ambiente em execução, é possível implantar qualquer versão do aplicativo já carregada no aplicativo ou fazer upload e implantar imediatamente uma nova versão do aplicativo. Por exemplo, você pode fazer upload de várias versões do aplicativo para testar as diferenças entre elas.

Environment

Um ambiente é uma coleção de AWS recursos executando uma versão do aplicativo. Cada ambiente executa somente uma versão do aplicativo por vez, no entanto, você pode executar a mesma versão ou diferentes versões do aplicativo em vários ambientes ao mesmo tempo. Quando você cria um ambiente, o Elastic Beanstalk provisiona os recursos necessários AWS em sua conta para executar a versão do aplicativo que você especificou.

Nível do ambiente

Ao iniciar um ambiente do Elastic Beanstalk, primeiro escolha um nível de ambiente. O nível do ambiente designa o tipo de aplicativo que o ambiente executa e determina quais recursos o Elastic Beanstalk provisiona para suportá-lo. Um aplicativo que atende a solicitações HTTP é executado em um nível de ambiente de servidor da web. Um ambiente de back-end que extrai tarefas de uma fila do Amazon Simple Queue Service (Amazon SQS) é executado em uma camada de ambiente de operador.

Configuração de ambientes

Uma configuração de ambiente identifica um conjunto de parâmetros e configurações que definem como um ambiente e seus recursos associados se comportam. Quando você atualiza as definições de configuração de um ambiente, o Elastic Beanstalk aplica automaticamente as alterações aos recursos existentes ou exclui e implanta novos recursos (dependendo do tipo de alteração).

Configuração salva

Uma configuração salva é um modelo que você pode usar como um ponto de partida para a criação de configurações exclusivas de ambiente. Você pode criar e modificar configurações salvas e aplicá-las aos ambientes usando o console, o EB AWS CLI CLI ou a API do Elastic Beanstalk. A API e o AWS CLI se referem às configurações salvas como modelos de configuração.

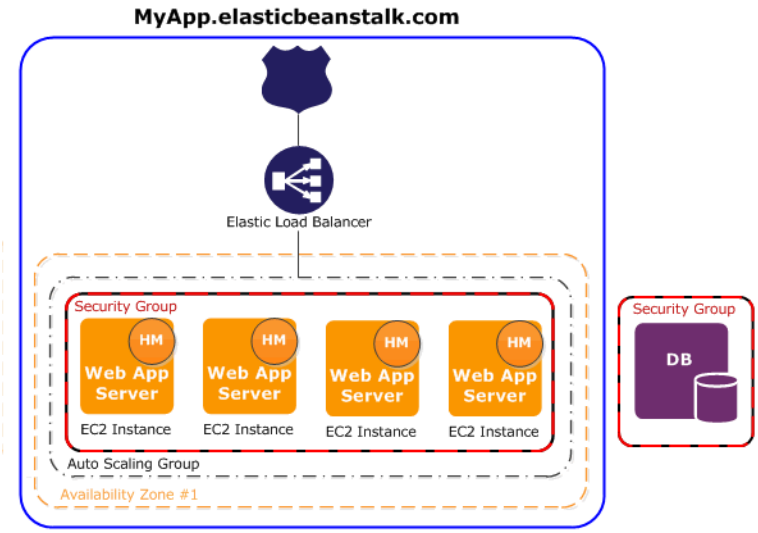
Plataforma

Uma plataforma é uma combinação de um sistema operacional, runtime da linguagem de programação, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. Projete e segmente a aplicação Web para uma plataforma. O Elastic Beanstalk oferece uma variedade de plataformas para criação de aplicações.

Para obter mais detalhes, consulte Plataformas do Elastic Beanstalk.

Ambientes de servidor Web do Elastic Beanstalk

O diagrama a seguir mostra uma arquitetura demonstrativa do Elastic Beanstalk para um nível de ambiente de servidor Web e mostra como os componentes desse tipo de nível de ambiente funcionam em conjunto.



O ambiente é a essência do aplicativo. No diagrama, o ambiente é mostrado na linha sólida de nível superior. Quando você cria um ambiente, o Elastic Beanstalk provisiona os recursos necessários para executar seu aplicativo. AWS os recursos criados para um ambiente incluem um balanceador de carga elástico (ELB no diagrama), um grupo de Auto Scaling e uma ou mais instâncias do Amazon Elastic Compute Cloud (EC2Amazon).

Cada ambiente tem um CNAME (URL) que aponta para um load balancer. O ambiente tem um URL, como `myapp.us-west-2.elasticbeanstalk.com`. O alias desse URL no Amazon Route 53 está definido como um URL do Elastic Load Balancing: algo como `abcdef-123456.us-west-2.elb.amazonaws.com`, usando um registro CNAME. O Amazon Route 53 é um serviço da web do Sistema de Nomes de Domínio (DNS) altamente disponível e dimensionável. Isso proporciona um roteamento seguro e confiável para a infraestrutura. O nome de domínio que você registrou com o seu provedor DNS encaminhará as solicitações para o CNAME.

O balanceador de carga fica na frente das EC2 instâncias da Amazon, que fazem parte de um grupo de Auto Scaling. O Amazon EC2 Auto Scaling inicia automaticamente EC2 instâncias adicionais da Amazon para acomodar o aumento da carga em seu aplicativo. Se a carga em seu aplicativo diminuir, o Amazon EC2 Auto Scaling interrompe as instâncias, mas sempre deixa pelo menos uma instância em execução.

A pilha de software em execução nas EC2 instâncias da Amazon depende do tipo de contêiner. O tipo de contêiner define a topologia da infraestrutura e a pilha de software a serem usadas nesse ambiente. Por exemplo, o ambiente do Elastic Beanstalk com um contêiner Apache Tomcat usa o sistema operacional Amazon Linux, o servidor Web Apache e o software Apache Tomcat. Para ver a lista de tipos de contêiner compatíveis, consulte Plataformas compatíveis com Elastic Beanstalk.

Cada EC2 instância da Amazon que executa seu aplicativo usa um desses tipos de contêiner. Além disso, um componente de software chamado gerenciador de host (HM) é executado em cada EC2 instância da Amazon. O gerenciador de host é responsável pelo seguinte:

- Implantação do aplicativo
- Agregação de eventos e métricas de recuperação por meio do console, API ou linha de comando
- Geração de eventos no nível de instância
- Monitoramento dos arquivos de log do aplicativo quanto a erros críticos
- Monitoramento do servidor de aplicativos
- Correção dos componentes de instância
- Rotação de arquivos de log do aplicativo e sua publicação no Amazon S3

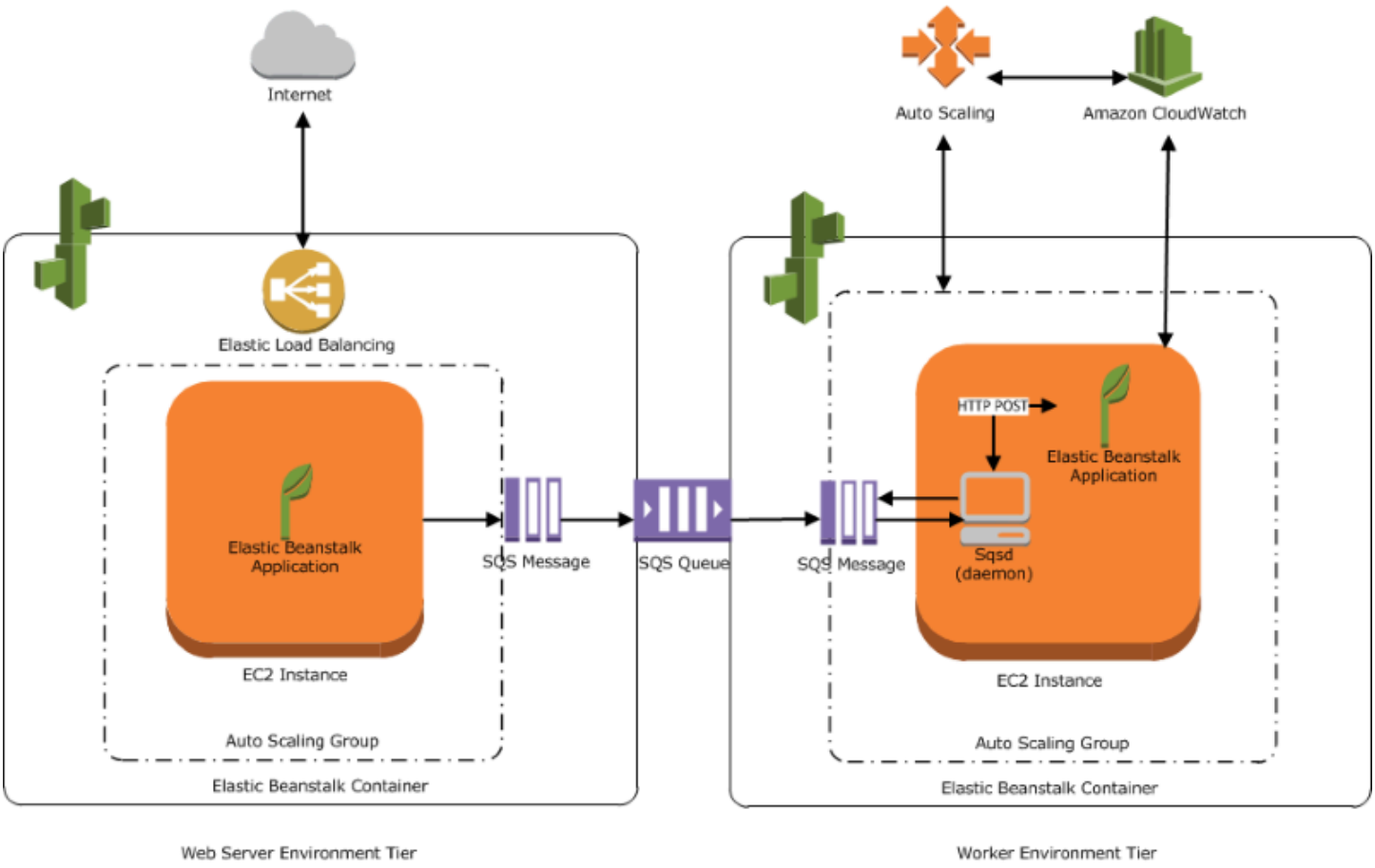
O gerente do host relata métricas, erros e eventos e o status da instância do servidor, que estão disponíveis por meio do console do Elastic APIs Beanstalk, e. CLIs

As EC2 instâncias da Amazon mostradas no diagrama fazem parte de um grupo de segurança. Um security group define as regras de firewall para suas instâncias. Por padrão, o Elastic Beanstalk define um grupo de segurança, que permite que qualquer pessoa se conecte usando a porta 80 (HTTP). Você pode definir mais de um security group. Por exemplo, você pode definir um grupo de segurança para seu servidor de banco de dados. Para obter mais informações sobre os grupos EC2 de segurança da Amazon e como configurá-los para seu aplicativo Elastic Beanstalk, consulte. EC2 grupos de segurança

Ambientes de operador do Elastic Beanstalk

AWS os recursos criados para um nível de ambiente de trabalho incluem um grupo de Auto Scaling, uma ou mais EC2 instâncias da Amazon e uma função do IAM. Para o nível de ambiente de operador, o Elastic Beanstalk também cria e provisiona uma fila do Amazon SQS se você ainda não tiver uma. Quando você inicia um ambiente de trabalho, o Elastic Beanstalk instala os arquivos de suporte necessários para sua linguagem de programação preferida e um daemon em EC2 cada instância do grupo Auto Scaling. O daemon lê mensagens de uma fila do Amazon SQS. O daemon envia os dados de cada mensagem que lê para o aplicativo web em execução no ambiente de operador para processamento. Se você tiver várias instâncias no ambiente de operador, cada instância terá seu próprio daemon, mas todas farão a leitura da mesma fila do Amazon SQS.

O diagrama a seguir mostra os diferentes componentes e suas interações entre ambientes e AWS serviços.



A Amazon CloudWatch é usada para alarmes e monitoramento de saúde. Para obter mais informações, acesse Relatórios de integridade básica.

Para ver mais detalhes sobre como funciona o nível de ambiente de operador, consulte Ambientes de operador do Elastic Beanstalk.

Considerações de design para seus aplicativos do Elastic Beanstalk

Como os aplicativos implantados usando Nuvem AWS recursos são AWS Elastic Beanstalk executados, você deve ter em mente vários fatores de configuração para otimizar seus aplicativos: escalabilidade, segurança, armazenamento persistente, tolerância a falhas, entrega de conteúdo, atualizações e patches de software e conectividade. Cada um desses fatores é abordado separadamente neste tópico. Para obter uma lista abrangente de AWS white papers técnicos,

abrangendo tópicos como arquitetura, segurança e economia, consulte Whitepapers sobre computação em AWS nuvem.

Escalabilidade

Quando estiver operando em um ambiente de hardware físico, não em um ambiente de nuvem, você pode abordar a escalabilidade de uma das duas maneiras. Você pode aumentar a escala na vertical ou aumentar a escala na horizontal. A abordagem de aumentar a escala na vertical exige que você invista em hardware poderoso, que possa suportar as demandas crescentes de seus negócios. A abordagem de aumentar a escala na horizontal exige que você adote um modelo de investimento distribuído. Dessa maneira, suas aquisições de hardware e de aplicações podem ser mais direcionadas, seus conjuntos de dados são federados e seu design é orientado a serviço. A abordagem de aumentar a escala vertical pode ser cara e ainda há o risco de que a demanda supere a capacidade. Nesse aspecto, a abordagem de aumentar a escala na horizontal geralmente é mais eficaz. Entretanto, quando usar essa abordagem, você deve ser capaz de prever a demanda em intervalos regulares e implantar a infraestrutura em blocos para atender a essa demanda. Por isso, essa abordagem, muitas vezes, pode resultar em capacidade não utilizada e exigir monitoramento cuidadoso.

Migrando para a nuvem, você pode alinhar bem a infraestrutura com a demanda, aproveitando a elasticidade da nuvem. A elasticidade ajuda a otimizar a aquisição e a liberação de recursos. Com ela, a escalda da infraestrutura pode aumentar e diminuir rapidamente, de acordo com a flutuação de demanda. Para usá-la, configure a autoescalabilidade para aumentar ou diminuir a escala com base nas métricas dos recursos de ambiente (uso dos servidores ou E/S de rede, por exemplo). Por exemplo, você pode definir métricas como utilização do servidor ou E/S de rede. Pode usar a autoescalabilidade para adicionar automaticamente capacidade computacional sempre que o uso aumentar e removê-la sempre que o uso cair. Você pode publicar métricas do sistema (por exemplo, CPU, memória, E/S de disco e E/S de rede) na Amazon. CloudWatch Em seguida, você pode usar CloudWatch para configurar alarmes para acionar ações de Auto Scaling ou enviar notificações com base nessas métricas. Para obter mais instruções sobre como configurar a autoescalabilidade, consulte Auto Scaling de suas instâncias do ambiente do Elastic Beanstalk.

Também recomendamos que você crie todas as aplicações do Elastic Beanstalk o mais sem estado possível, usando componentes com baixo acoplamento e tolerantes a falhas, cuja escala possa ser aumentada conforme necessário. Para obter mais informações sobre como projetar arquiteturas de aplicativos escaláveis para AWS, consulte Well-Architected AWS Framework.

Segurança

A segurança ativada AWS é uma responsabilidade compartilhada. A Amazon Web Services protege os recursos físicos do ambiente e garante que a nuvem seja um lugar seguro para a execução de aplicações. Você é responsável pela segurança dos dados que chegam e saem do ambiente do Elastic Beanstalk e pela segurança da aplicação.

Configure o SSL para proteger os fluxos entre a aplicação e os clientes. Para configurar o SSL, você precisa de um certificado gratuito do AWS Certificate Manager (ACM). Se já tiver um certificado emitido por uma autoridade de certificação (CA) externa, você poderá usar o ACM para importar esse certificado. Caso contrário, você pode importá-lo usando AWS CLI o.

Se o ACM não estiver disponível no seu Região da AWS, você poderá comprar um certificado de uma CA externa, como VeriSign ou Entrust. Em seguida, use o AWS Command Line Interface (AWS CLI) para carregar um certificado de terceiros ou autoassinado e uma chave privada para AWS Identity and Access Management (IAM). A chave pública do certificado autentica o seu servidor para o navegador. Também serve como base para criar a chave de sessão compartilhada que criptografa os dados nas duas direções. Para obter instruções sobre como criar, carregar e atribuir um certificado SSL para o ambiente, consulte Configurar HTTPS para o seu ambiente do Elastic Beanstalk.

Quando você configura um certificado SSL para o ambiente, os dados são criptografados entre o cliente e o balanceador de carga do Elastic Load Balancing do ambiente. Por padrão, a criptografia é encerrada no balanceador de carga e o tráfego entre o balanceador de carga e as EC2 instâncias da Amazon não é criptografado.

Armazenamento persistente

Os aplicativos do Elastic Beanstalk são executados EC2 em instâncias da Amazon que não têm armazenamento local persistente. Quando as EC2 instâncias da Amazon são encerradas, o sistema de arquivos local não é salvo. As novas EC2 instâncias da Amazon começam com um sistema de arquivos padrão. Recomendamos que você configure sua aplicação para armazenar dados em uma origem dos dados persistente. A AWS oferece vários serviços de armazenamento persistente que você pode usar em sua aplicação. Veja-os na tabela a seguir.

Serviços de armazenamento	Documentação do serviço	Integração do Elastic Beanstalk
Amazon S3	Documentação do Amazon Simple Storage Service	Usar o Elastic Beanstalk com o Amazon S3
Amazon Elastic File System	Documentação do Amazon Elastic File System	Usar o Elastic Beanstalk com o Amazon Elastic File System
Amazon Elastic Block Store	Amazon Elastic Block Store Guia de recursos: Elastic Block Store	
Amazon DynamoDB	Documentação do Amazon DynamoDB	Usar o Elastic Beanstalk com o Amazon DynamoDB
Amazon Relational Database Service (RDS)	Documentação do Amazon Relational Database Service	Usar o Elastic Beanstalk com o Amazon RDS

Note

O Elastic Beanstalk cria um usuário de webapp para você configurar como proprietário dos diretórios de aplicativos nas instâncias. EC2 Para as versões da plataforma Amazon Linux 2 lançadas a partir de 3 de fevereiro de 2022, o Elastic Beanstalk atribui ao usuário webapp um valor de uid (id do usuário) e gid (id do grupo) de 900 para novos ambientes. Ele faz o mesmo para os ambientes existentes após uma atualização de versão da plataforma. Essa abordagem mantém uma permissão consistente de acesso. ao armazenamento do sistema de arquivos permanente para o usuário webapp.
Na situação improvável de que outro usuário ou processo já esteja usando 900, o sistema operacional usa, por padrão, outro valor para o uid e o gid do usuário webapp. Execute o comando Linux id webapp em suas EC2 instâncias para verificar os valores uid e gid atribuídos ao usuário do webapp.

Tolerância a falhas

Como regra geral, seja pessimista ao projetar a arquitetura para a nuvem. Aproveite a elasticidade que ela oferece. Sempre projete, implemente e implante visando a recuperação automatizada de falha. Use várias zonas de disponibilidade para suas EC2 instâncias da Amazon e para o Amazon RDS. As zonas de disponibilidade são conceitualmente semelhantes a datacenters lógicos. Use CloudWatch a Amazon para obter mais visibilidade sobre a integridade do seu aplicativo Elastic Beanstalk e tomar as medidas apropriadas em caso de falha de hardware ou degradação do desempenho. Defina suas configurações de Auto Scaling para manter sua frota de EC2 instâncias da Amazon em um tamanho fixo para que as instâncias insalubres da EC2 Amazon sejam substituídas por novas. Se você estiver usando o Amazon RDS, defina o período de retenção de backups, para que o Amazon RDS possa fazer backups automatizados.

Entrega de conteúdo

Quando os usuários se conectam ao seu site, suas solicitações podem ser roteadas por meio de várias redes individuais. Como resultado, os usuários podem experimentar uma performance insatisfatória devido à alta latência. A Amazon CloudFront pode ajudar a melhorar os problemas de latência distribuindo seu conteúdo da web, como imagens e vídeos, em uma rede de pontos de presença em todo o mundo. As solicitações dos usuários são encaminhadas para o ponto de presença mais próximo, para que o conteúdo seja entregue com o melhor desempenho possível. CloudFront funciona perfeitamente com o Amazon S3, que armazena de forma duradoura as versões originais e definitivas dos seus arquivos. Para obter mais informações sobre a Amazon CloudFront, consulte o Amazon CloudFront Developer Guide.

Atualizações e correções de software

AWS Elastic Beanstalk lança regularmente atualizações de plataforma para fornecer correções, atualizações de software e novos recursos. O Elastic Beanstalk oferece várias opções de como lidar com atualizações de plataforma. As atualizações da plataforma gerenciadas atualizam automaticamente o ambiente para a versão mais recente da plataforma durante uma janela de manutenção programada, enquanto sua aplicação está em manutenção. Em ambientes criados a partir de 25 de novembro de 2019 usando o console do Elastic Beanstalk, as atualizações gerenciadas são habilitadas por padrão sempre que possível. Você também pode iniciar atualizações manualmente usando o console do Elastic Beanstalk ou a CLI do EB.

Conectividade

O Elastic Beanstalk precisa ser capaz de se conectar a instâncias do ambiente para concluir as implantações. Na implantação de uma aplicação do Elastic Beanstalk em uma Amazon VPC, a configuração necessária para permitir a conectividade depende do tipo de ambiente da Amazon VPC que você cria:

- Para ambientes de instância única, nenhuma configuração adicional é necessária. Isso ocorre porque, nesses ambientes, o Elastic Beanstalk atribui a EC2 cada instância da Amazon um endereço IP elástico público que permite que a instância se comunique diretamente com a Internet.
- Para os ambientes com balanceamento de carga e escaláveis em uma Amazon VPC com sub-redes públicas e privadas, faça o seguinte:
  - Crie um balanceador de carga na sub-rede pública para rotear o tráfego de entrada da Internet para as instâncias da Amazon. EC2
  - Crie um dispositivo de tradução de endereços de rede (NAT) para rotear o tráfego de saída das EC2 instâncias da Amazon em sub-redes privadas para a Internet.
  - Crie regras de roteamento de entrada e saída para as EC2 instâncias da Amazon dentro da sub-rede privada.
  - Se você estiver usando uma instância NAT, configure os grupos de segurança da instância NAT e das EC2 instâncias da Amazon para permitir a comunicação pela Internet.
- Para um ambiente escalável e com balanceamento de carga em uma Amazon VPC que tenha uma sub-rede pública, nenhuma configuração adicional é necessária. Isso ocorre porque, com esse ambiente, suas EC2 instâncias da Amazon são configuradas com um endereço IP público que permite que as instâncias se comuniquem com a Internet.

Para obter mais informações sobre como usar o Elastic Beanstalk com a Amazon VPC, consulte Usar o Elastic Beanstalk com Amazon VPC.

Gerenciamento de aplicações Elastic Beanstalk

Este capítulo descreve como gerenciar e configurar suas aplicações Elastic Beanstalk. A primeira etapa do uso AWS Elastic Beanstalk é criar um aplicativo, que represente seu aplicativo web em AWS. No Elastic Beanstalk, uma aplicação serve como contêiner para os ambientes que executam sua aplicação Web e para as versões de seu código-fonte, as configurações salvas, os logs e outros artefatos da aplicação Web criados ao usar o Elastic Beanstalk.

Para criar uma aplicação

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Applications (Aplicações) e Create a new application (Criar uma nova aplicação).
3. Use o formulário na tela para fornecer um nome de aplicativo.
4. (Opcional) Forneça uma descrição e adicione chaves e valores de etiqueta.
5. Escolha Criar.

Após a criação do aplicativo, o console solicita que você crie um ambiente para ele. Para obter informações detalhadas sobre todas as opções disponíveis, consulte Criar um ambiente do Elastic Beanstalk.

Se você não precisar mais de um aplicativo, poderá excluí-lo.

> **Warning**
>
> A exclusão de um aplicativo encerra todos os ambientes associados e exclui todas as versões e configurações salvas do aplicativo.

Como excluir uma aplicação do

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Applications (Aplicativos) e selecione seu aplicativo na lista.
3. Escolha Ações e Excluir aplicativo.

Tópicos

- Console de gerenciamento de aplicações do Elastic Beanstalk
- Gerenciar versões de aplicativos
- Crie um pacote de origem da aplicação Elastic Beanstalk
- Usando o EB CLI com AWS CodeBuild
- Marcar aplicativos
- Marcar recursos da aplicação do Elastic Beanstalk

Console de gerenciamento de aplicações do Elastic Beanstalk

Este tópico explica como você pode usar o AWS Elastic Beanstalk console para gerenciar aplicativos, versões de aplicativos e configurações salvas.

Para acessar o console de gerenciamento de aplicativos

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

   A página de visão geral do aplicativo exibe uma lista com uma visão geral de todos os ambientes associados ao aplicativo.
3. Você tem algumas maneiras de continuar:

   a. No menu suspenso Ações, você pode escolher uma das ações de gerenciamento de aplicações: Criar ambiente, Excluir aplicação, Visualizar versões da aplicação, Visualizar configurações salvas, Restaurar ambiente encerrado.

      Para iniciar um ambiente nesta aplicação, escolha diretamente Criar ambiente. Para obter detalhes, consulte the section called “Criar ambientes”.

   b. A página lista o nome do ambiente ao lado das aplicações implantadas em um ambiente. Escolha um nome de ambiente para acessar o console de gerenciamento de ambiente desse ambiente em que é possível configurar, monitorar ou gerenciar o ambiente.

   c. Quando você seleciona uma aplicação na lista, o painel de navegação esquerdo lista essa aplicação.

      - Escolha Application versions (Versões do aplicativo) após o nome do aplicativo no painel de navegação para visualizar e gerenciar as versões do aplicativo.

Uma versão do aplicativo é uma versão carregada do código do aplicativo. É possível fazer upload de novas versões, implantar uma versão existente em qualquer um dos ambientes do aplicativo ou excluir versões antigas. Para obter mais informações, consulte Gerenciar versões de aplicativos.

- Escolha Saved configurations (Configurações salvas) após o nome do aplicativo no painel de navegação para visualizar e gerenciar configurações salvas de ambientes em execução.

  Uma configuração salva é um conjunto de configurações que você pode usar para restaurar configurações de um ambiente para um estado anterior ou para criar um ambiente com as mesmas configurações. Para obter mais informações, consulte Usar configurações salvas do Elastic Beanstalk.

Gerenciar versões de aplicativos

Este tópico explica as versões de aplicações e como criá-las e gerenciá-las.

O Elastic Beanstalk cria uma versão da aplicação sempre que você faz upload do código-fonte. Isso normalmente ocorre quando você cria um ambiente ou faz upload e implanta o código usando o console de gerenciamento do ambiente ou a CLI do EB. O Elastic Beanstalk exclui essas versões da aplicação de acordo com a política de ciclo de vida da aplicação e quando a aplicação é excluída. Para obter detalhes sobre a política de ciclo de vida de aplicativo, consulte Definir as configurações de ciclo de vida da versão do aplicativo.

Também é possível fazer upload de um pacote de origem sem implantá-lo usando o console de gerenciamento da aplicaçãoou com o comando eb appversion da EB CLI. O Elastic Beanstalk armazena pacotes de origem no Amazon Simple Storage Service (Amazon S3) e não os exclui automaticamente.

Você pode aplicar tags a uma versão do aplicativo ao criá-la e editar tags de versões existentes do aplicativo. Para obter detalhes, consulte Atribuir tags em versões do aplicativo.

Criação de versões de aplicações

Você também pode criar outra versão da aplicação usando a EB CLI. Para obter mais informações, consulte eb appversion no capítulo Comandos da EB CLI.

> **Note**
>
> Com o tempo, seu aplicativo pode acumular várias versões do aplicativo. Para economizar espaço de armazenamento e evitar atingir a cota de versões da aplicação, é recomendável excluir versões da aplicação que não são mais necessárias.

O arquivo especificado no procedimento a seguir está associado à sua aplicação. Você pode implantar a versão do aplicativo em um ambiente novo ou existente.

Criar uma nova versão da aplicação

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Escolha Carregar. Use o formulário na tela para carregar o pacote de origem do aplicativo.

   > **Note**
   >
   > O tamanho máximo do arquivo do pacote de fontes é 500 MB.

5. Opcionalmente, forneça uma breve descrição e adicione chaves e valores de tag.
6. Escolha Carregar.

Exclusão de versões de aplicações

Você também pode excluir uma versão da aplicação usando a EB CLI. Para obter mais informações, consulte eb appversion no capítulo Comandos da EB CLI.

> **Note**
>
> A exclusão de uma versão do aplicativo não afeta os ambientes que a estão executando no momento.

Também é possível configurar o Elastic Beanstalk para excluir versões antigas automaticamente ao definir as configurações do ciclo de vida de versões da aplicação. Se você definir essas configurações de ciclo de vida do aplicativo, elas serão aplicadas a novas versões do aplicativo. Por exemplo, se você configurar, no máximo, 25 versões da aplicação, o Elastic Beanstalk excluirá a versão mais antiga quando você fizer upload da 26ª versão. Se você definir um período máximo de 90 dias, todas as versões com mais de 90 dias serão excluídas quando você fizer upload de uma nova versão. Para obter detalhes, consulte the section called “Ciclo de vida da versão”.

Excluir uma versão da aplicação

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Selecione uma ou mais versões do aplicativo que você deseja excluir.
5. Escolha Actions (Ações) e Delete (Excluir).
6. (Opcional) Para deixar o pacote de origem da aplicação para essas versões de aplicações no bucket do Amazon Simple Storage Service (Amazon S3), desmarque a caixa Delete versions from Amazon S3 (Excluir versões do Amazon S3).
7. Escolha Excluir.

Se você não optar por excluir o pacote de origem do Amazon S3, o Elastic Beanstalk ainda excluirá a versão de seus registros. No entanto, o pacote de origem é deixado no bucket de armazenamento do Elastic Beanstalk. A cota de versões da aplicação se aplica apenas a versões monitoradas pelo Elastic Beanstalk. Portanto, se for necessário, você poderá excluir versões para se manter dentro da cota, mas retenha todos os pacotes de origem no Amazon S3.

> **Note**
>
> A cota de versões da aplicação não se aplica a pacotes de origem, mas ainda assim pode haver cobranças do Amazon S3 e reter informações pessoais além do tempo necessário. O Elastic Beanstalk nunca exclui pacotes de origem automaticamente. Você deve excluir os pacotes de origem quando não precisar mais deles.

Definir as configurações de ciclo de vida da versão do aplicativo

Este tópico explica as políticas e cotas que o Elastic Beanstalk aplica às versões da sua aplicação em um determinado ambiente, incluindo por quanto tempo uma versão da aplicação permanece em um ambiente.

Cada vez que você faz upload de uma nova versão da sua aplicação com o console do Elastic Beanstalk ou a CLI do EB, o Elastic Beanstalk cria uma versão da aplicação. Se você não excluir as versões que não usa mais, acabará atingindo a cota de versões da aplicação e não poderá criar novas versões dela.

Você pode evitar atingir a cota aplicando uma política de ciclo de vida de versão do aplicativo aos aplicativos. Uma política de ciclo de vida instrui o Elastic Beanstalk a excluir versões da aplicação antigas ou excluí-las quando o total de versões de uma determinada aplicação excede um número especificado.

O Elastic Beanstalk aplica uma política de ciclo de vida de uma aplicação cada vez que uma nova versão da aplicação é aplicada e exclui até 100 versões cada vez que a política de ciclo de vida é aplicada. O Elastic Beanstalk exclui versões antigas depois de criar a nova versão e não considera a nova versão no número máximo de versões definido na política.

O Elastic Beanstalk não exclui versões da aplicação que estão sendo usadas por um ambiente nem versões da aplicação implantadas em ambientes que foram encerrados menos de dez semanas antes do acionamento da política.

A cota de versões do aplicativo é utilizada em todos os aplicativos em uma região. Se você tiver vários aplicativos, configure cada um com uma política de ciclo de vida adequada para evitar atingir a cota. Por exemplo, se você tiver 10 aplicações em uma região e a cota for 1.000 versões da aplicação, considere definir uma política de ciclo de vida com uma cota de 99 versões para todas as aplicações ou definir outros valores em cada aplicação, desde que o total seja inferior a 1.000 versões. O Elastic Beanstalk somente aplicará a política se a criação da versão da aplicação for bem-sucedida, portanto, se você já atingiu a cota, deverá excluir algumas versões manualmente antes de criar uma nova.

Por padrão, o Elastic Beanstalk deixa o pacote de origem da versão da aplicação no Amazon S3 para evitar a perda de dados. Você pode excluir esse pacote para economizar espaço.

Você pode definir as configurações do ciclo de vida por meio da CLI do Elastic Beanstalk e. APIs Consulteeb appversion, CreateApplication(usando o `ResourceLifecycleConfig` parâmetro) e UpdateApplicationResourceLifecyclepara obter detalhes.

Definir as configurações de ciclo de vida do aplicativo no console

É possível especificar as configurações do ciclo de vida no console do Elastic Beanstalk.

Como especificar configurações de ciclo de vida do seu aplicativo

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Escolha Settings.
5. Use o formulário na tela para definir as configurações do ciclo de vida do aplicativo.
6. Escolha Salvar.

Na página de configurações, você pode fazer o seguinte:

- Defina as configurações do ciclo de vida com base na contagem total de versões do aplicativo ou na idade das versões do aplicativo.
- Especifique se deseja excluir o pacote de origem do S3 quando a versão do aplicativo for excluída.
- Especifique a função de serviço na qual a versão do aplicativo é excluída. Para incluir todas as permissões necessárias para a exclusão da versão, escolha a função de serviço padrão do Elastic Beanstalk, denominada `aws-elasticbeanstalk-service-role`, ou outra função de serviço usando as políticas de serviço gerenciadas do Elastic Beanstalk. Para obter mais informações, consulte Gerenciar funções de serviço do Elastic Beanstalk.

Atribuir tags em versões do aplicativo

Este tópico explica os benefícios de marcar as versões da aplicação Elastic Beanstalk e como gerenciar as etiquetas.

Você pode aplicar tags às versões do seu AWS Elastic Beanstalk aplicativo. As tags são pares de valores-chave associados AWS aos recursos. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte Marcar recursos da aplicação do Elastic Beanstalk.

Você pode especificar tags quando criar uma versão do aplicativo. Em uma versão de aplicativo existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada versão do aplicativo.

Adicionar tags durante a criação da versão do aplicativo

Ao usar o console do Elastic Beanstalk para criar um ambiente e optar por fazer upload de uma versão do código da aplicação, você pode especificar chaves e valores de tags para associar à nova versão da aplicação.

Também é possível usar o console do Elastic Beanstalk para fazer upload de uma versão da aplicação sem usá-lo imediatamente em um ambiente. Você pode especificar valores e chaves de tag ao carregar uma versão do aplicativo.

Com o AWS CLI ou outros clientes baseados em API, adicione tags usando o `--tags` parâmetro no create-application-version comando.

```
$ aws elasticbeanstalk create-application-version \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --version-label v1
```

Quando você usa a CLI do EB para criar ou atualizar um ambiente, uma versão do aplicativo é criada a partir do código implementado. Não há uma maneira direta de atribuir uma tag a uma versão do aplicativo durante sua criação por meio da CLI do EB. Consulte a seção a seguir para saber como adicionar tags a uma versão do aplicativo existente.

Gerenciar tags de uma versão do aplicativo existente

É possível adicionar, atualizar e excluir tags em uma versão existente da aplicação do Elastic Beanstalk.

Como gerenciar as tags de uma versão da aplicação usando o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Selecione a versão do aplicativo que você deseja gerenciar.

5. Escolha Actions (Ações) e Manage tags (Gerenciar tags).
6. Use o formulário na tela para adicionar, atualizar ou excluir tags.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar a versão do aplicativo, use eb tags para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em uma versão do aplicativo.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

O comando a seguir atualiza a tag `mytag1` e exclui a 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"
```

Para obter uma lista de opções e mais exemplos, consulte `eb tags`.

Com o AWS CLI ou outros clientes baseados em API, use o list-tags-for-resource comando para listar as tags de uma versão do aplicativo.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:applicationversion/my-app/my-version"
```

Use o comando update-tags-for-resource para adicionar, atualizar ou excluir tags em uma versão do aplicação.

```
$ 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"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do update-tags-for-resource. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

> **Note**
>
> Para usar alguns AWS CLI comandos e CLI do EB com uma versão do aplicativo Elastic Beanstalk, você precisa do ARN da versão do aplicativo. Você pode recuperar o ARN usando o comando a seguir.
>
> ```
> $ aws elasticbeanstalk describe-application-versions --application-name my-app
>  --version-label my-version
> ```

Crie um pacote de origem da aplicação Elastic Beanstalk

Este tópico explica como carregar arquivos de origem da sua aplicação para o Elastic Beanstalk em um pacote de origem. Ele explica os requisitos de um pacote de origem, sua estrutura e as abordagens para criá-lo.

Ao usar o AWS Elastic Beanstalk console para implantar um novo aplicativo ou uma versão do aplicativo, você precisará carregar os arquivos do aplicativo em um pacote de origem. O pacote de origem deve cumprir os seguintes requisitos:

- Consistir em um único arquivo `ZIP` ou `WAR` (você pode incluir vários arquivos `WAR` em seu arquivo `ZIP`)
- Não exceder 500 MB
- Não incluir uma pasta pai ou o diretório de nível superior (subdiretórios podem ser usados)

Se você deseja implantar um aplicativo do operador que processa tarefas periódicas em segundo plano, o pacote de origem do aplicativo deverá incluir também um arquivo `cron.yaml`. Para obter mais informações, consulte Tarefas periódicas.

Se você estiver implantando seu aplicativo com a Interface de Linha de Comando do Elastic Beanstalk (EB CLI), o AWS Toolkit for Eclipse ou o AWS Toolkit for Visual Studio, o arquivo ZIP ou WAR será automaticamente estruturado corretamente. Para ter mais informações, consulte Configuração da interface de linha de comando do EB (EB CLI) para gerenciar o Elastic Beanstalk, Implantação de aplicações Java com o Elastic Beanstalk e O AWS Toolkit for Visual Studio.

Seções

- Criar pacote de origem a partir da linha de comando

- Criar pacote de origem com o Git
- Compactar arquivos no Mac OS X Finder ou Windows Explorer
- Criar pacote de origem para um aplicativo .NET
- Testar pacote de origem

Criar pacote de origem a partir da linha de comando

Crie um pacote de origem usando o comando `zip`. Para incluir arquivos e pastas ocultos, use um padrão, como os seguintes.

```
~/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%)
```

Isso garante que os arquivos de configuração do Elastic Beanstalk e outros arquivos e pastas que começam com um ponto sejam incluídos no arquivo.

Para aplicativos web Tomcat, use `jar` para criar um arquivamento web.

```
~/myapp$ jar -cvf myapp.war .
```

Os comandos acima incluem arquivos ocultos que podem aumentar o tamanho do pacote de origem desnecessariamente. Para obter mais controle, use um padrão de arquivo mais detalhado ou crie seu pacote de origem com o Git.

Criar pacote de origem com o Git

Se você estiver usando o Git para gerenciar o código-fonte do aplicativo, use o comando `git archive` para criar o pacote de origem.

```
$ git archive -v -o myapp.zip --format=zip HEAD
```

`git archive` inclui apenas os arquivos que são armazenados em git e exclui arquivos ignorados e git. Isso ajuda a manter o pacote de origem o menor possível. Para obter mais informações, acesse a página do manual git-archive.

Compactar arquivos no Mac OS X Finder ou Windows Explorer

Quando você criar um arquivo `ZIP` no Mac OS X Finder ou Windows Explorer, certifique-se de compactar os arquivos e subpastas, em vez da pasta pai.

> **Note**
>
> A interface gráfica de usuário (GUI) nos sistemas operacionais baseados em Mac OS X e Linux não exibe os arquivos e pastas com nomes que começam com um ponto (.). Use a linha de comando em vez da GUI para compactar o aplicativo se o arquivo `ZIP` deve incluir uma pasta oculta, como `.ebextensions`. Para os procedimentos de linha de comando para criar um arquivo `ZIP` no sistema operacional baseado em Mac OS X ou Linux, consulte Criar pacote de origem a partir da linha de comando.

Example

Suponhamos que você tenha uma pasta de projeto Python chamada `myapp`, que inclui os seguintes arquivos e subpastas:

```
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
```

Como observado na lista de requisitos acima, seu pacote de origem deve ser compactado sem uma pasta pai, para que a estrutura descompactada não inclua um diretório de nível superior extra. Neste exemplo, nenhuma pasta `myapp` deverá ser criada quando os arquivos forem descompactados (ou, na linha de comando, nenhum segmento `myapp` deverá ser adicionado aos caminhos de arquivo).

Este exemplo de estrutura de arquivos é usado em todo este tópico para ilustrar como compactar arquivos.

Criar pacote de origem para um aplicativo .NET

Se você usa o Visual Studio, pode usar a ferramenta de implantação incluída no AWS Toolkit for Visual Studio para implantar seu aplicativo.NET no Elastic Beanstalk. Para obter mais informações, consulte Implantar aplicações do Elastic Beanstalk em .NET usando a ferramenta de implantação.

Se você precisa criar manualmente um pacote de origem para o seu aplicativo .NET, não pode simplesmente criar um arquivo `ZIP` que contenha o diretório do projeto. É necessário criar um pacote de implantação da Web para seu projeto que seja adequado à implantação no Elastic Beanstalk. Há vários métodos que podem ser usados para criar um pacote de implantação:

- Crie o pacote de implantação usando o assistente Publicar na Web no Visual Studio. Para obter mais informações, acesse Como criar um pacote de implantação Web no Visual Studio.

  > **Important**
  >
  > Ao criar o pacote de implantação da Web, inicie o Nome do site com `Default Web Site`.

- Se tiver um projeto .NET, você poderá criar o pacote de implantação usando o comando msbuild, conforme exibido no exemplo a seguir.

  > **Important**
  >
  > O parâmetro `DeployIisAppPath` deve começar com `Default Web Site`.

  ```
  C:/> msbuild <web_app>.csproj /t:Package /p:DeployIisAppPath="Default Web Site"
  ```

- Se você tem um projeto de site, pode usar a ferramenta de implantação da Web do IIS para criar o pacote de implantação. Para obter mais informações, acesse Packaging and Restoring a Web site.

  > **Important**
  >
  > O parâmetro `apphostconfig` deve começar com `Default Web Site`.

Se estiver implantando vários aplicativos ou um aplicativo ASP.NET Core, coloque sua `.ebextensions` pasta na raiz do pacote de origem, lado a lado com os pacotes do aplicativo e arquivo manifesto:

```
~/workspace/source-bundle/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- AspNetCore101HelloWorld.zip
|-- AspNetCoreHelloWorld.zip
|-- aws-windows-deployment-manifest.json
`-- VS2015AspNetWebApiApp.zip
```

Testar pacote de origem

Convém testar seu pacote de origem localmente antes de fazer upload para o Elastic Beanstalk. Como o Elastic Beanstalk usa basicamente a linha de comando para extrair os arquivos, é melhor fazer os testes a partir da linha de comando em vez de usar uma ferramenta GUI.

Certifique-se de que os arquivos descompactados aparecem na mesma pasta que o próprio arquivo, não em uma nova pasta ou diretório de nível superior.

Usando o EB CLI com AWS CodeBuild

O AWS CodeBuild compila o código-fonte, roda testes de unidade e produz artefatos prontos para implantação. É possível usar o CodeBuild com a CLI do EB para automatizar a criação do aplicativo a partir de seu código-fonte. A criação do ambiente e cada implantação subsequente iniciam com uma etapa de criação e com a implantação do aplicativo resultante.

> **Note**
>
> Algumas regiões não oferecem CodeBuild. A integração entre o Elastic CodeBuild Beanstalk não funciona nessas regiões.
> Para obter informações sobre os AWS serviços oferecidos em cada região, consulte a Tabela de regiões.

Como criar uma aplicação do

Para criar um aplicativo do Elastic Beanstalk que usa CodeBuild

1. Inclua um arquivo de especificação de CodeBuild compilação `buildspec.yml`,, na pasta do aplicativo.
2. Adicione uma entrada `eb_codebuild_settings` com opções específicas do Elastic Beanstalk ao arquivo.
3. Execute eb init na pasta.

> **Note**
>
> Não use os caracteres ponto (`.`) ou espaço ( ) no nome do aplicativo ao usar o EB CLI com. CodeBuild

O Elastic Beanstalk CodeBuild estende o formato do arquivo de especificação de compilação para incluir as seguintes configurações adicionais:

```
eb_codebuild_settings:
  CodeBuildServiceRole: role-name
  ComputeType: size
  Image: image
  Timeout: minutes
```

`CodeBuildServiceRole`

O ARN ou o nome da função de serviço AWS Identity and Access Management (IAM) que CodeBuild pode ser usada para interagir com AWS serviços dependentes em seu nome. Este valor é obrigatório. Se você omiti-lo, qualquer comando eb create ou eb deploy subsequente apresentará falha.

Para saber mais sobre como criar uma função de serviço para CodeBuild, consulte Criar uma função CodeBuild de serviço no Guia AWS CodeBuild do usuário.

> **Note**
>
> Você também precisa de permissões para realizar ações CodeBuild por si só. A política de usuários gerenciados do AdministratorAccessElastic AWSElastic Beanstalk

- Beanstalk inclui todas as permissões de ação necessárias. CodeBuild Se você não estiver usando a política gerenciada, certifique-se de conceder as permissões a seguir na política de usuário.

```
"codebuild:CreateProject",
"codebuild:DeleteProject",
"codebuild:BatchGetBuilds",
"codebuild:StartBuild"
```

Para obter detalhes, consulte Gerenciar políticas de usuário do Elastic Beanstalk.

`ComputeType`

A quantidade de recursos usados pelo contêiner Docker no ambiente de CodeBuild compilação. Os valores válidos são BUILD_ GENERAL1 _SMALL, BUILD_ _MEDIUM e BUILD_ GENERAL1 _LARGE. GENERAL1

`Image`

O nome da imagem do Docker Hub ou do Amazon ECR CodeBuild usada para o ambiente de construção. Essa imagem do Docker deve conter todas as ferramentas e bibliotecas de tempo de execução necessárias para criar seu código e deve corresponder à plataforma de destino do seu aplicativo. CodeBuild gerencia e mantém um conjunto de imagens especificamente destinado a ser usado com o Elastic Beanstalk. É recomendado que você use uma delas. Para obter detalhes, consulte Imagens do Docker fornecidas por CodeBuild no Guia do AWS CodeBuild usuário.

O valor `Image` é opcional. Se você omiti-lo, o comando eb init tentará escolher uma imagem que melhor corresponde à plataforma de destino. Além disso, se você executar eb init no modo interativo e ele falhar na escolha de uma imagem para você, será solicitado que você escolha uma. Ao final de uma inicialização bem-sucedida, eb init grava a imagem escolhida no arquivo `buildspec.yml`.

`Timeout`

A duração, em minutos, da execução da CodeBuild compilação antes do tempo limite. Este valor é opcional. Para obter detalhes sobre valores válidos e padrão, consulte Criar um projeto de compilação em CodeBuild.

> **Note**
>
> Esse tempo limite controla a duração máxima de uma CodeBuild execução, e a CLI do EB também o respeita como parte de sua primeira etapa para criar uma versão do aplicativo. É diferente do valor que você pode especificar na opção `--timeout` dos comandos eb create ou eb deploy. O último valor controla a duração máxima que a CLI do EB aguarda para a criação ou atualização do ambiente.

Criar e implantar o código do aplicativo

Sempre que o código do aplicativo precisa ser implantado, a CLI do EB CodeBuild usa para executar uma compilação e, em seguida, implanta os artefatos de compilação resultantes em seu ambiente. Isso ocorre ao criar um ambiente do Elastic Beanstalk para a aplicação usando o comando eb create e sempre que você implantar alterações no código no ambiente usando o comando eb deploy.

Se a CodeBuild etapa falhar, a criação ou implantação do ambiente não será iniciada.

Marcar aplicativos

Este tópico explica os benefícios de marcar aplicações Elastic Beanstalk. Também fornece instruções para criar e gerenciar etiquetas de aplicações. As tags são pares de valores-chave associados AWS aos recursos. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte Marcar recursos da aplicação do Elastic Beanstalk.

Você pode especificar tags quando cria um aplicativo. Em um aplicativo existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada aplicativo.

Adicionar tags durante a criação do aplicativo

Ao usar o console do Elastic Beanstalk para criar uma aplicação, é possível especificar chaves e valores de tags na caixa de diálogo Create New Application (Criar nova aplicação).

Se você usar a CLI do EB para criar um aplicativo, use a opção `--tags` com eb init para adicionar tags.

```
~/workspace/my-app$ eb init --tags mytag1=value1,mytag2=value2
```

Com o AWS CLI ou outros clientes baseados em API, adicione tags usando o `--tags` parâmetro no create-application comando.

```
$ aws elasticbeanstalk create-application \
      --tags Key=mytag1,Value=value1 Key=mytag2,Value=value2 \
      --application-name my-app --version-label v1
```

Gerenciar tags de um aplicativo existente

É possível adicionar, atualizar e excluir tags em uma aplicação existente do Elastic Beanstalk.

Como gerenciar as tags de uma aplicação no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. Escolha Actions (Ações) e Manage tags (Gerenciar tags).
4. Use o formulário na tela para adicionar, atualizar ou excluir tags.
5. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar o aplicativo, use eb tags para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em um aplicativo.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

O comando a seguir atualiza a tag `mytag1` e exclui a 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"
```

Para obter uma lista de opções e mais exemplos, consulte `eb tags`.

Com o AWS CLI ou outros clientes baseados em API, use o list-tags-for-resource comando para listar as tags de um aplicativo.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:application/my-app"
```

Use o comando update-tags-for-resource para adicionar, atualizar ou excluir tags em uma aplicação.

```
$ 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"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do update-tags-for-resource. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

> **Note**
>
> Para usar alguns AWS CLI comandos e CLI do EB com um aplicativo do Elastic Beanstalk, você precisa do ARN do aplicativo. Você pode recuperar o ARN usando o comando a seguir.
>
> ```
> $ aws elasticbeanstalk describe-applications --application-names my-app
> ```

Marcar recursos da aplicação do Elastic Beanstalk

Este tópico explica os benefícios de usar etiquetas com os recursos da aplicação Elastic Beanstalk, juntamente com as restrições de fazê-lo. Ele também explica como criar e gerenciar etiquetas para recursos da aplicação.

Você pode aplicar tags aos recursos de seus AWS Elastic Beanstalk aplicativos. As tags são pares de valores-chave associados AWS aos recursos. As tags podem ajudar você a classificar recursos. Eles são particularmente úteis se você gerencia muitos recursos como parte de vários AWS aplicativos.

Veja a seguir algumas maneiras de usar tags com recursos do Elastic Beanstalk:

- Estágios de implantação: identifique recursos associados a diferentes estágios da aplicação, como desenvolvimento, beta e produção.
- Alocação de custos — Use relatórios de alocação de custos para monitorar o uso de AWS recursos associados a várias contas de despesas. Os relatórios incluem recursos marcados

e não marcados e agregam os custos de acordo com as tags. Para obter informações sobre como os relatórios de alocação de custo usam etiquetas, consulte Usar etiquetas de alocação de custos para relatórios de faturamento personalizados no Guia do usuário do AWS Billing and Cost Management.

- Controle de acesso: use etiquetas para gerenciar permissões para solicitações e recursos. Por exemplo, um usuário que só pode criar e gerenciar ambientes beta deve ter acesso somente aos recursos do estágio beta. Para obter detalhes, consulte Usar tags para controlar o acesso aos recursos do Elastic Beanstalk.

Você pode adicionar até 50 tags a cada recurso. Os ambientes são um pouco diferentes: o Elastic Beanstalk adiciona três tags de sistema padrão aos ambientes, e não é possível editar nem excluir essas tags. Além das tags padrão, você pode adicionar até 47 tags adicionais a cada ambiente.

As seguintes restrições se aplicam a chaves e valores de tags:

- As chaves e os valores podem conter letras, números, espaço em branco e os seguintes símbolos: `_ . : / = + - @`
- As chaves podem conter até 127 caracteres. Os valores podem conter até 255 caracteres.

> **Note**
>
> Esses limites de tamanho são para caracteres Unicode em UTF-8. Para outras codificações multibyte, os limites podem ser menores.

- As chaves fazem distinção entre maiúsculas e minúsculas.
- As chaves de tag não podem começar com `aws:` ou `elasticbeanstalk:`.

Recursos que podem ser marcados

Veja a seguir os tipos de recursos do Elastic Beanstalk que podem ser marcados com tags, bem como os links para tópicos específicos sobre gerenciamento de tags para cada um deles:

- Aplicativos
- Ambientes do
- Versões do aplicativo
- Configurações salvas

Propagação de tags para modelos de execução

O Elastic Beanstalk fornece uma opção para permitir a propagação de tags de ambiente para modelos de execução. Essa opção fornece suporte contínuo ao controle de acesso baseado em tags (TBAC) com modelos de execução.

> **Note**
>
> As configurações de lançamento estão sendo descontinuadas e substituídas por modelos de execução. Para obter mais informações, consulte Configurações de lançamento no Guia do usuário do Amazon EC2 Auto Scaling.

Para evitar o tempo de inatividade das EC2 instâncias em execução, AWS CloudFormation não propaga tags para os modelos de execução existentes. Se houver um caso de uso que exija tags para os recursos do seu ambiente, você poderá habilitar o Elastic Beanstalk para criar modelos de execução com tags para esses recursos. Para fazer isso, defina a opção `LaunchTemplateTagPropagationEnabled` no namespace aws:autoscaling:launchconfiguration como `true`. O valor padrão é `false`.

O exemplo de arquivo de configuração a seguir permite a propagação de tags para modelos de execução.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    LaunchTemplateTagPropagationEnabled: true
```

O Elastic Beanstalk só pode propagar tags para modelos de execução para os seguintes recursos:

- Volumes do EBS
- EC2 instâncias
- EC2 interfaces de rede
- AWS CloudFormation modelos de lançamento que definem um recurso

Essa restrição existe porque CloudFormation só permite tags na criação de modelos para recursos específicos. Para obter mais informações, consulte TagSpecificationo Guia AWS CloudFormation do usuário.

Important

- Alterar o valor dessa opção de `false` para `true` para um ambiente existente pode ser uma alteração significativa para tags já existentes.
- Quando esse recurso estiver ativado, a propagação das tags precisará ser EC2 substituída, o que pode resultar em tempo de inatividade. É possível habilitar atualizações contínuas para aplicar alterações de configuração em lotes e evitar tempo de inatividade durante o processo de atualização. Para obter mais informações, consulte Alterações de configuração.

Para obter mais informações sobre modelos de execução, consulte:

- Modelos de lançamento no Guia do usuário do Amazon EC2 Auto Scaling
- Trabalhar com modelos no Guia do usuário do AWS CloudFormation
- Trechos de modelos do Elastic Beanstalk no Guia do usuário do AWS CloudFormation

Criação de ambientes no Elastic Beanstalk

Este capítulo descreve como criar e gerenciar seus ambientes do Elastic Beanstalk. Esta página introdutória fornece uma visão geral das atualizações, manutenções e configurações que você aplicará ao longo do tempo, conforme a aplicação e o ambiente evoluem.

Funções do ambiente

Você pode criar e gerenciar ambientes separados para uso em desenvolvimento, teste e produção, e pode implantar qualquer versão do seu aplicativo em qualquer ambiente. Os ambientes podem ser de execução prolongada ou temporários. Quando você encerra um ambiente, pode salvar sua configuração para recriá-lo posteriormente.

Implantações de aplicações

À medida que desenvolve seu aplicativo, você o implanta com frequência, possivelmente em muitos ambientes diferentes para diversas finalidades. O Elastic Beanstalk permite configurar como as implantações são executadas. Você pode fazer a implantação em todas as instâncias do seu ambiente simultaneamente ou dividi-la em lotes com implantações contínuas.

Alterações de configuração

As alterações de configuração são processados separadamente das implantações e têm escopo próprio. Por exemplo, se você alterar o tipo das EC2 instâncias que executam seu aplicativo, todas as instâncias deverão ser substituídas. Por outro lado, se você modificar a configuração do load balancer do ambiente, essa alteração poderá ser feita no local, sem interromper o serviço nem reduzir a capacidade. Você também pode aplicar alterações de configuração em lotes, que modificam as instâncias em seu ambiente com atualizações de configuração contínuas.

> **Note**
>
> Modifique os recursos no ambiente usando somente o Elastic Beanstalk. Se você modificar recursos usando o console de outro serviço, comandos CLI ou SDKs o Elastic Beanstalk não conseguirá monitorar com precisão o estado desses recursos e você não poderá salvar a configuração ou recriar o ambiente de forma confiável. Alterações fora de banda também podem causar problemas ao atualizar ou encerrar um ambiente.

Atualizações da plataforma

Ao iniciar um ambiente, você escolhe uma versão de plataforma. Nós atualizamos as plataformas periodicamente com novas versões de plataforma para aprimorar o desempenho e lançar novos recursos. Você pode atualizar seu ambiente para a versão de plataforma mais recente a qualquer momento. Consulte o guia Plataformas do AWS Elastic Beanstalk para uma lista de plataformas compatíveis e um histórico de versões da plataforma que inclui os intervalos de datas em que estavam atualizados.

Opções de arquitetura

À medida que a complexidade de seu aplicativo aumenta, você pode dividi-lo em vários componentes, sendo cada um executado em um ambiente separado. Para cargas de trabalho de longa duração, é possível executar ambientes de operador que processam trabalhos a partir de uma fila do Amazon Simple Queue Service (Amazon SQS).

Tópicos

- Usar o console de gerenciamento de ambiente do Elastic Beanstalk
- Criar um ambiente do Elastic Beanstalk
- Gerenciar vários ambientes do Elastic Beanstalk como um grupo com a CLI do EB
- Implantar aplicações em ambientes do Elastic Beanstalk
- Alterações de configuração
- Atualizar a versão de plataforma do ambiente Elastic Beanstalk
- Cancelar atualizações de configuração do ambiente e implantações de aplicativos
- Recriar ambientes do Elastic Beanstalk
- Tipos de ambiente
- Ambientes de operador do Elastic Beanstalk
- Criar links entre ambientes do Elastic Beanstalk
- Recuperando seu ambiente do Elastic Beanstalk de um estado inválido

Usar o console de gerenciamento de ambiente do Elastic Beanstalk

Esta seção descreve gerenciar um ambiente Elastic Beanstalk usando o console de gerenciamento do ambiente. Esse console oferece a capacidade de gerenciar a configuração do ambiente e executar ações comuns. Essas ações incluem reiniciar os servidores Web em execução no seu ambiente, clonar seu ambiente ou reconstruir seu ambiente a partir do zero.

Tópicos

- Acessar o console de gerenciamento do ambiente
- Painel de visão geral do ambiente
- Detalhe do ambiente
- Ações do ambiente

Acessar o console de gerenciamento do ambiente

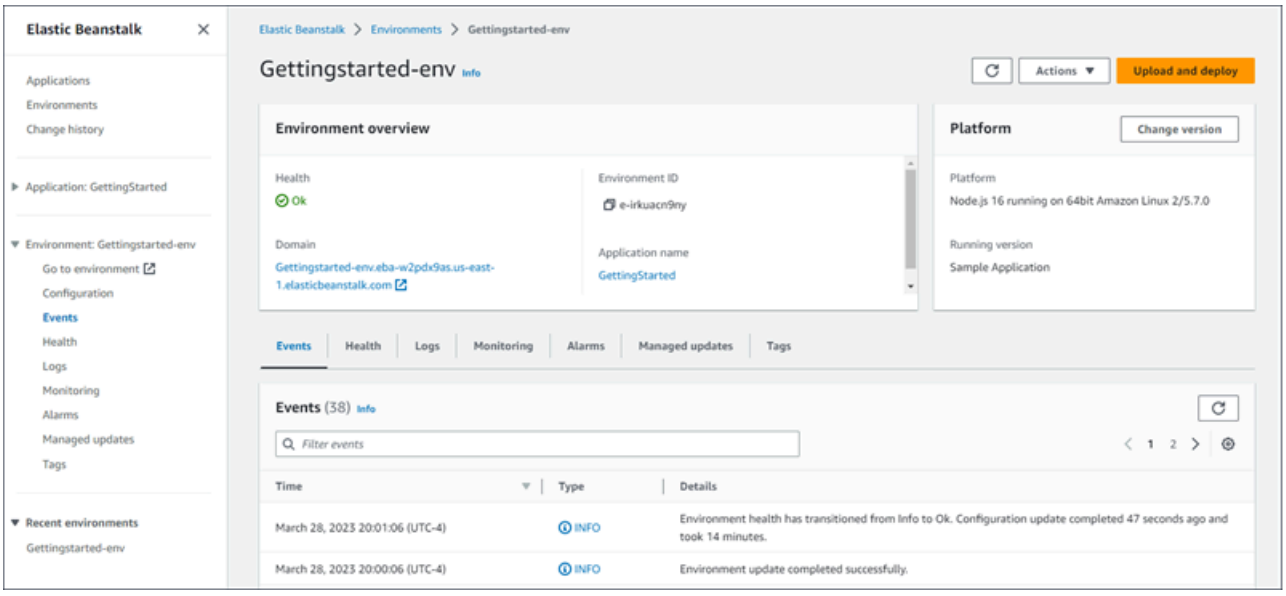
O procedimento a seguir fornece etapas para iniciar o console de gerenciamento do ambiente.

Se já estiver conectado ao console do Elastic Beanstalk, você também poderá iniciar a página de gerenciamento de ambiente no Console de gerenciamento de aplicativos. Selecione um ambiente na lista para exibir os detalhes do console de gerenciamento do ambiente selecionado.

Como acessar o console de gerenciamento do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

A imagem a seguir ilustra o console de gerenciamento do ambiente.



O painel superior é a página Visão geral do ambiente. Ele mostra informações de nível superior sobre o ambiente.

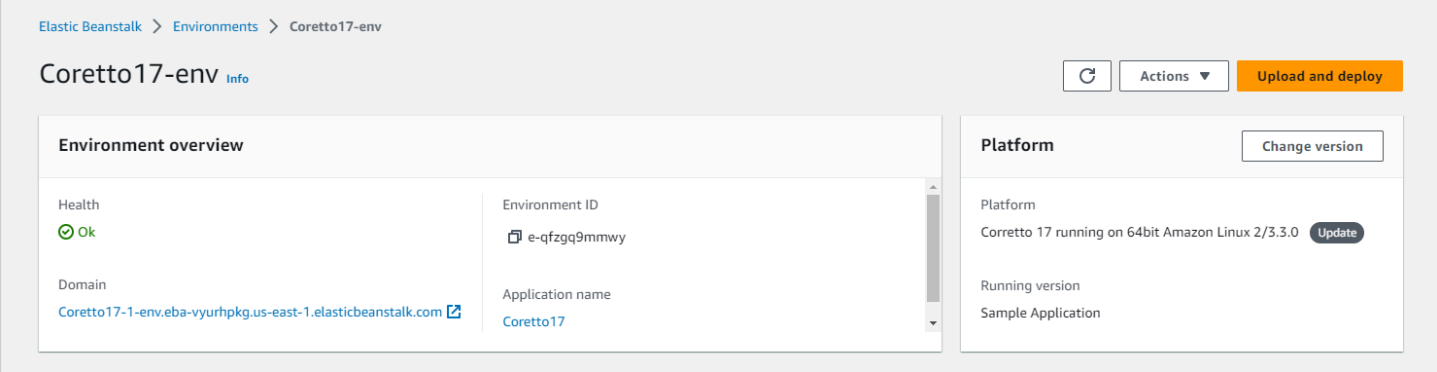
A metade inferior da página mostra guias que fornecem informações mais detalhadas. A guia Eventos está visível por padrão. As páginas vinculadas às guias também são listadas no painel de navegação esquerdo, abaixo do ambiente.

O painel de navegação do console mostra o nome da aplicação implantada no ambiente, com páginas de gerenciamento de aplicações relacionadas. O nome do ambiente também aparece na página de navegação, seguido pelas páginas de gerenciamento do ambiente. Os links sob o nome do ambiente também incluem Ir para o ambiente e Configuração, além das páginas com guias mencionadas anteriormente.

Painel de visão geral do ambiente

Este tópico descreve as informações fornecidas pelo painel Visão geral do ambiente. Ele mostra informações de nível superior sobre o ambiente e está localizado na metade superior do console de gerenciamento do ambiente.

A imagem a seguir mostra o painel Visão geral do ambiente.



Integridade

A integridade geral do ambiente. Se a integridade do seu ambiente piorar, o link Exibir causas será exibido ao lado da integridade do ambiente. Selecione este link para ver a guia Integridade com mais detalhes.

Domínio

O Domain (Domínio), ou URL, do ambiente encontra-se na parte superior da página Environment overview (Visão geral do ambiente), abaixo de Health (Integridade) do ambiente. Este é o URL do aplicativo web no qual o ambiente é executado. Você pode iniciar a aplicação selecionando o URL.

ID do ambiente

O ID do ambiente. Esse é um ID interno que é gerado quando o ambiente é criado.

Nome da aplicação

O nome da versão da aplicação que está implantada e em execução no ambiente.

Versão em execução

O nome da versão do aplicativo que está implantado e em execução em seu ambiente. Escolha Upload and deploy (Fazer upload e implantar) para fazer upload de um pacote de origem e implantá-lo em seu ambiente. Essa opção cria uma nova versão do aplicativo.

Plataforma

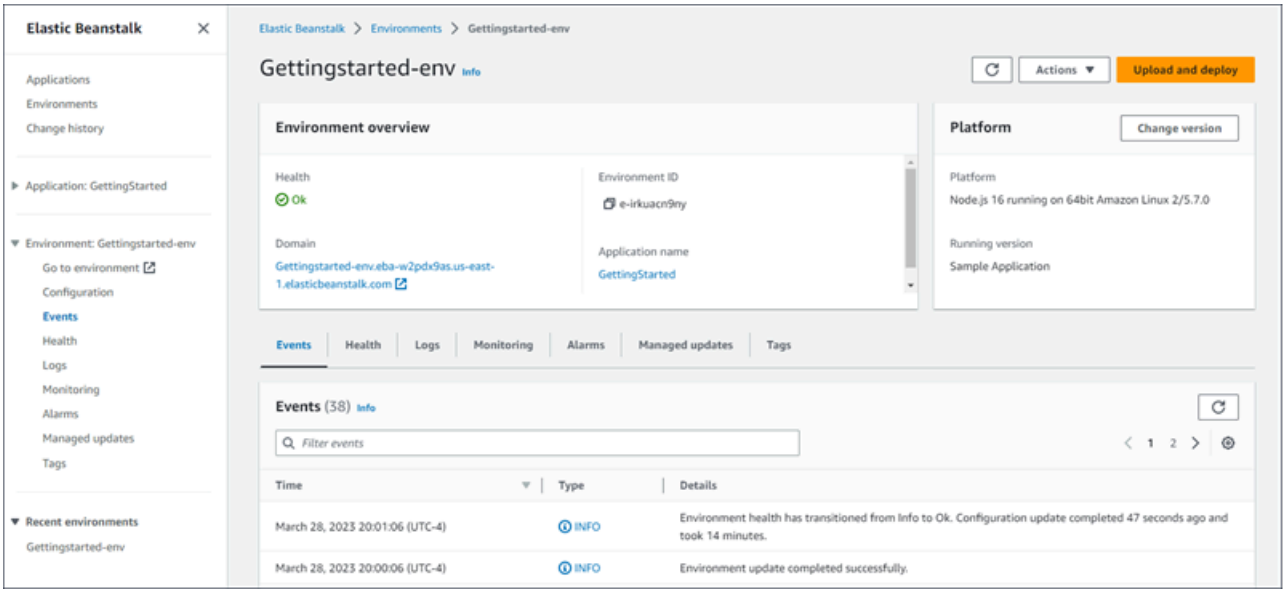
O nome da versão da plataforma em execução no seu ambiente. Normalmente, isso inclui a arquitetura, o sistema operacional (SO), a linguagem e o servidor de aplicações (coletivamente conhecidos como ramificação da plataforma), com um número de versão específico da plataforma.

Se a versão da plataforma não for a mais recente disponível, um rótulo de status será exibida ao lado dela na seção Plataforma. O rótulo Atualizar indica que, embora a versão da plataforma seja compatível, uma versão mais recente está disponível. A versão da plataforma também pode ser rotulada como Obsoleta ou Descontinuada. Selecione Alterar versão para atualizar sua ramificação da plataforma para uma versão mais recente. Para obter mais informações sobre os estados de uma versão da plataforma, consulte a seção Ramificação da plataforma noGlossário de plataformas do Elastic Beanstalk. A imagem anterior nesta página ilustra o rótulo de status Atualizar da plataforma fornecida.

Detalhe do ambiente

Este tópico descreve as informações adicionais que fornecidas pelo console de gerenciamento do ambiente no painel de navegação esquerdo e nas páginas com guias.

A imagem a seguir ilustra o console de gerenciamento do ambiente.



A metade inferior do console de gerenciamento de ambiente inclui guias que fornecem informações mais detalhadas e variadas sobre o ambiente. É possível selecionar a página da guia ou o rótulo da página no painel de navegação esquerdo.

No painel de navegação esquerdo do console, abaixo do nome do ambiente, há duas opções que não estão nas páginas com guias. Estas são Ir para o ambiente e Configuração.

> **Note**
>
> Selecione Ir para o ambiente para iniciar a aplicação.

Configuração

Use a página Configuração no painel de navegação esquerdo para visualizar e atualizar as configurações atuais do ambiente e seus recursos. Isso inclui configuração de rede, configuração de banco de dados, balanceamento de carga, notificações, configurações de monitoramento de integridade, configuração de atualização de plataforma gerenciada, configuração de implantação, streaming de registros de instâncias AWS X-Ray, CloudWatch integração, configurações do servidor proxy, propriedades do ambiente e opções específicas da plataforma. Use as configurações nessa página para personalizar o comportamento do seu ambiente durante as implantações, habilitar recursos adicionais e modificar o tipo de instância e outras configurações que você escolher durante a criação do ambiente.

Para obter mais informações, consulte Configurar ambientes do Elastic Beanstalk.

Eventos

A página Events (Eventos) exibe o stream de eventos do seu ambiente. O Elastic Beanstalk envia mensagens de evento sempre que você interage com o ambiente e quando recursos do ambiente são criados ou modificados como resultado.

Para obter mais informações, consulte Visualizar o stream de eventos de um ambiente do Elastic Beanstalk.

Integridade

Se o monitoramento aprimorado da integridade estiver ativado, esta página EC2 listará as instâncias em seu ambiente e as informações de saúde ativas de cada instância.

A página Integridade geral mostra os dados de integridade como uma média de todas as instâncias do ambiente combinadas.

O painel de integridade da instância aprimorada mostra informações de saúde ao vivo para cada EC2 instância individual em seu ambiente. O monitoramento de integridade avançado permite que o Elastic Beanstalk monitore de perto os recursos no ambiente para que possa avaliar a integridade de sua aplicação com mais precisão.

Quando o monitoramento avançado de integridade está ativado, essa página mostra informações sobre as solicitações atendidas pelas instâncias em seu ambiente e as métricas do sistema operacional, incluindo latência, carga e utilização de CPU.

Para obter mais informações, consulte Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk.

Logs

A página Registros permite que você recupere registros das EC2 instâncias em seu ambiente. Quando logs são solicitados, o Elastic Beanstalk envia um comando para as instâncias que, depois, faz upload de logs para o bucket de armazenamento do Elastic Beanstalk no Amazon S3. Quando logs são solicitados nesta página, o Elastic Beanstalk os exclui automaticamente do Amazon S3 após 15 minutos.

Também é possível configurar as instâncias do seu ambiente para fazer upload dos logs para o Amazon S3, para armazenamento permanente depois que forem alternadas localmente.

Para obter mais informações, consulte Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

Monitoramento

A página Monitoring (Monitoramento) exibe uma visão geral de informações de integridade de seu ambiente. Isso inclui o conjunto padrão de métricas fornecido pelo Elastic Load Balancing e pela Amazon EC2, além de gráficos que mostram como a saúde do ambiente mudou ao longo do tempo.

Para obter mais informações, consulte Monitorando a integridade do ambiente no console AWS de gerenciamento.

Alarmes

A página Existing alarms (Alarmes existentes) exibe informações sobre todos os alarmes que você configurou em seu ambiente. Você pode usar as opções dessa página para modificar ou excluir alarmes.

Para obter mais informações, consulte Gerenciar alarmes.

Atualizações gerenciadas

A página Atualizações gerenciadas exibe informações sobre as atualizações gerenciadas da Plataforma concluídas ou futuras e a substituição de instâncias.

O recurso de atualizações gerenciadas permite que você configure o ambiente para ser atualizado automaticamente com a última versão da plataforma durante uma janela de manutenção semanal de sua escolha. Entre os lançamentos da plataforma, você pode escolher que seu ambiente substitua todas as EC2 instâncias da Amazon durante a janela de manutenção. Isso pode diminuir os problemas que ocorrem quando a aplicação é executada por longos períodos.

Para obter mais informações, consulte Atualizações gerenciadas de plataforma.

Tags

A página Tags exibe as tags que o Elastic Beanstalk aplicou ao ambiente quando o criou e as tags adicionadas. É possível adicionar, editar e excluir tags personalizadas. Não é possível editar ou excluir as tags aplicadas pelo Elastic Beanstalk.

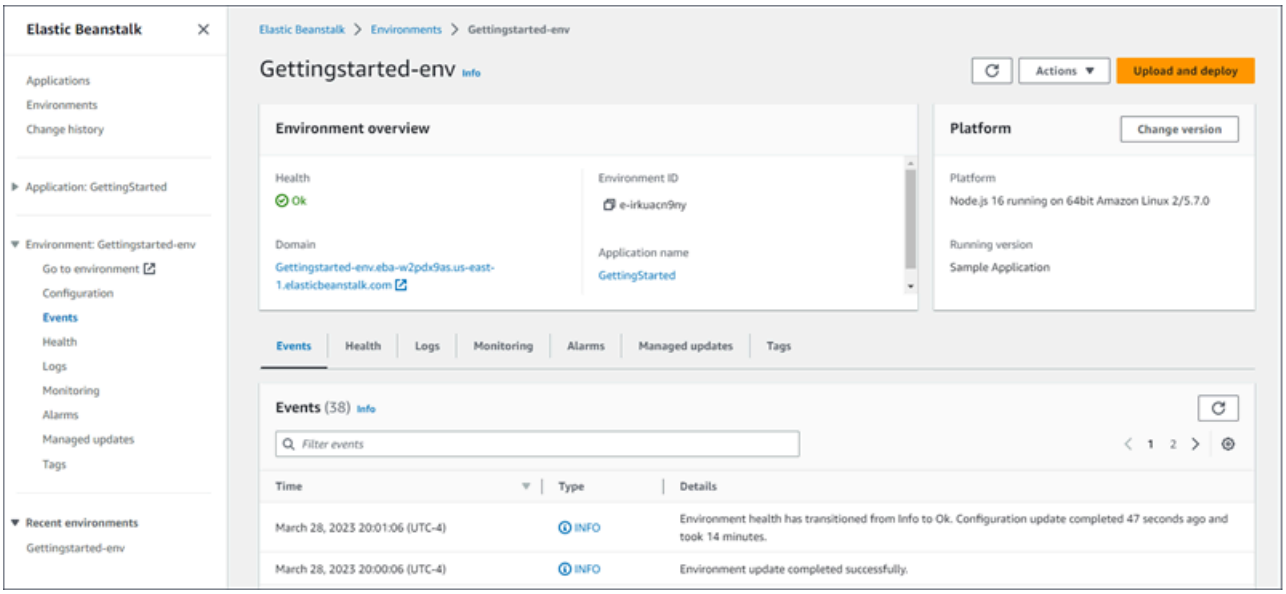
As tags do ambiente são aplicadas a todos os recursos que o Elastic Beanstalk cria para oferecer suporte à aplicação.

Para obter mais informações, consulte Marcar recursos em seus ambientes do Elastic Beanstalk.

Ações do ambiente

Este tópico descreve as operações comuns que você pode selecionar para executar no ambiente a partir do menu suspenso Ações no console de gerenciamento do ambiente.

A imagem a seguir ilustra o console de gerenciamento do ambiente. O menu suspenso Ações fica no lado direito do cabeçalho que mostra o nome do ambiente, ao lado do botão Atualizar.



> **Note**
>
> Algumas ações só estão disponíveis sob certas condições, permanecendo desativadas até que as condições certas sejam atendidas.

Carregar configuração

Carregue uma configuração salva anteriormente. As configurações são salvas no seu aplicativo e podem ser carregadas por qualquer ambiente associado. Se você fizer alterações na configuração do seu ambiente, poderá carregar uma configuração salva para desfazê-las. Você também pode carregar uma configuração que você salvou de um ambiente diferente que esteja executando o mesmo aplicativo para propagar as alterações de configuração entre eles.

Salvar configuração

Salve a configuração atual do seu ambiente no seu aplicativo. Antes de fazer alterações na configuração do ambiente, salve a configuração atual para que você possa reverter posteriormente, se necessário. Você também pode aplicar uma configuração salva ao iniciar um novo ambiente.

Domínios do ambiente de troca () URLs

Troque o CNAME do ambiente atual por um novo ambiente. Depois de uma troca de CNAME, todo o tráfego para o aplicativo que usa a URL do ambiente vai para o novo ambiente. Quando você estiver pronto para implantar uma nova versão de seu aplicativo, poderá iniciar um ambiente separado na nova versão. Quando o novo ambiente estiver pronto para começar a receber solicitações, execute uma troca de CNAME para começar a rotear o tráfego para o novo ambiente. Fazer isso não interrompe seus serviços. Para obter mais informações, consulte Implantações azuis/verdes com o Elastic Beanstalk.

Clonar ambiente

Inicie um novo ambiente com as mesmas configurações que o seu atual ambiente em execução.

Clonar com a plataforma mais recente

Clone o seu ambiente atual com a versão mais recente da plataforma do Elastic Beanstalk em uso. Essa opção só fica disponível quando uma versão mais recente da plataforma do ambiente atual está disponível para uso.

Abortar operação atual

Pare uma atualização de ambiente em andamento. Interromper uma operação pode fazer com que algumas das instâncias no seu ambiente fiquem em um estado diferente das outras, dependendo do progresso da operação. Essa opção está disponível apenas quando o ambiente está sendo atualizado.

Reiniciar servidores de aplicativos

Reinicie o servidor web em execução nas instâncias do seu ambiente. Essa opção não encerra nem reinicia nenhum AWS recurso. Se seu ambiente estiver agindo de modo estranho em resposta a solicitações ruins, reiniciar o servidor de aplicativo pode restaurar temporariamente a funcionalidade, enquanto você soluciona a causa principal do problema.

Recriar ambiente

Encerre todos os recursos no ambiente em execução e crie um novo ambiente com as mesmas configurações. Essa operação leva vários minutos, semelhante ao tempo necessário para implantar um novo ambiente a partir do zero. Todas as instâncias do Amazon RDS em execução na camada de dados de seu ambiente são excluídas durante a recriação. Se você precisa dos dados, crie um snapshot. É possível criar um snapshot manualmente no console do RDS ou configurar política de exclusão da camada de dados para criar um snapshot automaticamente antes de excluir a instância. Essa é a configuração padrão quando você cria uma camada de dados.

Encerrar ambiente

Encerre todos os recursos no ambiente em execução e remova o ambiente da aplicação. Se você tiver uma instância do RDS em execução em uma camada de dados e precisar reter os dados, certifique-se de que a política de exclusão de banco de dados esteja definida como `Snapshot` ou `Retain`. Para obter mais informações, consulte Ciclo de vida do banco de dados no capítulo Configuração de ambientes deste guia.

Criar um ambiente do Elastic Beanstalk

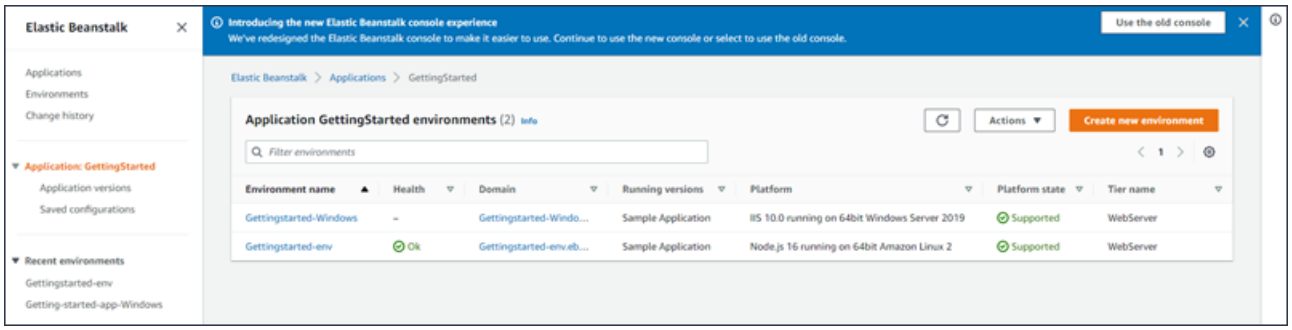
O procedimento a seguir inicia um novo ambiente executando o aplicativo padrão. Essas etapas são simplificadas para colocar seu ambiente em funcionamento rapidamente, usando valores padrão de opções.

> **Nota sobre permissões**
>
> A criação de um ambiente requer as permissões na política gerenciada de acesso completo do Elastic Beanstalk. Para mais detalhes, consulte Política de usuário do Elastic Beanstalk.

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Aplicativos. Selecione uma aplicação existente na lista. Também é possível optar por criar uma aplicação, seguindo as instruções em Como gerenciar aplicações do .
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).

A imagem a seguir mostra a página de visão geral da aplicação.



Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.

4. Em Nível de ambiente, escolha o nível de ambiente de Ambiente do servidor Web ou Ambiente do operador. Você não pode alterar o nível de um ambiente após a criação.

> **Note**
>
> O .NET na Plataforma de servidor Windows não tem suporte para a camada de ambiente do operador.

Os campos Informações sobre a aplicação são definidos por padrão, com base na aplicação que você escolheu anteriormente.

No grupo Informações do ambiente, o Nome do ambiente é definido por padrão com base no nome da aplicação. Se preferir um nome de ambiente diferente, insira outro valor no campo. Existe a opção de inserir um nome de Domínio. Se isso não for feito, o Elastic Beanstalk gerará automaticamente um valor. Você também pode opcionalmente inserir uma Descrição do ambiente.

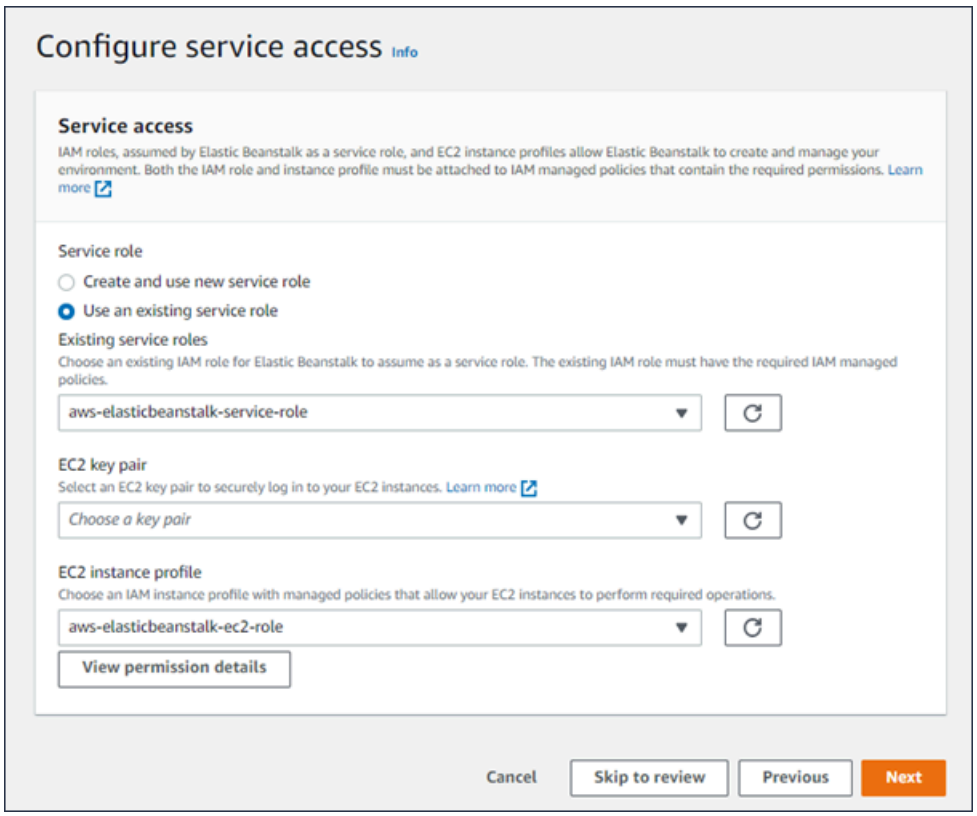
5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

> **Note**
>
> O Elastic Beanstalk é compatível com várias versões para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá

selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte the section called “Plataformas compatíveis”.

6. Em Código da aplicação, existem algumas opções para iniciar uma aplicação de exemplo.
   - Para iniciar a aplicação de exemplo padrão sem fornecer o código-fonte, escolha Aplicação de exemplo. Essa ação escolhe a aplicação de página única que o Elastic Beanstalk fornece para a plataforma selecionada anteriormente.
   - Se você baixou uma aplicação de exemplo deste guia ou de outra fonte, siga as etapas abaixo.
     a. Selecione Carregue seu código.
     b. Em seguida, escolha Arquivo local e, em Carregar aplicação, selecione Escolher arquivo.
     c. O sistema operacional do computador apresentará uma interface para selecionar o arquivo local baixado. Selecione o arquivo do pacote de origem e continue.
7. Em Predefinições, escolha Instância única.
8. Escolha Próximo.
9. A página Configurar acesso ao serviço é exibida.

   A imagem a seguir ilustra a página Configurar acesso ao serviço.



10. Escolha um valor do menu suspenso Perfis de serviço existentes.
11. (Opcional) Se você criou um par de EC2 chaves anteriormente, poderá selecioná-lo na lista suspensa do campo de par de EC2 chaves. Você o usaria para fazer login com segurança na EC2 instância da Amazon que o Elastic Beanstalk provisiona para seu aplicativo. Se você pular essa etapa, sempre poderá criar e atribuir um par de EC2 chaves após a criação do ambiente. Para obter mais informações, consulte EC2 key pair.
12. Em seguida, vamos nos concentrar na lista suspensa do perfil da EC2 instância. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

    Escolha um dos seguintes itens, com base nos valores exibidos na lista.

    - Se `aws-elasticbeanstalk-ec2-role` aparecer na lista suspensa, selecione-o.
    - Se outro valor for exibido na lista e for o perfil de EC2 instância padrão destinado aos seus ambientes, selecione-o na lista suspensa.
    - Se a lista suspensa do perfil da EC2 instância não listar nenhum valor, você precisará criar um perfil da instância.

> **criar um perfil de instância**
>
> Para criar um perfil de instância, faremos um desvio para outro procedimento nessa mesma página. Vá até o final desse procedimento e expanda o procedimento a seguir, Create IAM Role for EC2 instance profile.
> Conclua as etapas em Criar função do IAM para o perfil da EC2 instância para criar uma função do IAM que você possa selecionar posteriormente para o perfil da EC2 instância. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione a função do IAM que você acabou de criar na lista suspensa do perfil da EC2 instância.

13. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

    Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

14. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

    Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

    - Nome do ambiente
    - Nome de domínio
    - Versão da plataforma
    - Processador
    - VPC
    - Nível

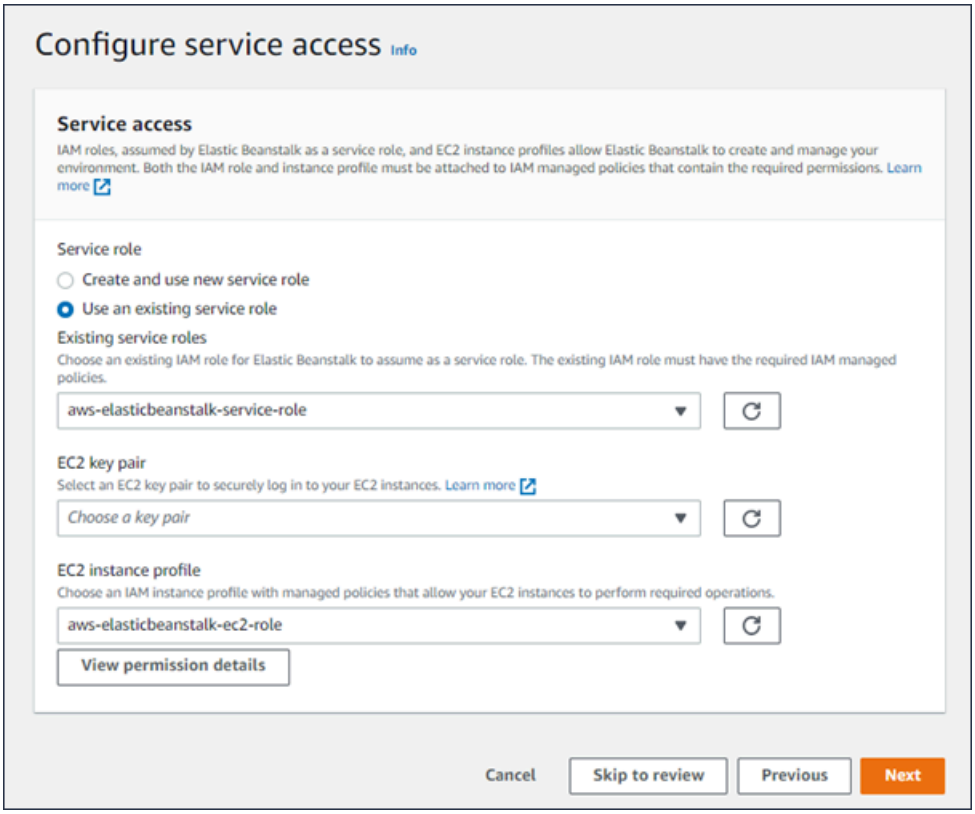
    Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte O assistente de criação de novo ambiente.

15. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

Crie uma função do IAM para o perfil da EC2 instância



Para criar o perfil da EC2 instância

1. Selecione Criar perfil.
2. Em Tipo de entidade confiável, escolha Serviços da AWS .
3. Para o caso de uso, escolha Elastic Beanstalk — Compute.
4. Escolha Próximo.

5. Verifique se as políticas de permissões incluem o seguinte e escolha Avançar:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
6. Selecione Criar perfil.
7. Volte para a guia Configurar acesso ao serviço, atualize a lista e selecione o perfil de EC2 instância recém-criado.

Enquanto o Elastic Beanstalk cria seu ambiente, você será redirecionado para o console do Elastic Beanstalk. Quando a integridade do ambiente ficar verde, escolha o URL próximo ao nome do ambiente para visualizar o aplicativo em execução. Esse URL costuma ser acessível pela internet, a menos que você configure seu ambiente para usar uma VPC personalizada com um load balancer interno.

Tópicos

- O assistente de criação de novo ambiente
- Clonar um ambiente do Elastic Beanstalk
- Encerrar um ambiente do Elastic Beanstalk
- Criação de ambientes do Elastic Beanstalk com a CLI da AWS
- Criar ambientes do Elastic Beanstalk com a API
- Construção de um URL "Iniciar agora"
- Criar e atualizar grupos de ambientes do Elastic Beanstalk

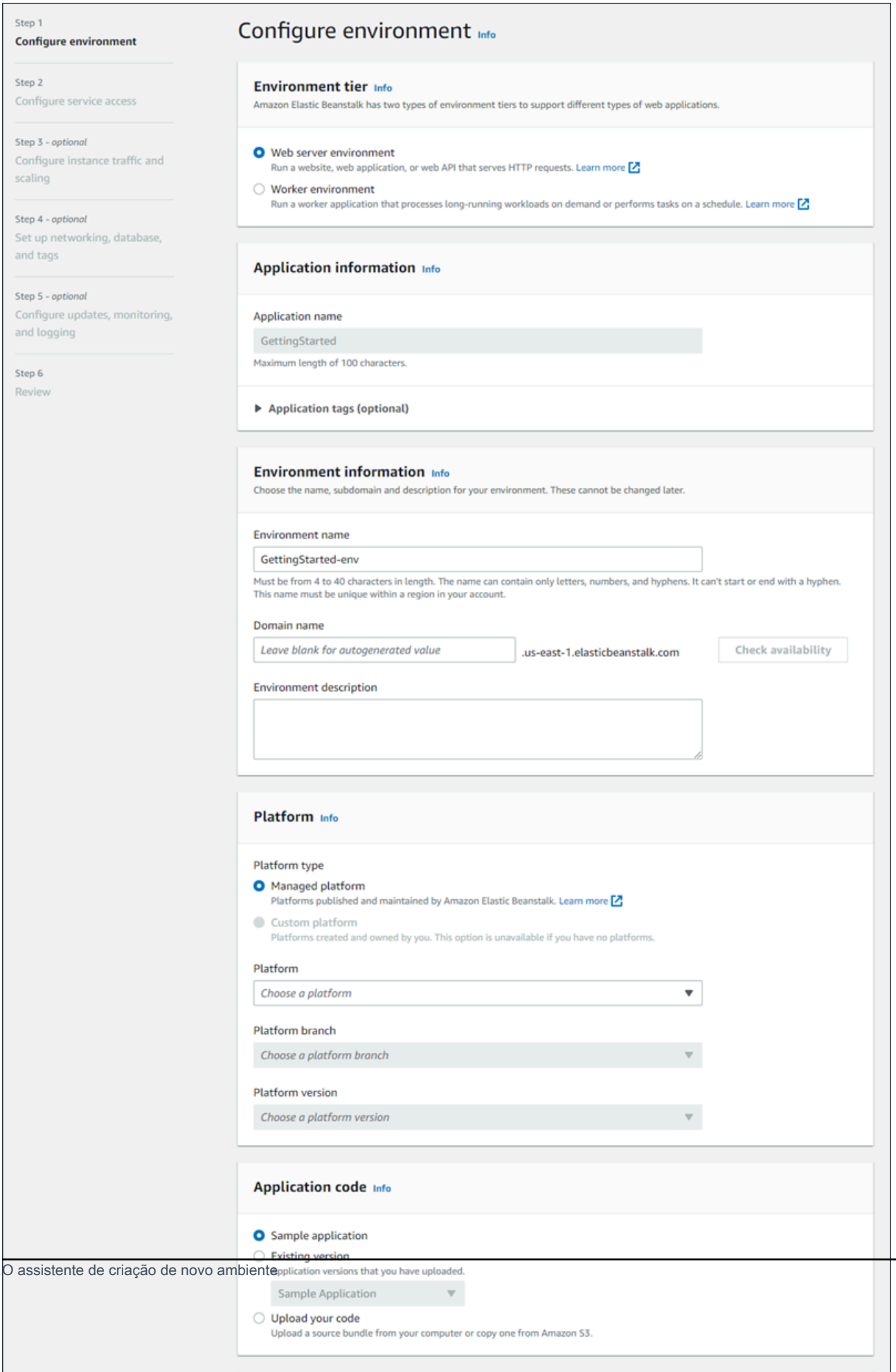
O assistente de criação de novo ambiente

Este tópico descreve o assistente Create new environment (Criar novo ambiente) e todas as maneiras de usá-lo para configurar o ambiente que você deseja criar.

> **Note**
>
> Em Criar um ambiente do Elastic Beanstalk, mostramos como iniciar o assistente Criar ambiente e criar rapidamente um ambiente com valores padrão e configurações recomendadas. Este tópico explicará todas as opções.

Página do assistente

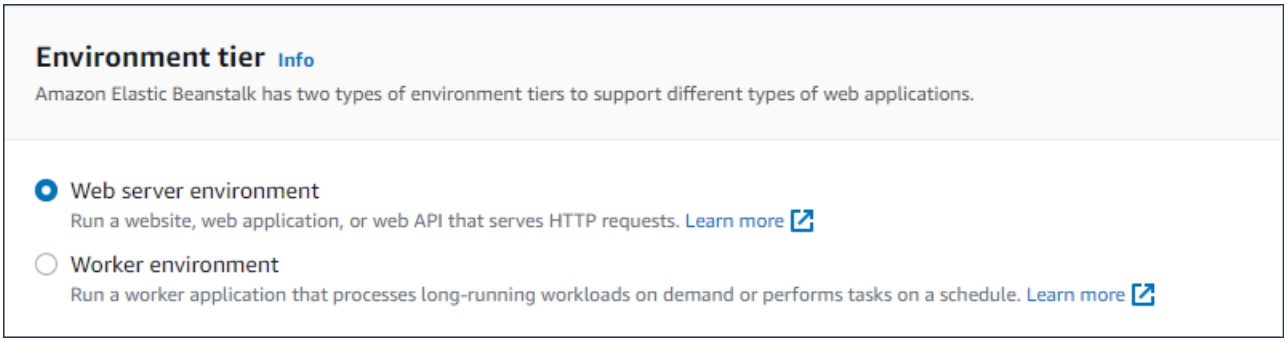
O assistente Create environment (Cria ambiente) fornece um conjunto de etapas para a criação de um novo ambiente.



O assistente de criação de novo ambiente

Nível do ambiente

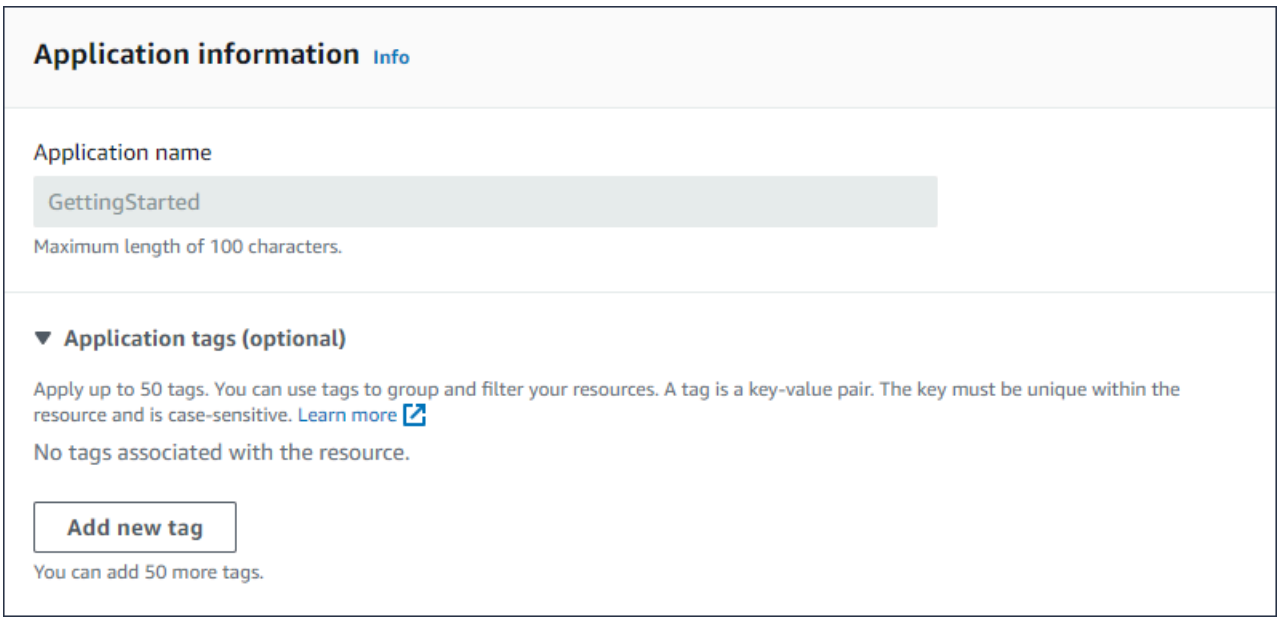
Em environment tier (nível de ambiente), escolha Web server environment (Ambiente do servidor Web) ou Worker environment (Ambiente do operador). Você não pode alterar o nível de um ambiente após a criação.



> **Note**
>
> O .NET na Plataforma de servidor Windows não tem suporte para a camada de ambiente do operador.

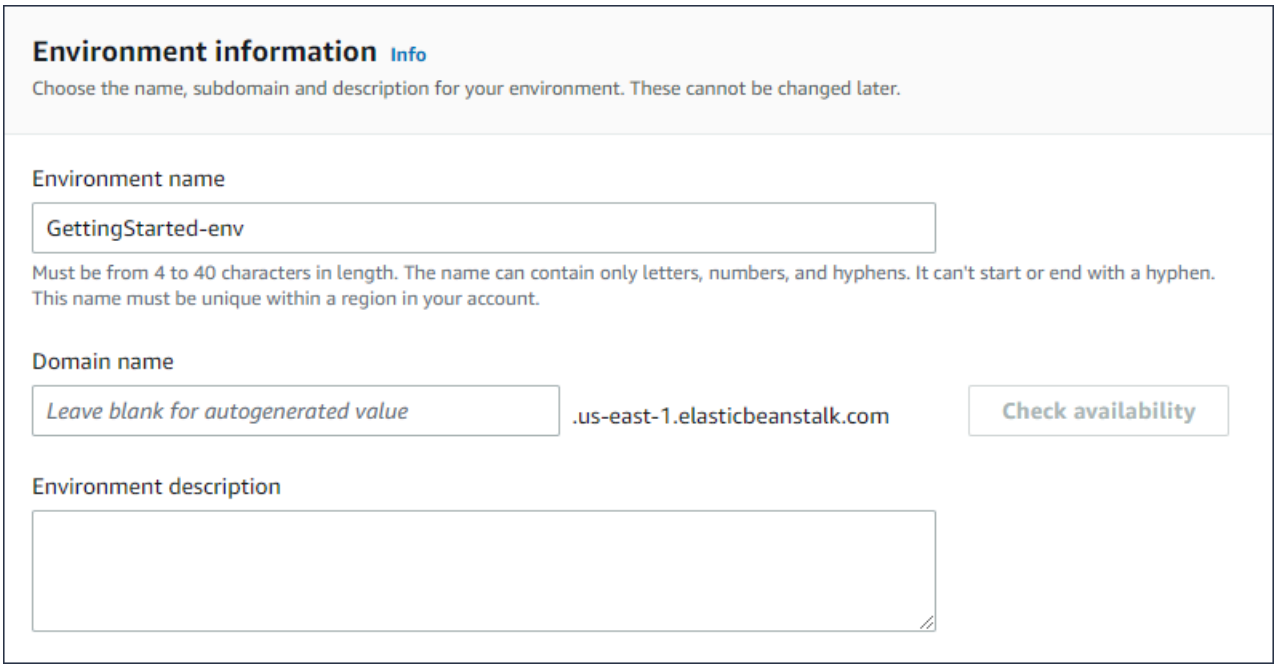
Informações sobre a aplicação

Se você iniciou o assistente selecionando Create new environment (Criar novo ambiente) na página de Application overview (Visão geral da aplicação, o Application name (Nome da aplicação) foi preenchido previamente. Do contrário, insira um nome de aplicação. Opcionalmente, adicione tags da aplicação.



Informações sobre o ambiente

Defina o nome e o subdomínio do ambiente, e crie uma descrição para o ambiente. Lembre-se de que essas configurações de ambiente não podem ser alteradas após a criação do ambiente.



- Name (Nome): insira um nome para o ambiente. O formulário fornece um nome gerado.
- Domain (Domínio): (ambientes de servidor Web) insira um nome de domínio exclusivo para o ambiente. O nome padrão é o nome do ambiente. Você pode inserir outro nome de domínio. O Elastic Beanstalk usa esse nome para criar um CNAME exclusivo para o ambiente. Para

verificar se o nome do domínio desejado está disponível, escolha Check Availability (Verificar disponibilidade).

- Description (Descrição): insira uma descrição para esse ambiente.

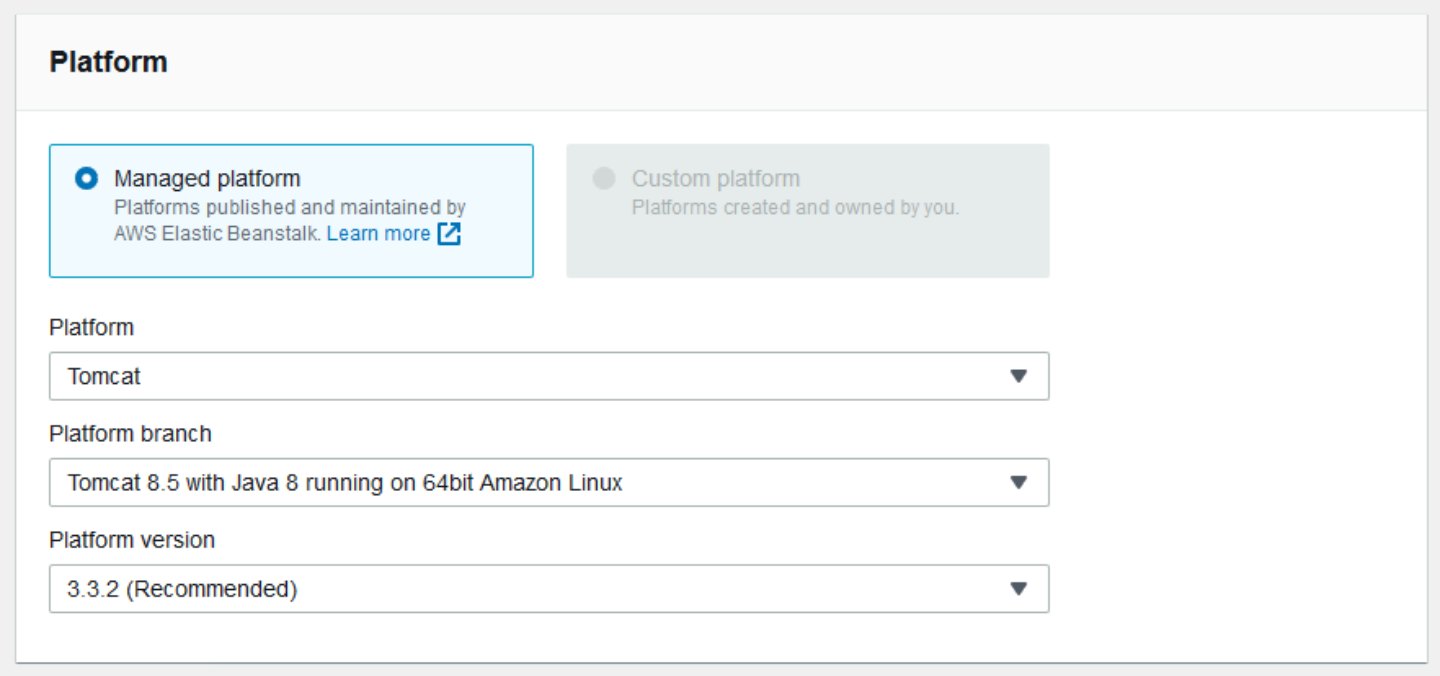
Selecionar uma plataforma para o novo ambiente

Você pode criar um novo ambiente a partir de dois tipos de plataformas:

- Plataforma gerenciada
- Plataforma personalizada

Plataforma gerenciada

Na maioria dos casos, é usada uma plataforma gerenciada do Elastic Beanstalk para seu novo ambiente. Quando o novo assistente de ambiente é iniciado, ele seleciona a opção Managed platform (Plataforma gerenciada) por padrão.



Selecione uma plataforma, uma ramificação de plataforma dentro dessa plataforma e uma versão específica da plataforma na ramificação. Quando você seleciona uma ramificação de plataforma, a versão recomendada dentro da ramificação é selecionada por padrão. Além disso, é possível selecionar qualquer versão de plataforma que você usou antes.

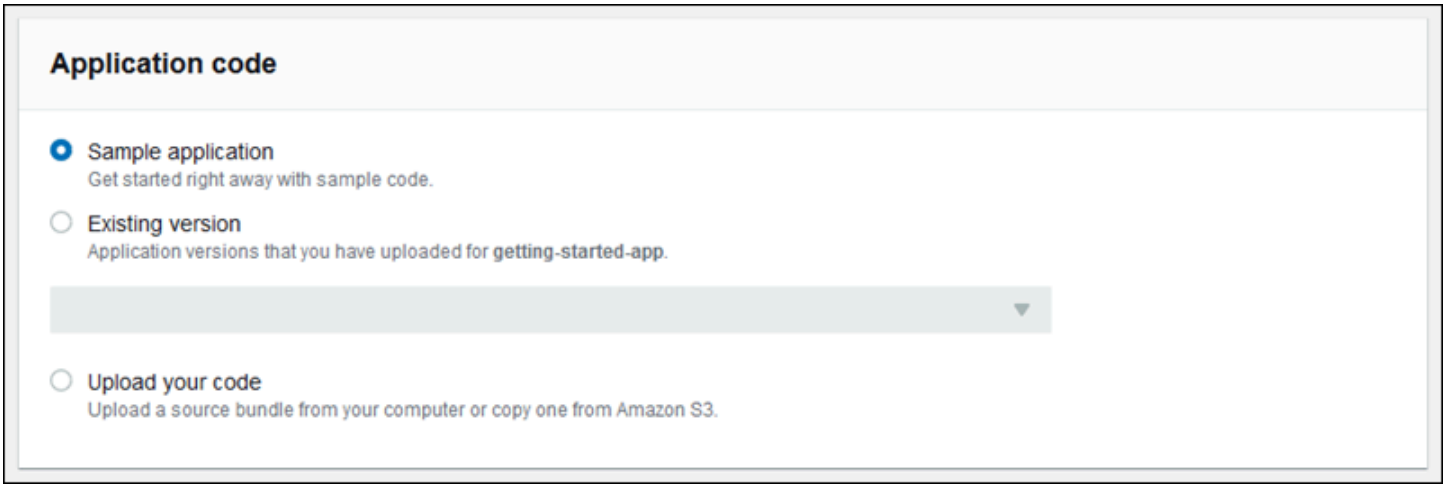
> **Note**
>
> Para um ambiente de produção, recomendamos escolher uma versão de plataforma em uma ramificação compatível da plataforma. Para obter detalhes sobre estados de ramificação da plataforma, consulte a definição de Ramificação da plataforma no the section called "Glossário de plataformas".

Plataforma personalizada

Se uma off-the-shelf plataforma não atender às suas necessidades, você poderá criar um novo ambiente a partir de uma plataforma personalizada. Para especificar uma plataforma personalizada, escolha a opção Custom platform (Plataforma personalizada) e selecione uma das plataformas personalizadas disponíveis. Se não houver plataformas personalizadas disponíveis, essa opção ficará esmaecida.

Fornecer o código do aplicativo

Agora que você selecionou a plataforma a ser usada, a próxima etapa é fornecer o código do aplicativo.



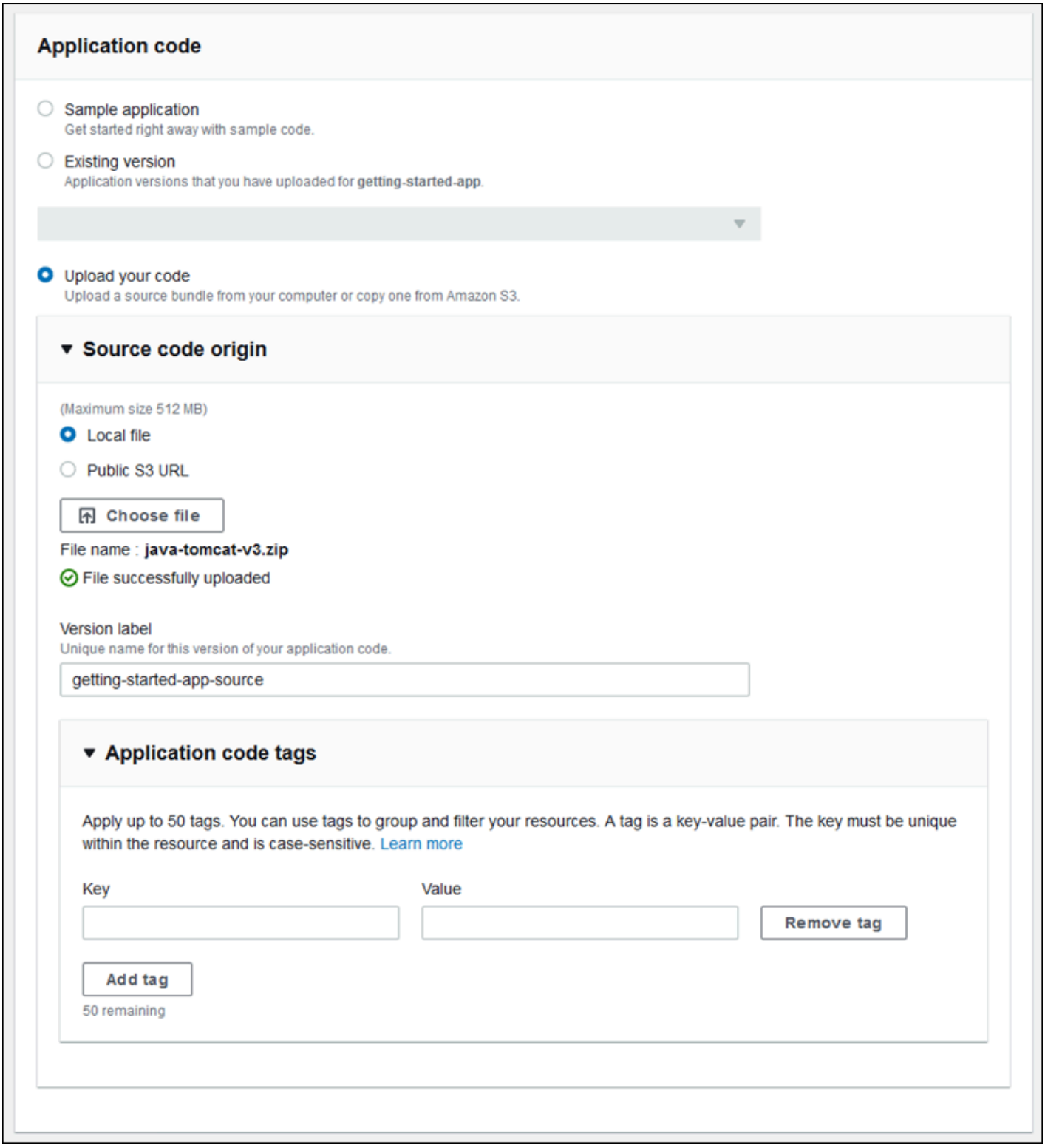
Você tem várias opções:

- É possível usar a aplicação demonstrativa que o Elastic Beanstalk fornece para cada plataforma.
- Você pode usar o código que já implantou no Elastic Beanstalk. Escolha Existing version (Versão existente) e seu aplicativo na seção Application code (Código do aplicativo).

- Você pode fazer upload do novo código. Escolha Upload your code (Fazer upload do código) e Upload (Fazer upload). Você pode fazer upload do novo código da aplicação de um arquivo local ou especificar o URL do bucket do Amazon S3 que contém o código da aplicação.

  > **Note**
  >
  > Dependendo da versão da plataforma selecionada, você pode fazer upload da aplicação em um pacote de origem ZIP, um arquivo WAR ou uma configuração do Docker de texto simples. O tamanho máximo do arquivo é 500 MB.

  Quando você opta por fazer upload de um novo código, também é possível fornecer tags para associar ao novo código. Para obter mais informações sobre como marcar uma versão do aplicativo, consulte the section called “Atribuir tags em versões do aplicativo”.


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

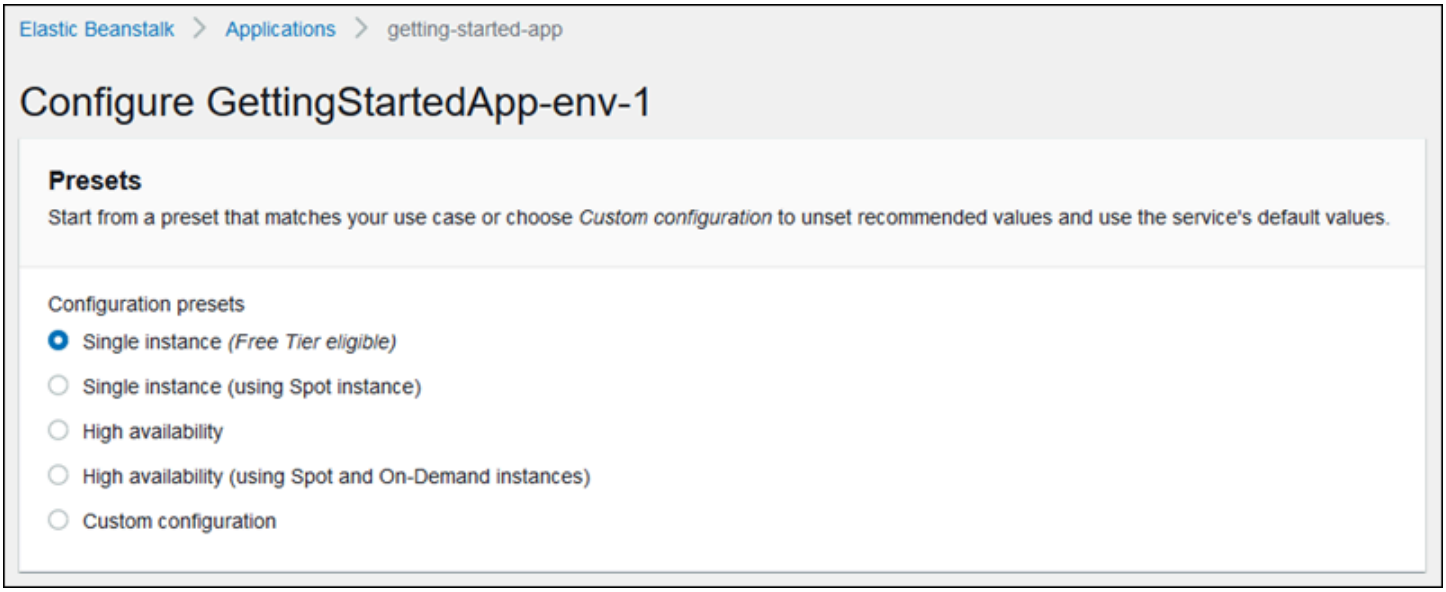
Para uma rápida criação do ambiente usando opções de configuração padrão, agora é possível escolher Create environment (Criar ambiente). Escolha Configure more options (Configurar mais opções) para fazer alterações adicionais na configuração, conforme descrito nas seções a seguir.

Página de configuração do assistente

Quando você escolhe Configure more options (Configurar mais opções), o assistente exibe a página Configure (Configurar). Nessa página, é possível selecionar uma predefinição de configuração, alterar a versão da plataforma que deseja que seu ambiente use ou fazer escolhas de configuração específicas para o novo ambiente.

Escolher uma configuração predefinida

Na seção Presets (Predefinições) da página, o Elastic Beanstalk fornece várias predefinições de configuração para diferentes casos de uso. Cada predefinição inclui valores recomendados para várias opções de configuração.



As predefinições de High availability (Alta disponibilidade) incluem um load balancer e são recomendadas para ambientes de produção. Escolha essa opção se você quiser um ambiente que possa executar várias instâncias com alta disponibilidade e reduzir a resposta à carga. As predefinições de Single instance (Instância única) são recomendadas, principalmente, para desenvolvimento. Duas das predefinições permitem solicitações de instâncias spot. Para obter detalhes sobre a configuração de capacidade do Elastic Beanstalk, consulte Auto Scaling group (Grupo do Auto Scaling).

A última predefinição, Custom configuration (Configuração personalizada), remove todos os valores recomendados, exceto as configurações de função, e usa os padrões de API. Escolha essa opção se você estiver implantando um pacote de origem com arquivos de configuração que definem opções de configuração. A Custom configuration (Configuração personalizada) também é selecionada automaticamente quando você modifica as predefinições de configuração Low cost (Baixo custo) ou High availability (Alta disponibilidade).

Personalizar sua configuração

Além de (ou em vez de) escolher uma predefinição de configuração, você pode ajustar as opções de configuração em seu ambiente. O assistente para Configure (Configurar) exibe várias categorias de configuração. Cada categoria de configuração exibe um resumo dos valores para um grupo de definições de configurações. Escolha Edit (Editar) para editar esse grupo de configurações.

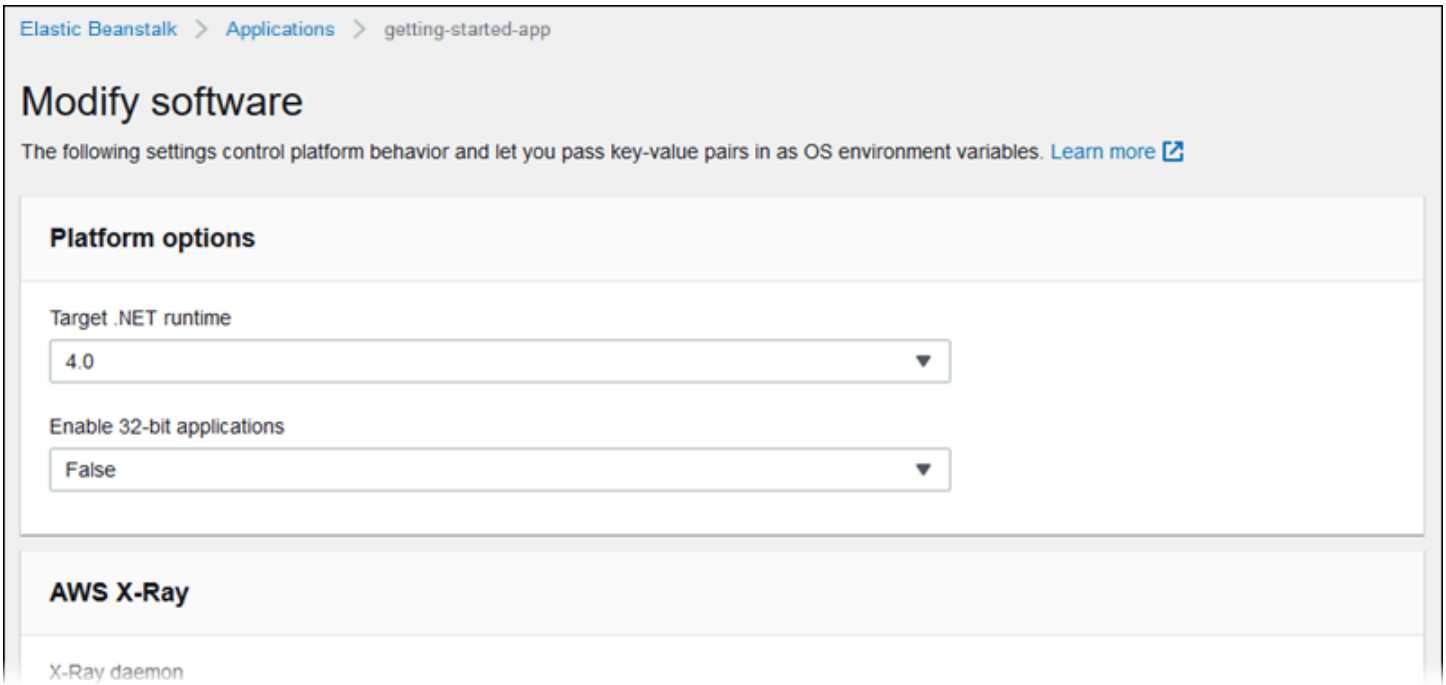
Categorias de configuração

- Configurações de software
- Instâncias
- Capacidade
- Load balancer
- Atualizações e implantações contínuas
- Segurança
- Monitoramento
- Atualizações gerenciadas
- Notificações
- Rede
- Banco de dados
- Tags
- Ambientes de operador

Configurações de software

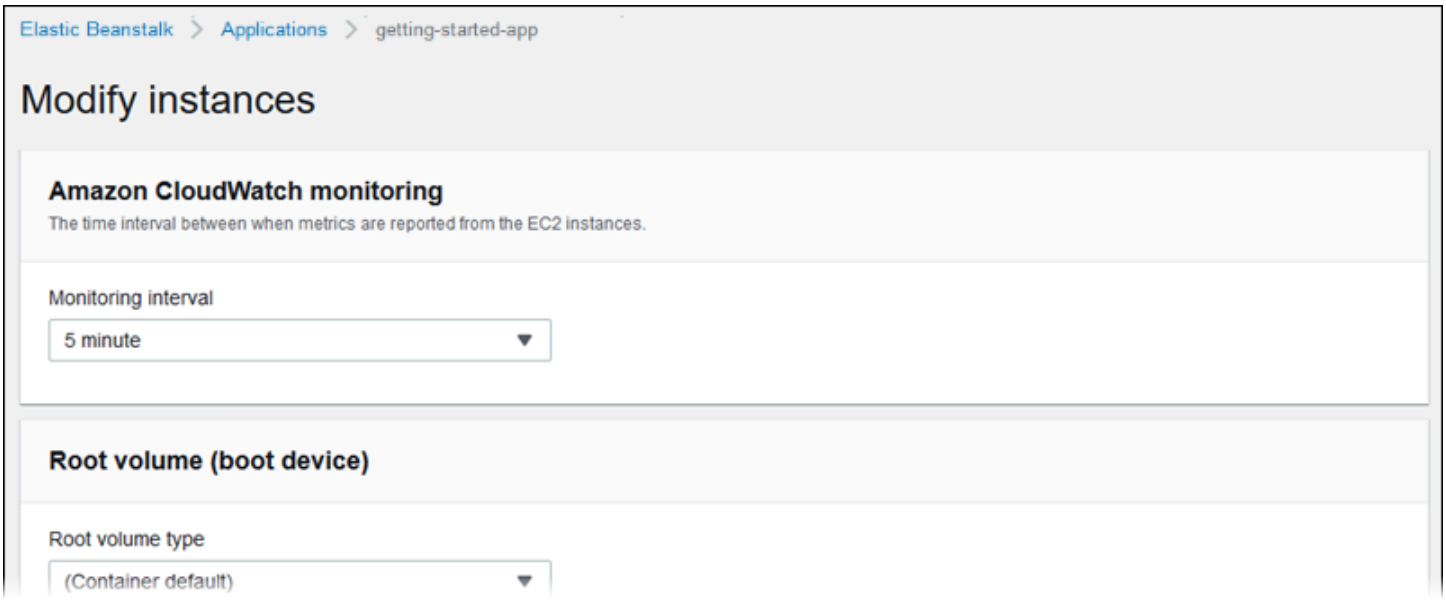
Use a página Modificar configuração de software para configurar o software nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que executam seu aplicativo. Você pode definir propriedades do ambiente, AWS X-Ray depuração, armazenamento e streaming de registros de

instâncias e configurações específicas da plataforma. Para obter detalhes, consulte the section called “Variáveis de ambiente e configurações de software”.



Instâncias

Use a página de configuração Modificar instâncias para configurar as EC2 instâncias da Amazon que executam seu aplicativo. Para obter detalhes, consulte the section called “ EC2 Instâncias da Amazon”.



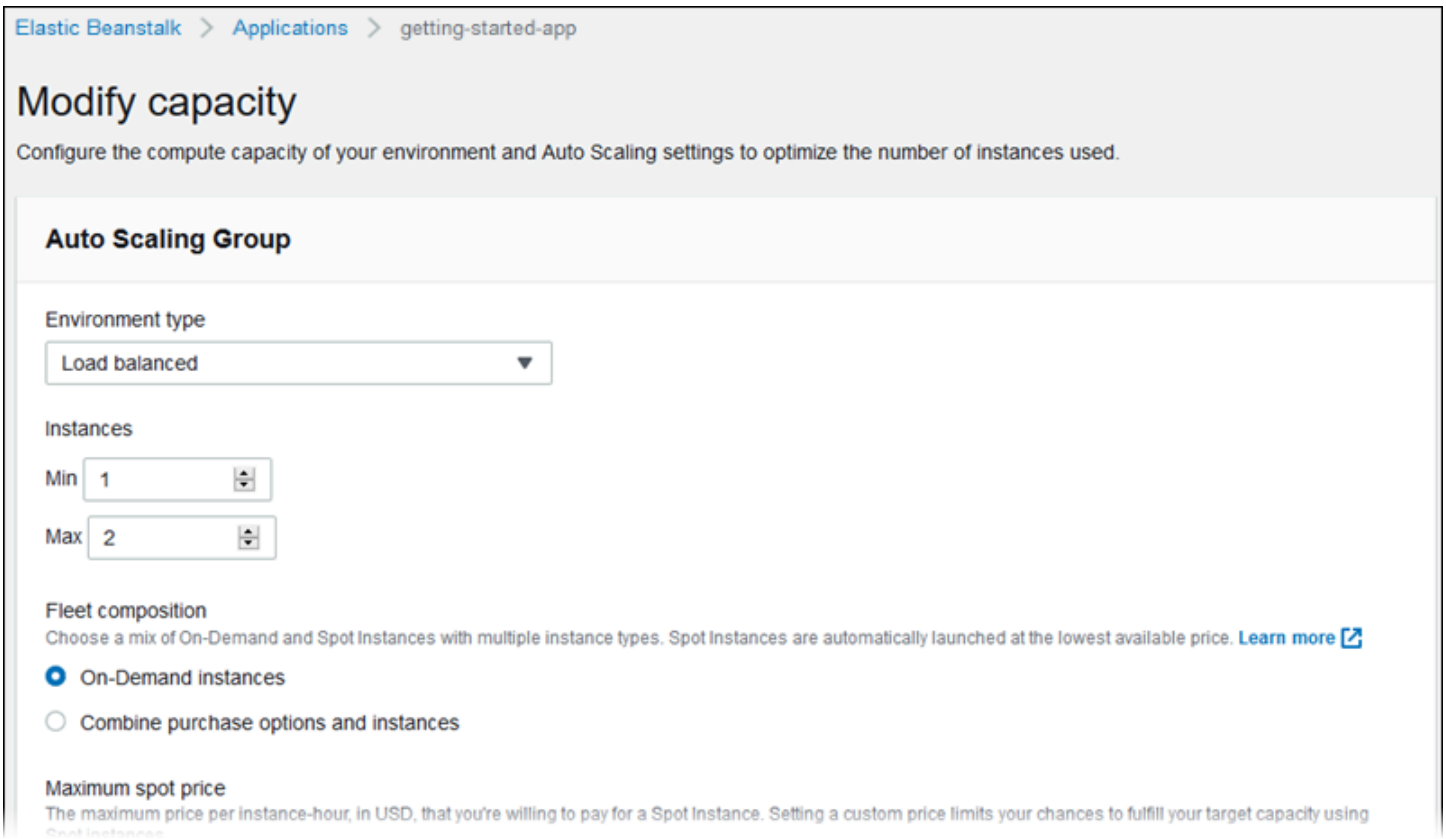
Capacidade

Use a página de configuração Modify capacity (Modificar capacidade) para configurar a capacidade computacional do ambiente e das definições Auto Scaling group (Grupo de Auto Scaling) para otimizar o número e o tipo de instâncias que você está usando. Também é possível alterar a capacidade do ambiente com base em triggers ou em uma programação.

Um ambiente com balanceamento de carga pode executar várias instâncias para alta disponibilidade e evitar o tempo de inatividade durante as atualizações de configuração e implantações. Em um ambiente com balanceamento de carga, o nome de domínio é mapeado para o load balancer. Em um ambiente de uma instância única, ele é mapeado para um endereço IP elástico na instância.

> **Warning**
>
> Um ambiente de única instância não está pronto para produção. Se a instância tornar-se instável durante a implantação, ou o Elastic Beanstalk encerrar e reiniciar a instância durante uma atualização de configuração, sua aplicação poderá ficar indisponível durante um determinado período. Use ambientes de única instância para desenvolvimento, teste ou preparação. Use ambientes com balanceamento de carga para produção.

Para obter mais informações sobre configurações de capacidade do ambiente, consulte the section called “Auto Scaling group (Grupo do Auto Scaling)” e the section called “ EC2 Instâncias da Amazon”.

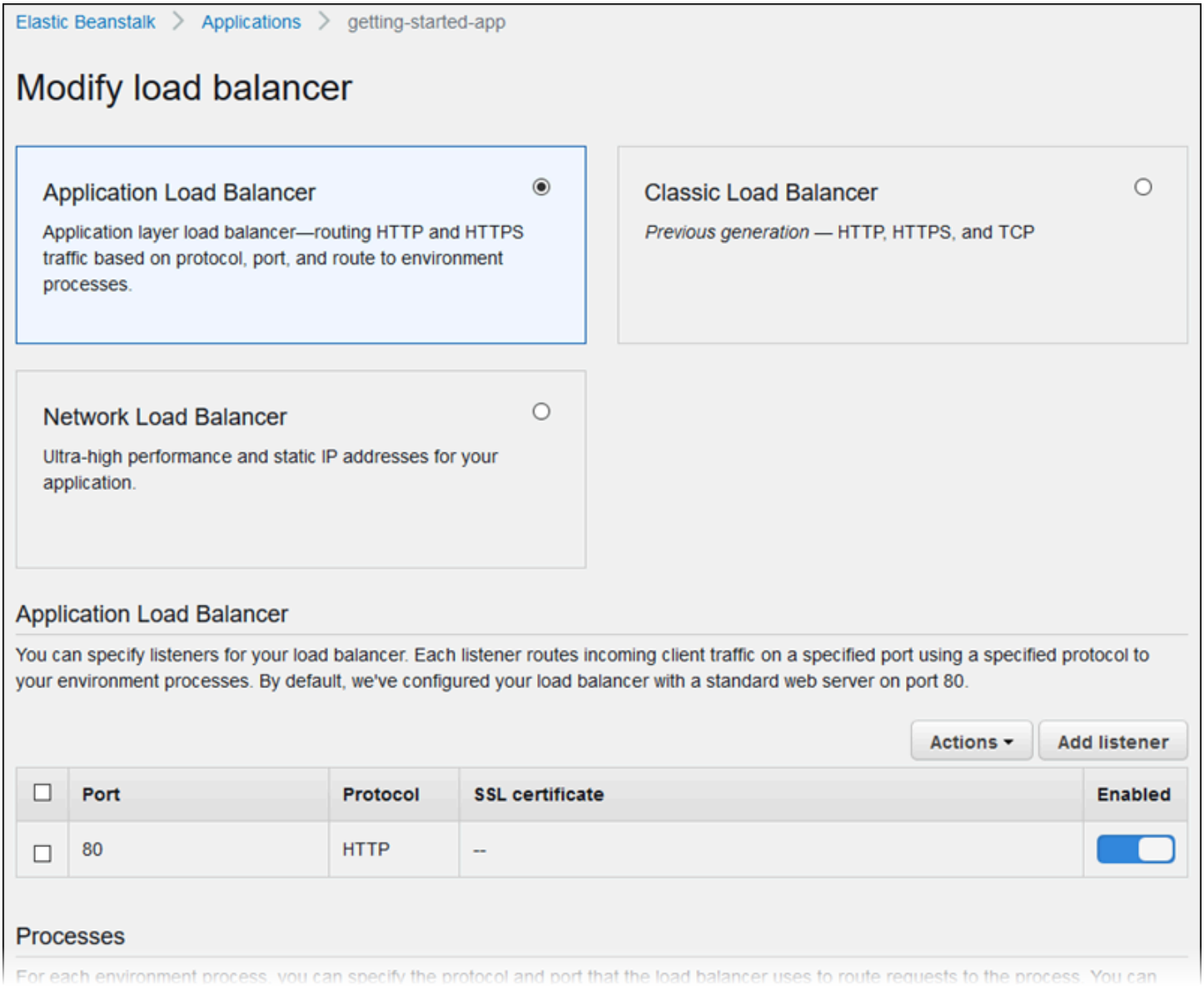


Load balancer

Use a página de configuração Modify Load balancer (Modificar load balancer) para selecionar um tipo de load balancer e definir suas configurações. Em ambientes com balanceamento de carga, o load balancer do ambiente é o ponto de entrada de todo o tráfego de seu aplicativo. O Elastic Beanstalk é compatível com vários tipos de balanceador de carga. Por padrão, o console do Elastic Beanstalk cria um Application Load Balancer e o configura para servir o tráfego HTTP na porta 80.

> **Note**
>
> É possível selecionar somente o tipo de load balancer do ambiente durante a criação do ambiente.

Para obter mais informações sobre os tipos e configurações de load balancer, consulte the section called “Load balancer” e the section called “HTTPS”.

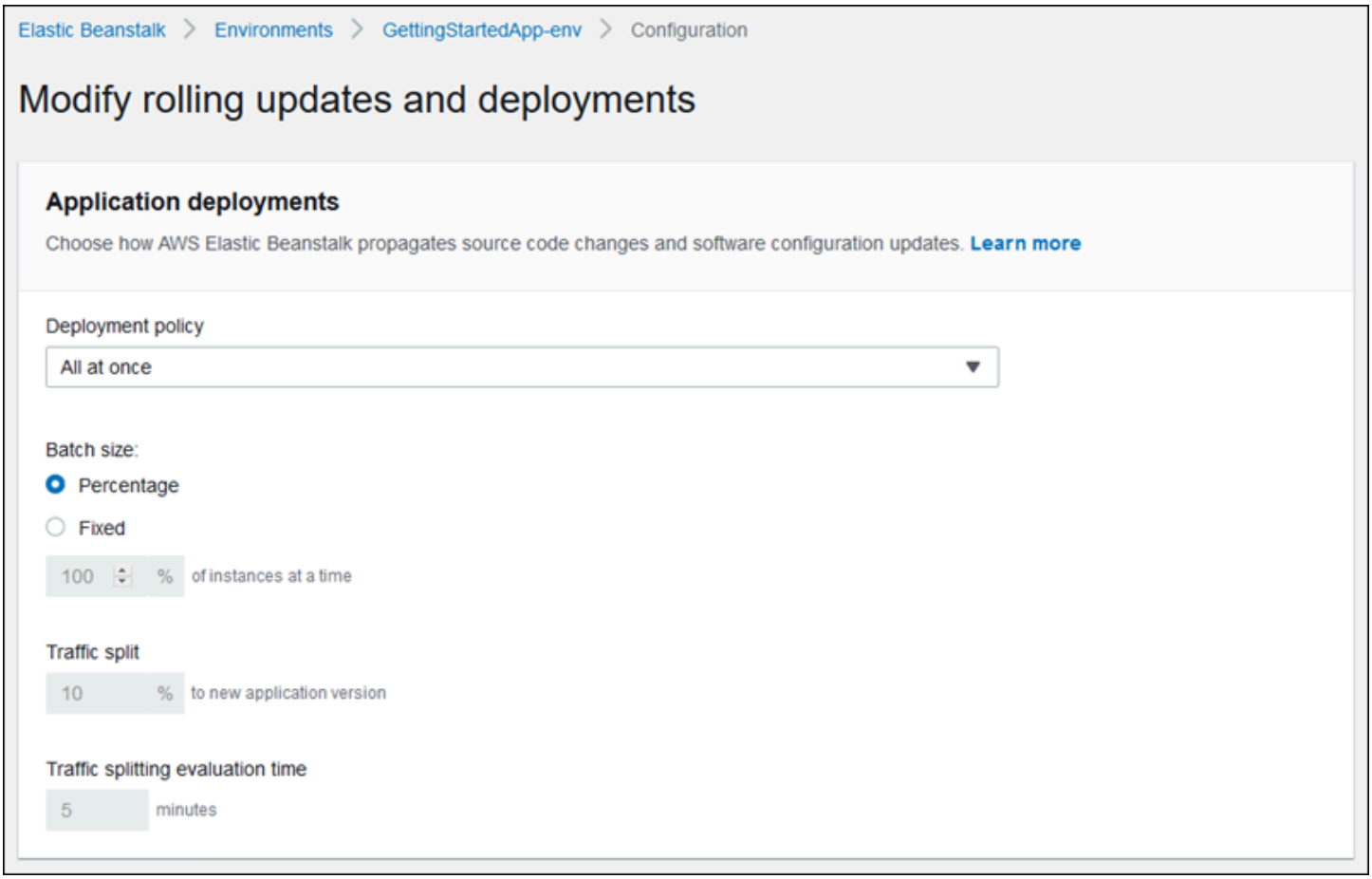


> **Note**
>
> A opção Classic Load Balancer (CLB) está desabilitada no assistente do console Criar ambiente. Se você tiver um ambiente existente configurado com um Classic Load Balancer, poderá criar um novo clonando o ambiente existente usando o console do Elastic Beanstalk ou a CLI do EB. Você também tem a opção de usar o EB CLI ou AWS CLIcriar um novo ambiente configurado com um Classic Load Balancer. Essas ferramentas de linha de comando criarão um novo ambiente com um CLB mesmo que ainda não exista um em sua conta.

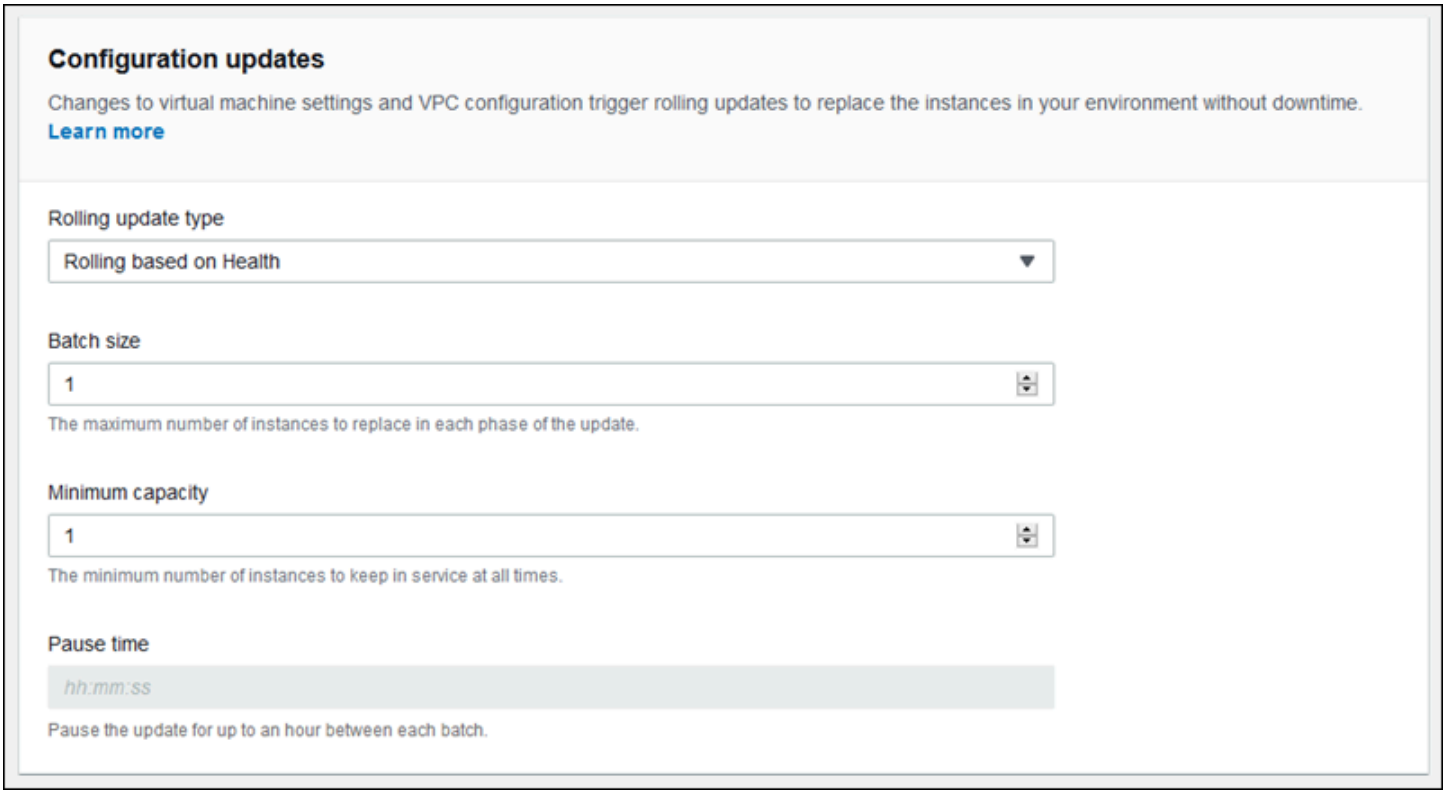
Atualizações e implantações contínuas

Use a página de configuração Modify rolling updates and deployments (Modificar atualizações contínuas e implantações) para configurar como o Elastic Beanstalk processa implantações de aplicações e atualizações de configuração para seu ambiente.

As implantações de aplicativos ocorrem quando você carrega um pacote de origem de aplicativo atualizado e o implementa em seu ambiente. Para obter mais informações sobre como configurar implantações, consulte the section called “Opções de implantação”.



As alterações de configuração que modificam a configuração de execução ou configurações de VPC exigem o encerramento de todas as instâncias no ambiente e a substituição. Para obter mais informações sobre como definir o tipo de atualização e outras opções, consulte the section called “Alterações de configuração”.



Segurança

Use a página Configure service access (Configurar acesso ao serviço) para configurar as definições de segurança dos serviços e das instâncias.

Para obter uma descrição dos conceitos de segurança do Elastic Beanstalk, consulte Permissões.

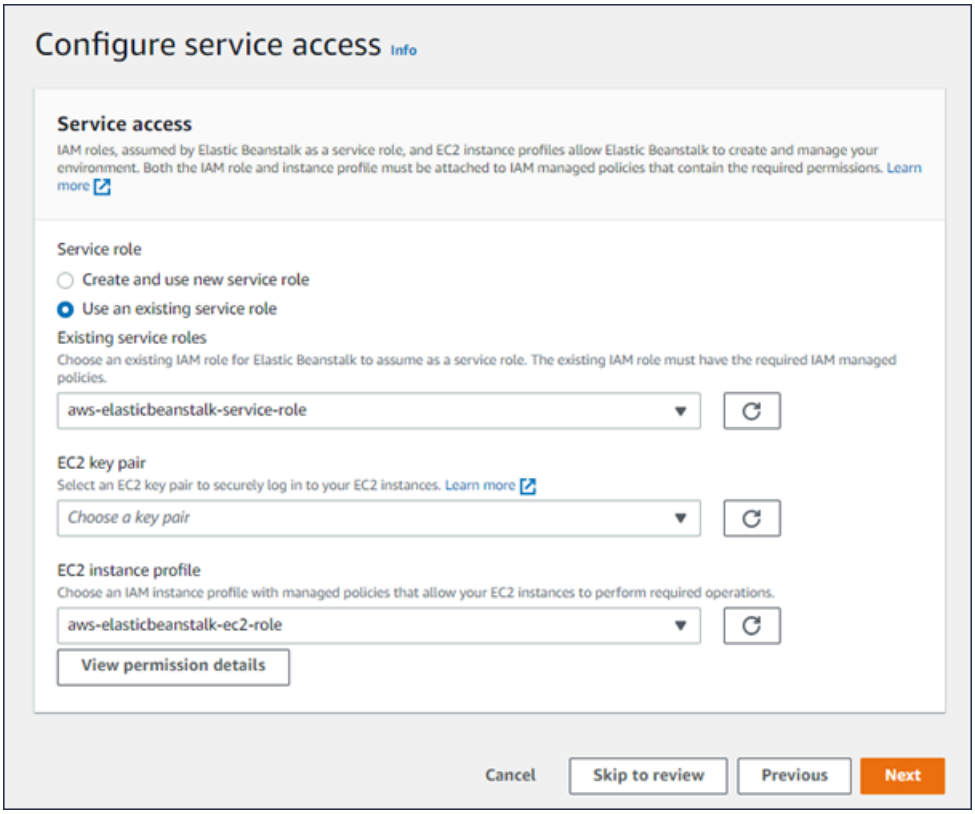
Na primeira vez que você cria um ambiente no console do Elastic Beanstalk, você deve EC2 criar um perfil de instância com um conjunto padrão de permissões. Se a lista suspensa do perfil da EC2 instância não mostrar nenhum valor para escolher, expanda o procedimento a seguir. Ele fornece etapas para criar uma função que você pode selecionar posteriormente para o perfil da EC2 instância.

Crie uma função do IAM para o perfil da EC2 instância

Para criar o perfil da EC2 instância

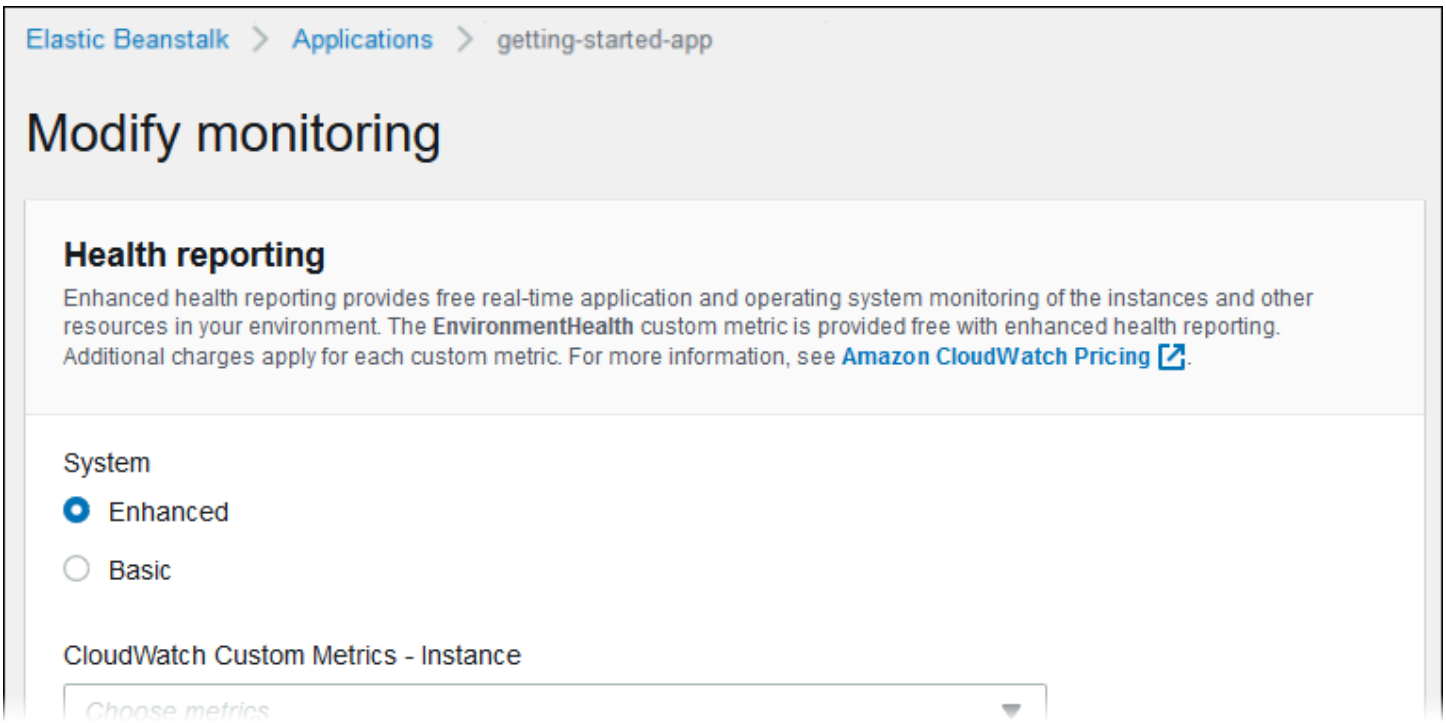
1. Selecione Criar perfil.
2. Em Tipo de entidade confiável, escolha Serviços da AWS .
3. Para o caso de uso, escolha Elastic Beanstalk — Compute.
4. Escolha Próximo.

5. Verifique se as políticas de permissões incluem o seguinte e escolha Avançar:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
6. Selecione Criar perfil.
7. Volte para a guia Configurar acesso ao serviço, atualize a lista e selecione o perfil de EC2 instância recém-criado.



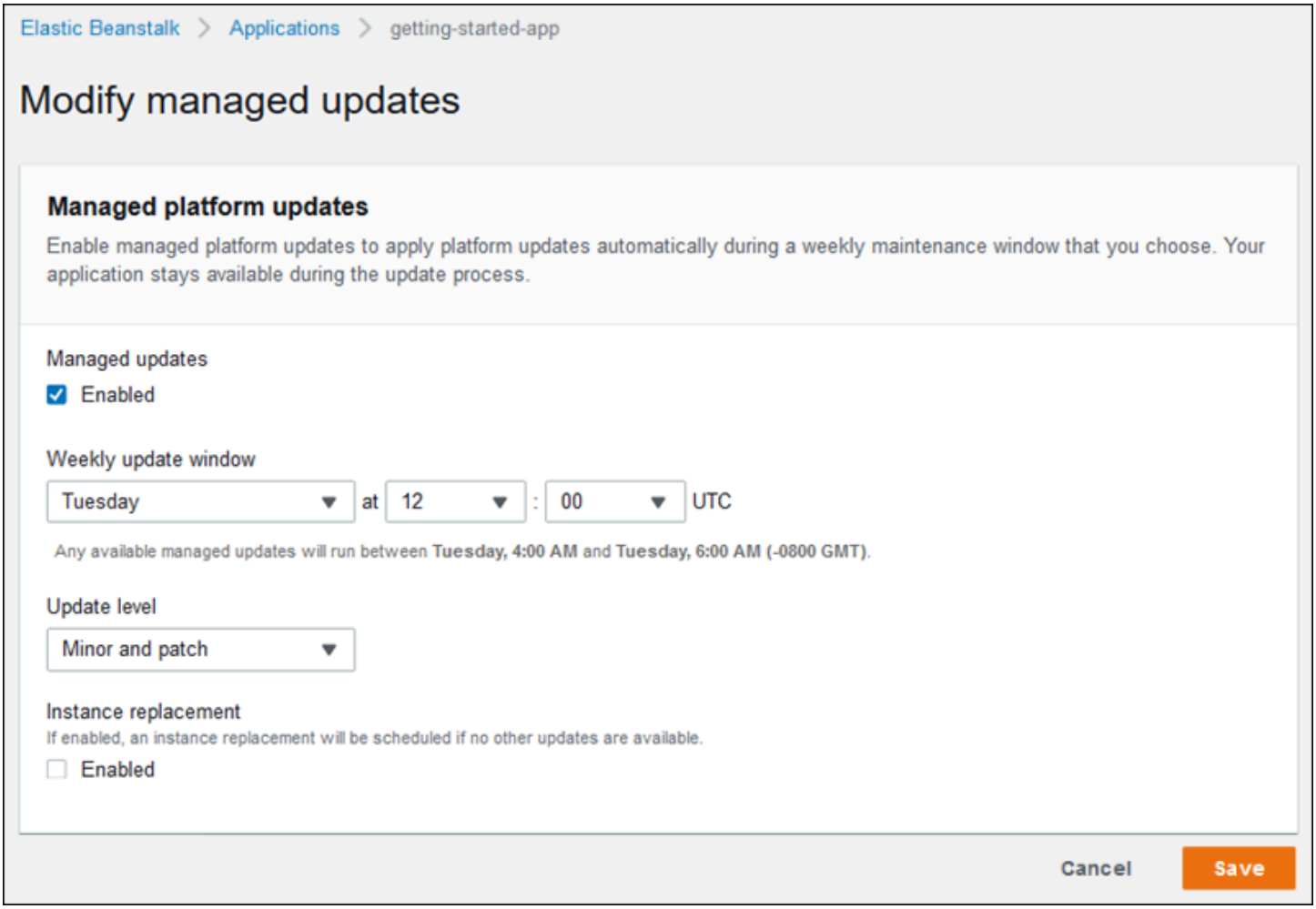
Monitoramento

Use a página de configuração Modify monitoring (Modificar monitoramento) para configurar relatórios de integridade, regras de monitoramento e streaming de eventos de integridade. Para obter detalhes, consulte the section called “Habilitar integridade aprimorada”, the section called “Regras de integridade aperfeiçoada”, e the section called “Streaming de integridade do ambiente”.



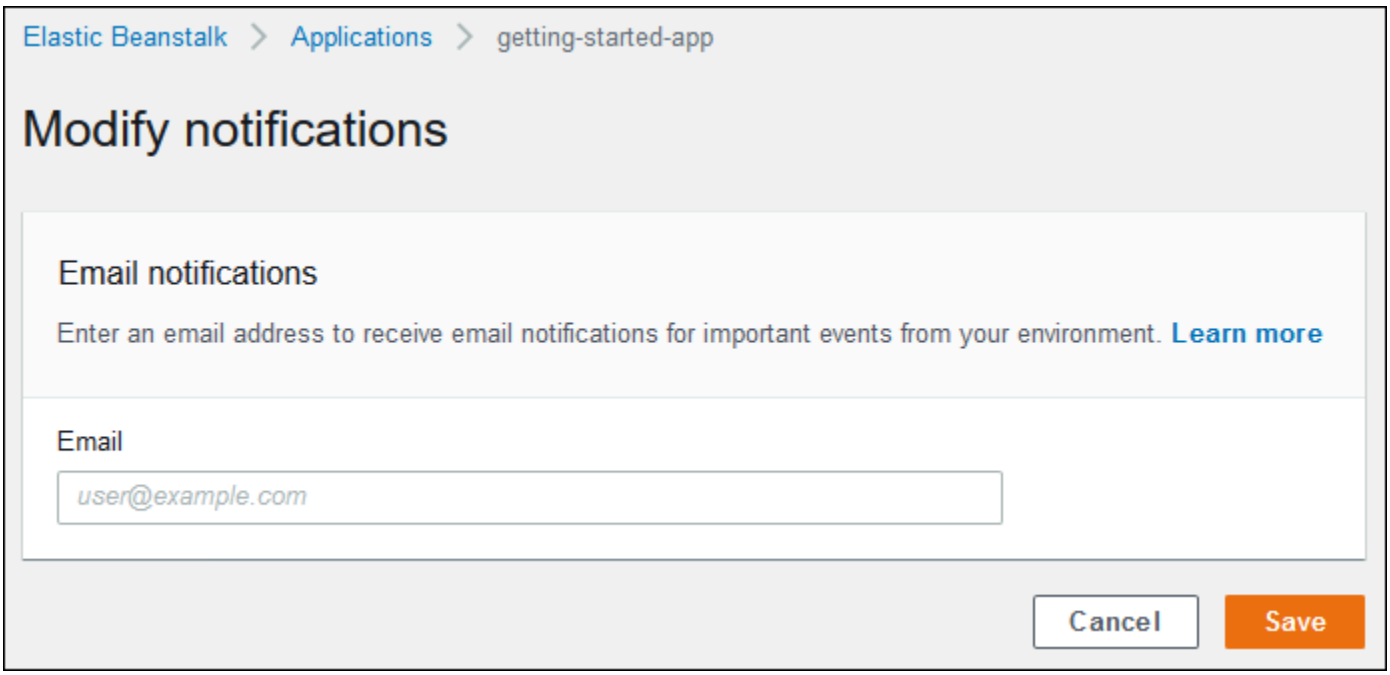
Atualizações gerenciadas

Use a página de configuração Modify managed updates (Modificar atualizações gerenciadas) para configurar atualizações gerenciadas da plataforma. É possível decidir se deseja habilitá-los, definir a programação e configurar outras propriedades. Para obter mais detalhes, consulte the section called “Atualizações gerenciadas”.



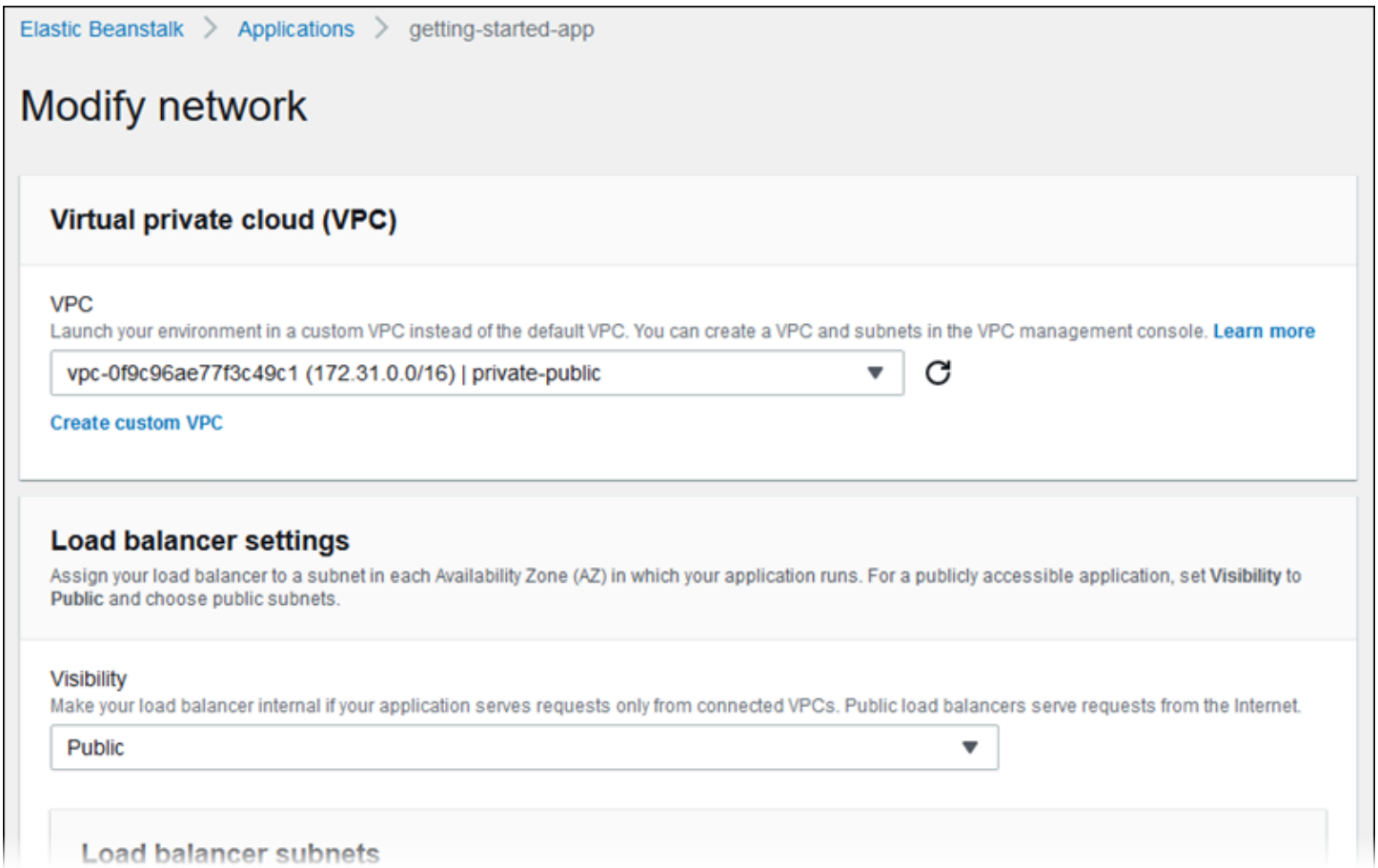
Notificações

Use a página de configuração Modify notifications (Modificar notificações) para especificar um endereço de e-mail para receber notificações por e-mail de eventos importantes do seu ambiente.



Rede

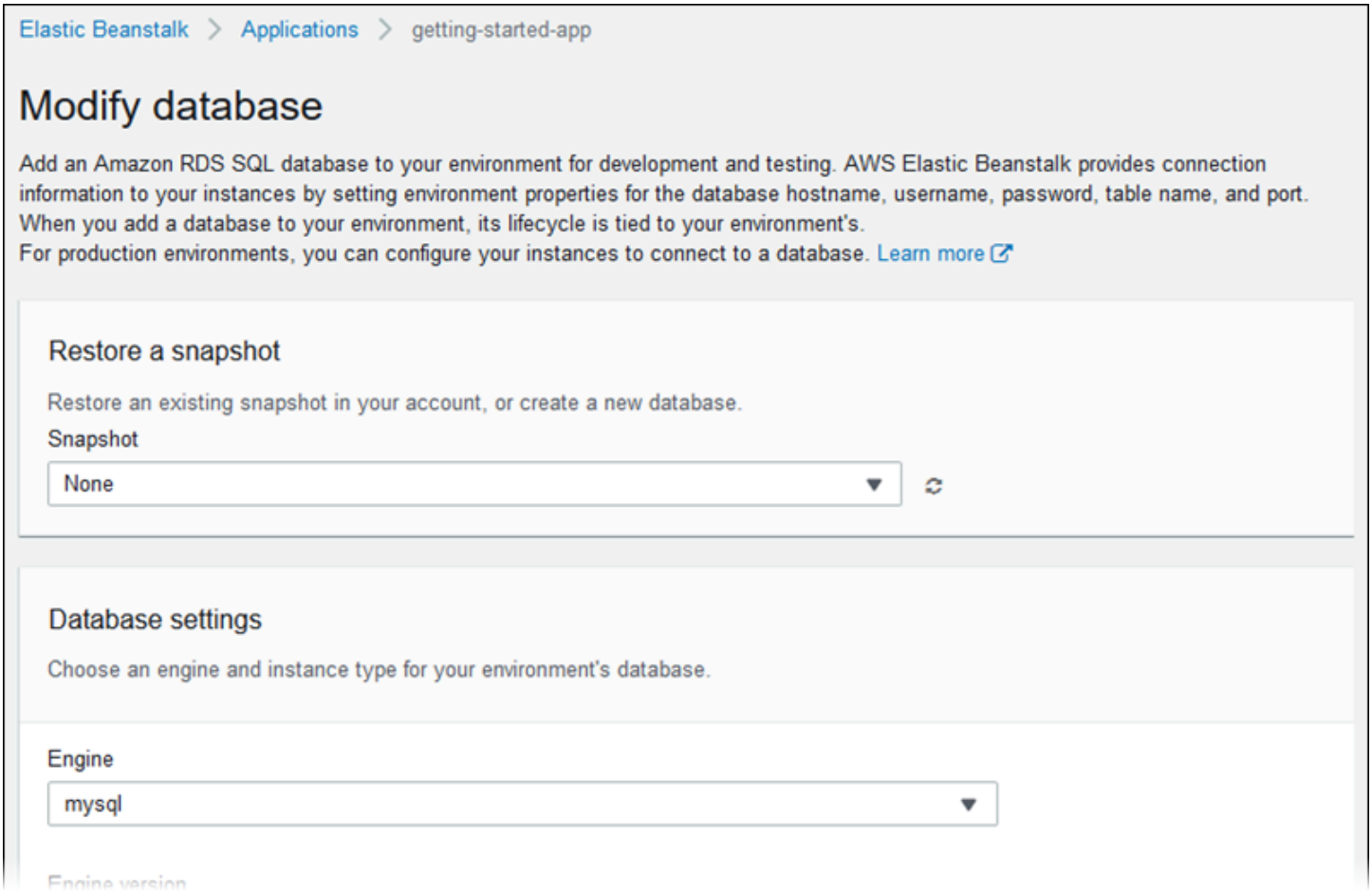
Se você criou uma VPC personalizada, use a página de configuração Modify network (Modificar rede) a fim de configurar seu ambiente para usá-lo. Se você não escolher uma VPC, o Elastic Beanstalk usará a VPC e as sub-redes padrão.



Banco de dados

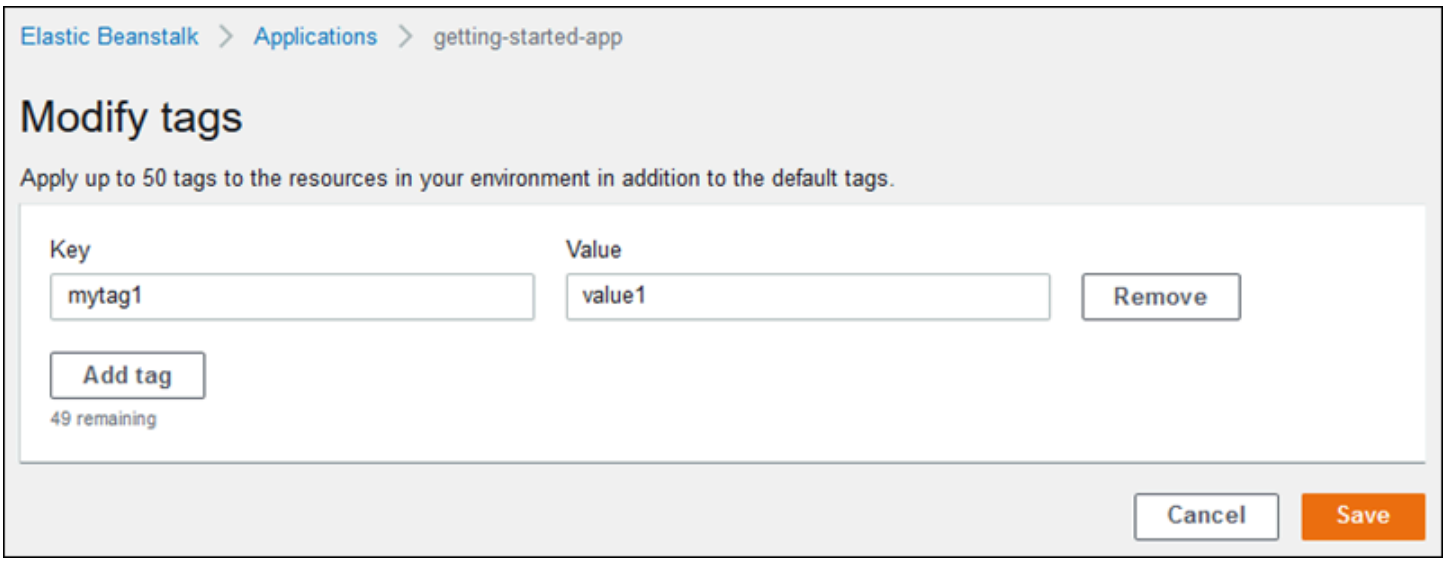
Use a página de configuração Modify database (Modificar banco de dados) para adicionar um banco de dados do Amazon Relational Database Service (Amazon RDS) ao seu ambiente para desenvolvimento e teste. O Elastic Beanstalk fornece informações de conexão a suas instâncias definindo as propriedades do ambiente para o nome do host do banco de dados, o nome do usuário, a senha, o nome da tabela e a porta.

Para obter detalhes, consulte the section called “Banco de dados”.



Tags

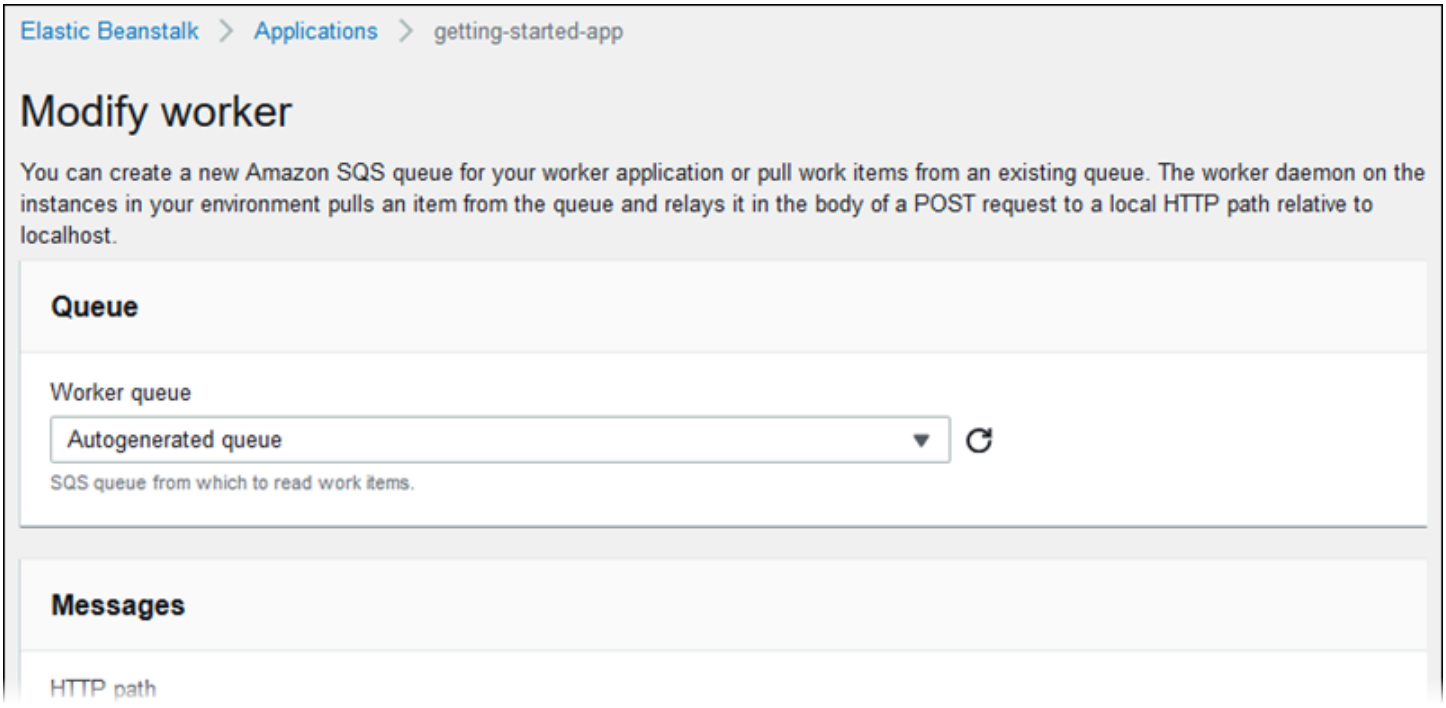
Use a página de configuração Modify tags (Modificar tags) para adicionar tags aos recursos no ambiente. Para obter mais informações sobre marcação do ambiente, consulte Marcar recursos em seus ambientes do Elastic Beanstalk.



Ambientes de operador

Se você estiver criando um ambiente de nível do operador, use a página de configuração Modify worker (Modificar operador) para configurar o ambiente de operador. O daemon do operador nas instâncias no ambiente extrai itens de uma fila do Amazon Simple Queue Service (Amazon SQS) e os retransmite como mensagens de publicação para a aplicação de operador. É possível escolher a fila do Amazon SQS na qual o daemon do operador lê (gerada automaticamente ou existente). Também é possível configurar as mensagens que o daemon do operador envia para seu aplicativo.

Para obter mais informações, consulte the section called “Ambientes de operador”.



Clonar um ambiente do Elastic Beanstalk

É possível criar um ambiente do Elastic Beanstalk existente como a base para um novo ambiente clonando o ambiente existente. Por exemplo, você pode criar um clone para usar uma versão mais recente da ramificação da plataforma utilizada pela plataforma do ambiente original. O Elastic Beanstalk configura o clone com as configurações de ambiente usadas pelo ambiente original. Ao clonar um ambiente existente em vez de criar um novo, você não precisa configurar manualmente as opções, variáveis de ambiente e outras configurações feitas com o serviço Elastic Beanstalk. O Elastic Beanstalk também cria uma cópia AWS de qualquer recurso associado ao ambiente original.

É importante estar ciente das situações a seguir:

- Durante o processo de clonagem, o Elastic Beanstalk não copia dados do Amazon RDS para o clone.
- O Elastic Beanstalk não inclui nenhuma alteração não gerenciada feita nos recursos do clone. As alterações feitas nos recursos da AWS usando ferramentas diferentes do console do Elastic Beanstalk, ferramentas de linha de comando ou API são consideradas não gerenciadas.
- Os grupos de segurança para entrada são considerados alterações não gerenciadas. Ambientes clonados do Elastic Beanstalk não transferem os grupos de segurança para entrada, deixando o ambiente aberto para todo o tráfego da Internet. Será necessário restabelecer os grupos de segurança de entrada para o ambiente clonado.

Só é possível clonar um ambiente para uma versão de plataforma diferente da mesma ramificação da plataforma. Não é garantido que uma ramificação de plataforma diferente seja compatível. Para usar uma ramificação de plataforma diferente, é necessário criar manualmente um ambiente, implantar o código do aplicativo e fazer as alterações necessárias no código e nas opções para garantir que o aplicativo funcione corretamente na nova ramificação da plataforma.

AWS console de gerenciamento

> **Important**
>
> Ambientes clonados do Elastic Beanstalk não transferem os grupos de segurança para entrada, deixando o ambiente aberto para todo o tráfego da Internet. Será necessário restabelecer os grupos de segurança de entrada para o ambiente clonado.

É possível ver recursos que não podem ser clonados verificando o status de desvio da configuração do ambiente. Para obter mais informações, consulte Detectar deriva em uma CloudFormation pilha inteira no Guia do AWS CloudFormation usuário.

Para clonar um ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Ações.
4. Escolha Ambiente de clonagem.
5. Na página Clone environment (Clonar ambiente), analise as informações na seção Original Environment (Ambiente original) para verificar se escolheu o ambiente com base no qual você deseja criar um clone.
6. Na seção New Environment (Novo ambiente), é possível alterar os valores de Environment name (Nome do ambiente), Environment URL (URL do ambiente), Description (Descrição), Platform version (Versão da plataforma) e Service role (Função de serviço) definidos automaticamente pelo Elastic Beanstalk com base no ambiente original.

   > **Note**
   >
   > Se a versão da plataforma usada no ambiente original não for a recomendada para uso na ramificação da plataforma, você será avisado de que uma versão da plataforma diferente é recomendada. Escolha a Platform version (Versão da plataforma), e será possível ver a versão recomendada da plataforma na lista, por exemplo, 3.3.2 (Recommended) (3.3.2 (Recomendado)).

7. Quando estiver pronto, escolha Clone (Clonar).

Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB)

> **Important**
>
> Ambientes clonados do Elastic Beanstalk não transferem os grupos de segurança para entrada, deixando o ambiente aberto para todo o tráfego da Internet. Será necessário restabelecer os grupos de segurança de entrada para o ambiente clonado.

É possível ver recursos que não podem ser clonados verificando o status de desvio da configuração do ambiente. Para obter mais informações, consulte Detectar deriva em uma CloudFormation pilha inteira no Guia do AWS CloudFormation usuário.

Use o comando eb clone para clonar um ambiente em execução, da seguinte maneira.

```
~/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
```

Você pode especificar o nome do ambiente de origem no comando de clone ou omiti-lo para clonar o ambiente padrão referente à pasta do projeto atual. A EB CLI solicita que você insira um nome e prefixo DNS para o novo ambiente.

Por padrão, o eb clone cria o novo ambiente com a versão mais recente disponível da plataforma do ambiente de origem. Para forçar a EB CLI a usar a mesma versão, mesmo que haja uma versão mais recente disponível, use a opção `--exact`.

```
~/workspace/my-app$ eb clone --exact
```

Para obter mais informações sobre esse comando, consulte eb clone.

Encerrar um ambiente do Elastic Beanstalk

Você pode encerrar um AWS Elastic Beanstalk ambiente em execução usando o console do Elastic Beanstalk. Ao fazer isso, você evita cobranças por recursos não utilizados AWS .

> **Note**
>
> Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Se você tiver dados de um ambiente que deseja manter, defina a política de exclusão de banco de dados como `Retain` antes de encerrar o ambiente. Isso mantém o banco de dados operacional fora do Elastic Beanstalk. Depois disso, qualquer ambiente do Elastic Beanstalk deve se conectar a ele

como um banco de dados externo. Se você quiser fazer backup dos dados sem manter o banco de dados operacional, defina a política de exclusão para fazer um snapshot do banco de dados antes de encerrar o ambiente. Para obter mais informações, consulte Ciclo de vida do banco de dados no capítulo Configuração de ambientes deste guia.

O Elastic Beanstalk pode falhar ao encerrar seu ambiente. Um motivo comum para isso é que outro grupo de segurança do ambiente tem uma dependência do grupo de segurança do ambiente que você está tentando terminar. Para obter instruções sobre como evitar esse problema, consulte a EC2 grupos de segurança página EC2Instâncias deste guia.

> **Important**
>
> Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.
>
> Para obter mais informações, consulte Proteção contra um registro de delegação pendente no Route 53 no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em Proteções de domínio aprimoradas para solicitações da CloudFront Amazon no AWS blog de segurança.

Console do Elastic Beanstalk

Para encerrar um ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

> **Note**
>
> Quando você termina o ambiente, o CNAME associado ao ambiente terminado torna-se disponível para qualquer pessoa utilizar.

São necessários alguns minutos para que o Elastic Beanstalk AWS encerre os recursos que estão sendo executados no ambiente.

AWS CLI

Encerrar um ambiente

- Execute o seguinte comando:

```
$ aws elasticbeanstalk terminate-environment --environment-name my-env
```

API

Encerrar um ambiente

- Chame `TerminateEnvironment` com o seguinte parâmetro:

  `EnvironmentName` = `SampleAppEnv`

```
https://elasticbeanstalk.us-west-2.amazon.com/?EnvironmentName=SampleAppEnv
&Operation=TerminateEnvironment
&AuthParams
```

Criação de ambientes do Elastic Beanstalk com a CLI da AWS

Para obter detalhes sobre os AWS CLI comandos do Elastic Beanstalk, consulte a Referência de comandos.AWS CLI

1. Verifique se o CNAME do ambiente está disponível.

```
$ aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

2. Verifique se a versão do aplicativo existe.

```
$ aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label v1
```

Se você não tiver uma versão do aplicativo para sua origem, crie-a. Por exemplo, o comando a seguir cria uma versão da aplicação a partir de um pacote de origem no 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. Crie um modelo de configuração para o aplicativo.

```
$ 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. Crie o 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
```

As configurações de opção são definidas no arquivo options.txt:

```
[
    {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": "aws-elasticbeanstalk-ec2-role"
    }
]
```

A configuração da opção acima define o perfil da instância do IAM. É possível especificar o ARN ou o nome do perfil.

5. Determine se o novo ambiente está verde e pronto.

```
$ aws elasticbeanstalk describe-environments --environment-names my-env
```

Se o novo ambiente não estiver verde e pronto, decida se deseja repetir a operação ou deixar o ambiente em seu estado atual para investigação. Certifique-se de encerrar o ambiente depois que você terminar e de limpar todos os recursos não utilizados.

> **Note**
>
> É possível ajustar o tempo limite se o ambiente não iniciar em um período razoável.

Criar ambientes do Elastic Beanstalk com a API

1. Chame `CheckDNSAvailability` com o seguinte parâmetro:

   - `CNAMEPrefix` = `SampleApp`

   Example

   ```
   https://elasticbeanstalk.us-east-2.amazonaws.com/?CNAMEPrefix=sampleapplication
   &Operation=CheckDNSAvailability
   &AuthParams
   ```

2. Chame `DescribeApplicationVersions` com os seguintes parâmetros:

   - `ApplicationName` = `SampleApp`
   - `VersionLabel` = `Version2`

   Example

   ```
   https://elasticbeanstalk.us-east-2.amazonaws.com/?ApplicationName=SampleApp
   &VersionLabel=Version2
   &Operation=DescribeApplicationVersions
   &AuthParams
   ```

3. Chame `CreateConfigurationTemplate` com os seguintes parâmetros:

   - `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. Chame CreateEnvironment com um dos seguintes conjuntos de parâmetros.

   a. Use o seguinte em um nível de ambiente de servidor 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. Use o seguinte em um nível de ambiente de operador:

- `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
```

Construção de um URL "Iniciar agora"

Você pode criar um localizador uniforme de recursos (URL) personalizado para que qualquer pessoa possa implantar e executar rapidamente uma aplicação Web predeterminada no AWS Elastic Beanstalk. Esse URL é chamado URL "Iniciar agora". Você pode precisar de um URL de início imediato, por exemplo, para demonstrar uma aplicação Web criada para ser executada no Elastic Beanstalk. Com a URL Launch Now (Iniciar agora), você pode usar parâmetros para adicionar antecipadamente as informações necessárias ao assistente de criação de aplicativo. Quando adiciona essa informação no assistente, qualquer pessoa pode usar o link do URL para iniciar um ambiente do Elastic Beanstalk com a fonte da aplicação Web em apenas algumas etapas. Isso significa que os usuários não precisam carregar ou especificar manualmente o local do pacote de origem da aplicação. Também não é necessário fornecer nenhuma informação adicional ao assistente.

Um URL de início imediato fornece ao Elastic Beanstalk as informações mínimas necessárias para criar uma aplicação: o nome da aplicação, a pilha de soluções, o tipo de instância e o tipo de ambiente. O Elastic Beanstalk usa valores padrão para os outros detalhes de configuração que não são explicitamente especificados no URL personalizado de início imediato.

Um URL "Iniciar agora" usa a sintaxe de URL padrão. Para obter mais informações, consulte RFC 3986 – URI (Uniform Resource Identifier): sintaxe genérica.

Parâmetros do URL

O URL deve conter os seguintes parâmetros, que diferenciam maiúsculas de minúsculas:

- região — Especifique uma AWS região. Para obter uma lista de regiões compatíveis com o Elastic Beanstalk, consulte AWS Elastic Beanstalk Endpoints e cotas no Referência geral da AWS.
- applicationName: especifique o nome da aplicação. O Elastic Beanstalk exibe o nome da aplicação no console do Elastic Beanstalk para distingui-la de outras aplicações. Por padrão, o nome do aplicativo também forma a base do nome e URL do ambiente.

- platform (plataforma): especifique a versão de plataforma a ser usada para o ambiente. Use um dos métodos a seguir e codifique sua escolha com URL:
  - Especifique um ARN de plataforma sem uma versão. O Elastic Beanstalk seleciona a versão da plataforma mais recente da versão principal da plataforma correspondente. Por exemplo, para selecionar a versão mais recente da plataforma Python 3.6, especifique `Python 3.6 running on 64bit Amazon Linux`.
  - Especifique o nome da plataforma. O Elastic Beanstalk seleciona a versão mais recente do runtime mais recente do idioma da plataforma (por exemplo, `Python`).

  Para obter uma descrição de todas as plataformas disponíveis e suas versões, consulte Plataformas compatíveis com Elastic Beanstalk.

  Você pode usar o AWS Command Line Interface(AWS CLI) para obter uma lista de todas as versões disponíveis da plataforma com suas respectivas ARNs. O comando `list-platform-versions` lista informações detalhadas sobre todas as versões da plataforma disponíveis. Use o argumento `--filters` para limitar o escopo da lista. Por exemplo, você pode listar todas as versões da plataforma de um idioma específico.

  O exemplo a seguir consulta todas as versões de plataforma Python e canaliza a saída por meio de uma série de comandos. O resultado é uma lista da versão da plataforma ARNs (sem a */*`version` cauda), em um formato legível por humanos, sem codificação de 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
```

  O exemplo a seguir adiciona um comando Perl ao último exemplo, para codificar a saída com URL.

```
$ 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
```

Um URL "Iniciar agora" pode, opcionalmente, conter os parâmetros a seguir. Se você não incluir os parâmetros opcionais em seu URL Launch Now (Iniciar agora), o Elastic Beanstalk usará valores padrão para criar e executar a aplicação. Quando você não inclui o sourceBundleUrlparâmetro, o Elastic Beanstalk usa o aplicativo de amostra padrão para a plataforma especificada.

- sourceBundleUrl— Especifique a localização do pacote de origem do seu aplicativo web em formato de URL. Por exemplo, se você carregou seu pacote de origem em um bucket do Amazon S3, você pode especificar o valor sourceBundleUrldo parâmetro como. `https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject`

  > **Note**
  >
  > Você pode especificar o valor do sourceBundleUrlparâmetro como uma URL HTTP, mas o navegador do usuário converterá caracteres conforme necessário aplicando a codificação de URL HTML.

- environmentType: especifique se o ambiente tem balanceamento de carga e é escalável ou se é de apenas uma única instância. Para obter mais informações, consulte Tipos de ambiente. Você pode especificar `LoadBalancing` ou `SingleInstance` como o valor do parâmetro.
- tierName: especifique se o ambiente é compatível com uma aplicação Web que processe solicitações da Web ou com uma aplicação Web que execute trabalhos em segundo plano. Para obter mais informações, consulte Ambientes de operador do Elastic Beanstalk. Você pode especificar `WebServer` ou `Worker`,
- instanceType: especifique um servidor com as características (incluindo o tamanho da memória e a capacidade da CPU) mais adequadas à aplicação. Para obter mais informações sobre famílias e tipos de EC2 instâncias da Amazon, consulte Tipos de instância no Guia EC2 do usuário da Amazon. Para obter mais informações sobre os tipos de instância disponíveis em todas as regiões, consulte Tipos de instância disponíveis no Guia EC2 do usuário da Amazon.
- withVpc: especifique se o ambiente será criado em uma Amazon VPC. Você pode especificar `true` ou `false`. Para obter mais informações sobre como usar o Elastic Beanstalk com a Amazon VPC, consulte Usar o Elastic Beanstalk com Amazon VPC.

- withRds: especifique uma instância do banco de dados do Amazon RDS será criada com esse ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon RDS. Você pode especificar `true` ou `false`.
- rds DBEngine — Especifique o mecanismo de banco de dados que você deseja usar para suas EC2 instâncias da Amazon nesse ambiente. Especifique `mysql`, `oracle-sel`, `sqlserver-ex`, `sqlserver-web` ou `sqlserver-se`. O valor padrão é `mysql`.
- DBAllocatedArmazenamento rds — Especifique o tamanho de armazenamento do banco de dados alocado em gigabytes (GB). Especifique os seguintes valores:
  - MySQL: `5` a `1024`. O padrão é `5`.
  - Oracle: `10` a `1024`. O padrão é `10`.
  - Microsoft SQL Server Express Edition – `30`.
  - Microsoft SQL Server Web Edition – `30`.
  - Microsoft SQL Server Standard Edition – `200`.
- DBInstanceClasse rds — Especifique o tipo de instância do banco de dados. O valor padrão é `db.t2.micro` (`db.m1.large` é para um ambiente que não esteja em execução em uma Amazon VPC). Para obter uma lista de classes de instância de banco de dados suportadas pelo Amazon RDS, consulte DB Instance Class (Classe de instância de banco de dados) no Amazon Relational Database Service User Guide (Guia do usuário do Amazon Relational Database Service).
- RDSmulti AZDatabase — Especifique se o Elastic Beanstalk precisa criar a instância do banco de dados em várias zonas de disponibilidade. É possível especificar `true` ou `false`. Para obter mais informações sobre implantações em várias zonas de disponibilidade com o Amazon RDS, acesse Regions and Availability Zones (Regiões e zonas de disponibilidade) no Amazon Relational Database Service User Guide (Guia do usuário do Amazon Relational Database Service).
- DBDeletionPolítica rds — Especifique se deseja excluir ou capturar um instantâneo da instância do banco de dados no encerramento do ambiente. É possível especificar `Delete` ou `Snapshot`.

Exemplo

Um exemplo de URL "Iniciar agora" está apresentado a seguir. Depois de criar seu próprio URL, você pode fornecê-lo aos usuários. Por exemplo, você pode incorporar a URL em uma página da Web ou em materiais de treinamento. Quando os usuários criam uma aplicação usando o URL Launch Now (Iniciar agora), o assistente de criação de aplicação do Elastic Beanstalk não requer entrada adicional.

```
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
```

Para usar o URL "Iniciar agora"

1. Escolha a URL Launch Now (Iniciar agora).
2. Quando o console do Elastic Beanstalk é aberto, na página Create a web app (Criar uma aplicação Web), escolha Review and launch (Revisar e iniciar) para visualizar as configurações que o Elastic Beanstalk usa para criar a aplicação e iniciar o ambiente no qual a aplicação é executada.
3. Na página Configure (Configurar), clique em Create app (Criar aplicativo) para criar o aplicativo.

Criar e atualizar grupos de ambientes do Elastic Beanstalk

Com a AWS Elastic Beanstalk ComposeEnvironmentsAPI, você pode criar e atualizar grupos de ambientes do Elastic Beanstalk em um único aplicativo. Cada ambiente do grupo pode executar um componente separado de um aplicativo de arquitetura orientada a serviços. A API `Compose Environments` usa uma lista de versões do aplicativo e um nome de grupo opcional. O Elastic Beanstalk cria um ambiente para cada versão do aplicativo ou, se os ambientes já existem, implanta as versões do aplicativo neles.

Crie links entre os ambientes do Elastic Beanstalk para designar um ambiente como uma dependência do outro. Quando você cria um grupo de ambientes com a API `Compose Environments`, o Elastic Beanstalk cria ambientes dependentes somente depois que suas dependências estão em execução. Para obter mais informações sobre links de ambiente, consulte Criar links entre ambientes do Elastic Beanstalk.

A API `Compose Environments` usa um manifesto de ambiente para armazenar detalhes de configuração que são compartilhados por grupos de ambientes. Cada aplicativo do componente deve ter um arquivo de configuração `env.yaml` em seu pacote de origem do aplicativo que especifique os parâmetros usados para criar o ambiente.

`Compose Environments` requer que `EnvironmentName` e `SolutionStack` sejam especificados no manifesto de ambiente para cada aplicativo do componente.

Você pode usar a `Compose Environments` API com a interface de linha de comando do Elastic Beanstalk (EB CLI), o ou um SDK. AWS CLI Para ver as instruções da EB CLI, consulte Gerenciar vários ambientes do Elastic Beanstalk como um grupo com a CLI do EB.

Uso da API **`Compose Environments`**

Por exemplo, você pode criar uma aplicação chamada `Media Library` que permita aos usuários fazer upload e gerenciar imagens e vídeos armazenados no Amazon Simple Storage Service (Amazon S3). O aplicativo tem um ambiente de front-end, `front`, que executa um aplicativo Web. Isso permite aos usuários fazer upload e download de arquivos individuais, visualizar sua biblioteca e iniciar trabalhos de processamento em lote.

Em vez de processar os trabalhos diretamente, a aplicação de front-end adiciona trabalhos a uma fila do Amazon SQS. O segundo ambiente, `worker`, obtém trabalhos da fila e os processa. O `worker` usa um tipo de instância G2, que tem uma GPU de alta performance e, ao mesmo tempo, `front` pode ser executado em um tipo de instância genérico mais econômico.

Organize a pasta do projeto, `Media Library`, em diretórios separados para cada componente, com cada diretório contendo um arquivo de definição de ambiente (`env.yaml`) com o código-fonte de cada:

```
~/workspace/media-library
|-- front
|   `-- env.yaml
`-- worker
    `-- env.yaml
```

As seguintes listas mostram o arquivo `env.yaml` para cada aplicativo do 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
```

Depois de criar uma versão do aplicativo para os componentes do aplicativo de front-end (`front-v1`) e operador (`worker-v1`), chame a API `Compose Environments` com os nomes de versão. Neste exemplo, usamos a AWS CLI para chamar a 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
```

A terceira chamada cria dois ambientes, `front-dev` e `worker-dev`. A API cria os nomes dos ambientes ao concatenar o `EnvironmentName` especificado no arquivo `env.yaml` com a opção `group name` especificada na chamada `Compose Environments`, separados por um hífen. O tamanho total dessas duas opções e o hífen não deve exceder o máximo permitido de 23 caracteres do nome do ambiente.

A aplicação em execução no ambiente `front-dev` pode acessar o nome da fila do Amazon SQS anexada ao ambiente `worker-dev` lendo a variável `WORKERQUEUE`. Para obter mais informações sobre links de ambiente, consulte Criar links entre ambientes do Elastic Beanstalk.

Gerenciar vários ambientes do Elastic Beanstalk como um grupo com a CLI do EB

Você pode usar o EB CLI para criar grupos de AWS Elastic Beanstalk ambientes, cada um executando um componente separado de um aplicativo de arquitetura orientada a serviços. O EB CLI gerencia esses grupos usando a ComposeEnvironmentsAPI.

> **Note**
>
> Grupos de ambientes não é a mesma coisa que vários contêineres em um ambiente do Docker de vários contêineres. Com grupos de ambientes, cada componente do seu aplicativo é executado em um ambiente separado do Elastic Beanstalk, com seu próprio conjunto dedicado de instâncias da Amazon. EC2 Cada componente pode ser escalado

separadamente. Com o Docker de vários contêineres, você combina vários componentes de um aplicativo em um único ambiente. Todos os componentes compartilham o mesmo conjunto de EC2 instâncias da Amazon, com cada instância executando vários contêineres Docker. Escolha uma dessas arquiteturas de acordo com as necessidades do seu aplicativo. Para obter detalhes sobre o Docker de vários contêineres, consulte Usando a ramificação da plataforma Docker gerenciada pelo ECS no Elastic Beanstalk.

Organize os componentes do aplicativo na seguinte estrutura de pastas:

```
~/project-name
|-- component-a
|   `-- env.yaml
`-- component-b
    `-- env.yaml
```

Cada subpasta contém o código-fonte de um componente independente de um aplicativo que será executado em seu próprio ambiente e um arquivo de definição do ambiente chamado `env.yaml`. Para ver detalhes sobre o formato do `env.yaml`, consulte Manifesto de ambiente (`env.yaml`).

Para usar a API `Compose Environments`, primeiro execute eb init da pasta do projeto, especificando cada componente pelo nome da pasta que o contém com a opção `--modules`:

```
~/workspace/project-name$ eb init --modules component-a component-b
```

A CLI do EB solicita a configuração de cada componente e criação do diretório `.elasticbeanstalk` em cada componente da pasta. A EB CLI não cria arquivos de configuração no diretório pai.

```
~/project-name
|-- component-a
|   |-- .elasticbeanstalk
|   `-- env.yaml
`-- component-b
    |-- .elasticbeanstalk
    `-- env.yaml
```

Em seguida, execute o comando eb create com uma lista de ambientes para criar, um para cada componente:

```
~/workspace/project-name$ eb create --modules component-a component-b --env-group-suffix group-name
```

Esse comando cria um ambiente para cada componente. Os nomes dos ambientes são criados concatenando o `EnvironmentName` especificado no arquivo `env.yaml` com o nome do grupo, separados por um hífen. O tamanho total dessas duas opções e o hífen não deve exceder o máximo permitido de 23 caracteres do nome do ambiente.

Para atualizar o ambiente, use o comando eb deploy:

```
~/workspace/project-name$ eb deploy --modules component-a component-b
```

Você pode atualizar os componentes individualmente ou como um grupo. Especifique os componentes que deseja atualizar com a opção `--modules`.

A EB CLI armazena o nome do grupo que você usou com eb create na seção `branch-defaults` do arquivo de configuração da EB CLI, em `/.elasticbeanstalk/config.yml`. Para implantar o aplicativo em um grupo diferente, use a opção `--env-group-suffix` quando executar eb deploy. Se o grupo ainda não existe, a EB CLI cria um novo grupo de ambientes:

```
~/workspace/project-name$ eb deploy --modules component-a component-b --env-group-suffix group-2-name
```

Para encerrar ambientes, execute eb terminate na pasta de cada módulo. Por padrão, a EB CLI mostrará um erro se você tentar encerrar um ambiente do qual um outro ambiente em execução depende. Encerre o ambiente dependente primeiro ou use a opção `--ignore-links` para substituir o comportamento padrão:

```
~/workspace/project-name/component-b$ eb terminate --ignore-links
```

Implantar aplicações em ambientes do Elastic Beanstalk

Você pode usar o AWS Elastic Beanstalk console para carregar um pacote de origem atualizado e implantá-lo em seu ambiente do Elastic Beanstalk ou reimplantar uma versão carregada anteriormente.

Cada implantação é identificada por um ID. IDs A implantação começa em 1 e aumenta em um com cada alteração na configuração da implantação e da instância. Se você habilitar os relatórios

avançados de integridade, o Elastic Beanstalk exibirá o ID da implantação no console de integridade e na CLI do EB ao relatar o status de integridade da instância. O ID de implantação ajuda a determinar o estado do ambiente quando uma atualização contínua falha.

O Elastic Beanstalk fornece várias políticas e configurações de implantação. Para obter detalhes sobre como configurar uma política e configurações adicionais, consulte the section called “Opções de implantação”. A tabela a seguir lista as políticas e os tipos de ambientes compatíveis com elas.

Políticas de implantação compatíveis

Política de implantação	Ambientes de carga equilibrada	Ambientes de instância única	Ambientes legados do Windows Server†
Tudo de uma vez	✓ Sim	✓ Sim	✓ Sim
Contínua	✓ Sim	✗ Não	✓ Sim
Contínua com lote adicional	✓ Sim	✗ Não	✗ Não
Imutável	✓ Sim	✓ Sim	✗ Não
Divisão de tráfego	✓ Sim (Application Load Balancer)	✗ Não	✗ Não

† Nessa tabela, um Ambiente legado do Windows Server é um ambiente baseado em uma configuração de plataforma do Windows Server que usa uma versão anterior do IIS 8.5.

> **Warning**
>
> Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Isso faz com que todos os saldos EC2 estourados acumulados da Amazon sejam perdidos. Isso acontece nos seguintes casos:
>
> - Atualizações de plataforma gerenciada com substituição de instância habilitada
> - Atualizações imutáveis
> - Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Como escolher uma política de implantação

Escolher a política de implantação correta para seu aplicativo é uma questão com algumas considerações a fazer e que depende de suas necessidades específicas. A página the section called "Opções de implantação" tem mais informações sobre cada política e uma descrição detalhada do funcionamento de algumas delas.

A lista a seguir fornece informações resumidas sobre as diferentes políticas de implantação e inclui considerações relacionadas.

- All at once (Tudo de uma vez): o método de implantação mais rápido. Adequado se você puder aceitar uma pequena perda de serviço e se implantações rápidas forem importantes para você. Com esse método, o Elastic Beanstalk implanta a nova versão da aplicação em cada instância. Em seguida, talvez o proxy da web ou o servidor do aplicativo precise reiniciar. Como resultado, seu aplicativo pode ficar indisponível para usuários (ou ter baixa disponibilidade) por um breve período.
- Rolling (Contínua): evita o tempo de inatividade e minimiza a disponibilidade reduzida, a um custo de implantação mais longo. Adequado se você não puder aceitar nenhum período de perda total de serviço. Com esse método, seu aplicativo é implantado no ambiente um lote de instâncias por vez. A maior parte da largura de banda é mantida durante toda a implantação.
- Rolling with additional batch (Contínua com lote adicional): evita qualquer disponibilidade reduzida, ao custo de um tempo de implantação ainda mais longo em comparação com o método Rolling (Contínuo). Adequado se você precisar manter a mesma largura de banda durante toda a implantação. Com esse método, o Elastic Beanstalk inicia um lote extra de instâncias e executa uma implantação contínua. A inicialização do lote extra leva tempo e garante que a mesma largura de banda seja mantida durante toda a implantação.
- Immutable (Imutável): um método de implantação mais lento, que garante que a nova versão da aplicação seja sempre implantada em novas instâncias, em vez de atualizar instâncias existentes. Ele também tem a vantagem adicional de uma reversão rápida e segura em caso de falha na implantação. Com esse método, o Elastic Beanstalk executa uma atualização imutável para implantar sua aplicação. Em uma atualização imutável, um segundo grupo de Auto Scaling é iniciado no ambiente e a nova versão atende o tráfego junto com a versão antiga até que a nova instância passe nas verificações de integridade.
- Traffic splitting (Divisão de tráfego): um método de implantação de testes canary. Adequado se você deseja testar a integridade da sua nova versão do aplicativo usando uma parte do tráfego recebido, mantendo o restante do tráfego atendido pela versão antiga do aplicativo.

A tabela a seguir compara as propriedades dos métodos de implantação.

Métodos de implantação

Método	Impacto de uma implantação malsucedida	Tempo de implantação	Tempo de inatividade de zero	Nenhuma alteração de DNS	Processo de reversão	Código implantado em
Tudo de uma vez	Tempo de inatividade	🕒	✗ Não	✓ Sim	Reimplantação manual	Instâncias existentes
Contínua	Lote único fora de serviço; qualquer lote bem-sucedido antes de uma falha ao executar uma nova versão do aplicativo	🕒🕒	✓ Sim	✓ Sim	Reimplantação manual	Instâncias existentes
Contínua com lote adicional	Mínimo, se o primeiro lote falhar; do contrário, é semelhante à Rolling (Contínua)	🕒🕒	✓ Sim	✓ Sim	Reimplantação manual	Instâncias novas e existentes
Imutável	Mínimo	🕒🕒	✓ Sim	✓ Sim	Encerra novas instâncias	Instâncias novas
Divisão de tráfego	Percentual de tráfego do cliente roteado para a nova versão afetada temporariamente	🕒🕒	✓ Sim	✓ Sim	Redirecionar tráfego e encerra novas	Instâncias novas

Método	Impacto de uma implantação malsucedida	Tempo de implantação	Tempo de inatividade de zero	Nenhuma alteração de DNS	Processo de reversão	Código implantado em
					instâncias	
Azul/verde	Mínimo		✓ Sim	X Não	Alternar URL	Instâncias novas

† Varia, dependendo do tamanho do lote.

†† Varia de acordo com a configuração da opção de tempo momento de avaliação.

Implantar uma nova versão do aplicativo

Você pode executar as implantações pelo painel do ambiente.

Como implantar uma nova versão da aplicação em um ambiente do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Upload and deploy (Fazer upload e implantar).
4. Use o formulário na tela para carregar o pacote de origem do aplicativo.
5. Escolha Implantar.

Reimplantar a versão anterior

Você também pode implantar uma versão do aplicativo carregado anteriormente em qualquer um dos ambientes da página de versões do aplicativo.

Para implantar uma versão existente do aplicativo no ambiente atual

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

3. No painel de navegação, encontre o nome do aplicativo e escolha Application versions (Versões do aplicativo).
4. Selecione a versão do aplicativo a ser implantada.
5. Escolha Actions (Ações) e escolha Deploy (Implantar).
6. Escolha um ambiente e escolha Deploy (Implantar).

Outras formas de implantar seu aplicativo

Se você implanta com frequência, considere o uso da Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB) para gerenciar os ambientes. A CLI do EB cria um repositório junto com o código-fonte. Ela também pode criar um pacote de origem, carregá-lo no Elastic Beanstalk e implantá-lo com um único comando.

Para implantações dependentes das alterações da configuração dos recursos ou de uma nova versão que não pode ser executada junto com a versão antiga, inicie um novo ambiente com a nova versão e faça uma troca de CNAME para uma implantação azul/verde.

Políticas e configurações de implantação

AWS Elastic Beanstalk fornece várias opções de como as implantações são processadas, incluindo políticas de implantação (tudo de uma vez, contínuo, contínuo com lote adicional, imutável e divisão de tráfego) e opções que permitem configurar o tamanho do lote e o comportamento da verificação de integridade durante as implantações. Por padrão, seu ambiente usa all-at-once implantações. Se você criou o ambiente com a CLI do EB e se ele for um ambiente de escalável (você não especificou a opção `--single`), ele usa implantações contínuas.

Com implantações contínuas, o Elastic Beanstalk divide as instâncias EC2 da Amazon do ambiente em lotes e implanta a nova versão do aplicativo em um lote por vez. Ele deixa o restante das instâncias no ambiente executando a versão antiga do aplicativo. Durante uma implantação contínua, algumas instâncias atenderão a solicitações com a versão antiga do aplicativo, enquanto as instâncias em lotes completos atenderão a outras solicitações com a nova versão. Para obter detalhes, consulte the section called “Como funcionam as implantações gradativas”.

Para manter a capacidade total durante as implantações, você pode configurar o ambiente para iniciar um novo lote de instâncias antes de tirar qualquer instância de serviço. Essa opção é conhecida como uma implantação contínua com um lote adicional. Quando a implantação for concluída, o Elastic Beanstalk encerrará os lotes adicionais de instâncias.

As implantações imutáveis executam uma atualização imutável para iniciar um conjunto completo de novas instâncias executando a nova versão da aplicação em outro grupo de Auto Scaling junto com as instâncias que executam a versão antiga. As implantações imutáveis podem impedir problemas causados por implantações contínuas parcialmente concluídas. Se as novas instâncias não passarem pela verificação de integridade, o Elastic Beanstalk as encerrará, deixando as instâncias originais inalteradas.

As implantações de divisão de tráfego permitem executar testes canary como parte da implantação de aplicativos. Em uma implantação de divisão de tráfego, o Elastic Beanstalk executa um conjunto completo de novas instâncias, como durante uma implantação imutável. Depois, ele encaminha uma porcentagem especificada de tráfego de entrada do cliente para a nova versão do aplicativo por um período de avaliação especificado. Se as novas instâncias permanecerem íntegras, o Elastic Beanstalk encaminhará todo o tráfego para elas e encerrará as antigas. Se as novas instâncias não passarem nas verificações de integridade ou se você optar por interromper a implantação, o Elastic Beanstalk moverá o tráfego de volta para as instâncias antigas e encerrará as novas. Nunca há interrupção de serviço. Para obter detalhes, consulte the section called “Como funcionam as implantações de divisão de tráfego”.

> **Warning**
>
> Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Isso faz com que todos os saldos EC2 estourados acumulados da Amazon sejam perdidos. Isso acontece nos seguintes casos:
>
> - Atualizações de plataforma gerenciada com substituição de instância habilitada
> - Atualizações imutáveis
> - Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Se o seu aplicativo não passar em todas as verificações de integridade, mas ainda estiver funcionando corretamente em um status de integridade inferior, você poderá permitir que as instâncias passem pelas verificações de integridade com um status inferior, como `Warning`, modificando a opção Limite de integridade. Se suas implantações falham por não passarem nas verificações de integridade e você precisar forçar uma atualização, independentemente do status de integridade, especifique a opção Ignorar verificação de integridade.

Ao especificar um tamanho de lote para atualizações constantes, o Elastic Beanstalk também usa esse valor para implantar reinicializações de aplicações. Use as reinicializações contínuas

quando precisar reiniciar o proxy e os servidores de aplicativos em execução em suas instâncias do ambiente sem tempo de inatividade.

Configurar implantações de aplicativos

No console de gerenciamento do ambiente, ative e configure as implantações em lote de versão do aplicativo editando Atualizações e implantações na página Configuração do ambiente.

Para configurar as implantações (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Rolling updates and deployments (Atualizações e implantações contínuas), escolha Edit (Editar).
5. Na seção Application Deployments (Implantações de aplicativos), escolha uma Deployment policy (Política de implantação), as configurações de lote e as opções de verificação de integridade.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A seção Implantações de aplicativos da página Atualizações e implantações contínuas tem as seguintes opções para implantações de aplicativos:

- Deployment policy (Política de implantação): escolha uma das seguintes opções de implantação:
  - All at once (Todos de uma vez): implante a nova versão em todas as instâncias simultaneamente. Todas as instâncias em seu ambiente ficam fora de serviço por um curto período durante a implantação.
  - Rolling (Contínua): implante a nova versão em lotes. Cada lote é retirado de serviço durante a fase de implantação, reduzindo a capacidade de seu ambiente de acordo com o número de instâncias em um lote.
  - Rolling with additional batch (Contínua com lote adicional): implante a nova versão em lotes, mas primeiro inicie um novo lote de instâncias para garantir a capacidade total durante o processo de implantação.
  - Immutable (Imutável): implante a nova versão em um novo grupo de instâncias executando uma atualização imutável.

- Traffic splitting (Divisão de tráfego): implante a nova versão em um novo grupo de instâncias e divida temporariamente o tráfego de entrada do cliente entre a versão existente da aplicação e a nova.

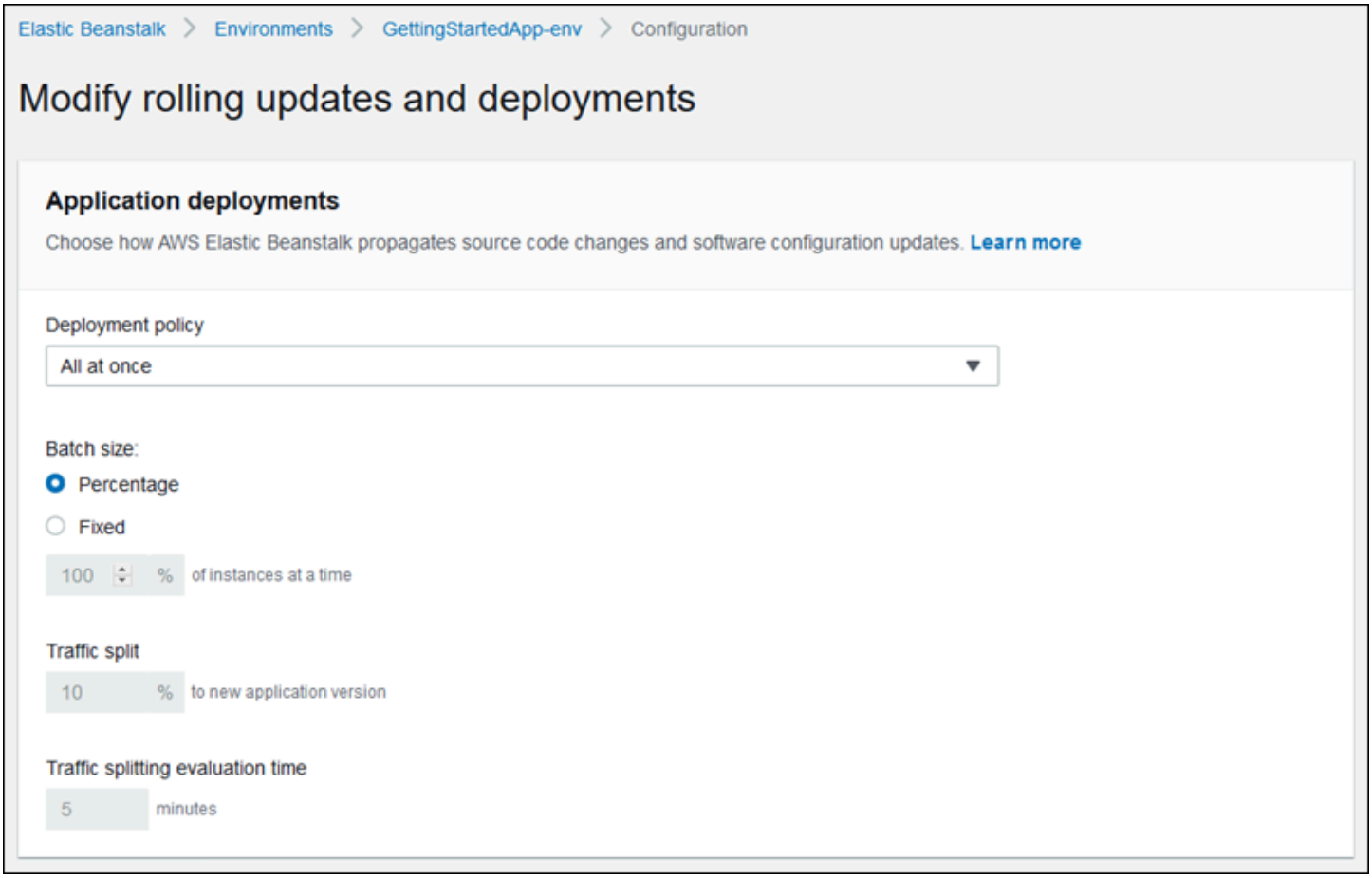
Para as políticas Contínua e Contínua com lote adicional, você pode configurar:

- Batch size (Tamanho do lote): o tamanho do conjunto de instâncias para implantar em cada lote.

  Escolha Porcentagem para configurar uma porcentagem do número total de EC2 instâncias no grupo Auto Scaling (até 100 por cento) ou escolha Fixo para configurar um número fixo de instâncias (até a contagem máxima de instâncias na configuração do Auto Scaling do seu ambiente).

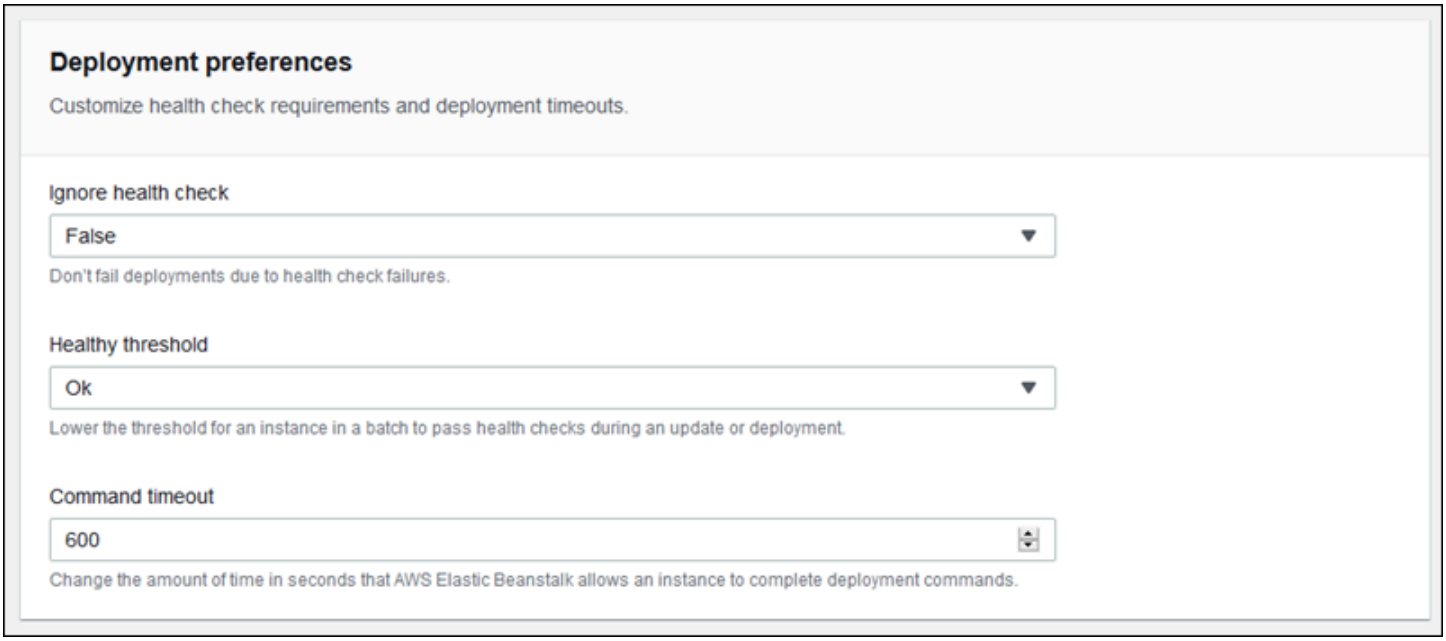
Para a política de implantação de divisão de tráfego, você pode fazer a seguinte configuração:

- Traffic split (Divisão de tráfego): a porcentagem inicial de tráfego de entrada do cliente que o Elastic Beanstalk desloca para instâncias de ambiente executando a nova versão da aplicação que está sendo implantada.
- Traffic splitting evaluation time (Tempo de avaliação da divisão de tráfego): o período, em minutos, que o Elastic Beanstalk aguarda após uma implantação íntegra inicial antes de prosseguir para deslocar todo o tráfego de entrada de cliente para a nova versão da aplicação que está sendo implantada.



A seção Deployment preferences (Preferências de implantação) contém opções relacionadas a verificações de integridade.

- Ignore health check (Ignorar verificação de integridade): impede que uma implantação seja revertida quando um lote não fica íntegro dentro do Command timeout (Tempo limite do comando).
- Healthy threshold (Limite de integridade): reduz o limite no qual uma instância é considerada íntegra durante as implantações e atualizações contínuas e as atualizações imutáveis.
- Command timeout (Tempo limite do comando): o número de segundos de espera para que uma instância se torne íntegra antes de cancelar a implantação ou, se a opção Ignore health check (Ignorar verificação de integridade) estiver definida, prosseguir para o próximo lote.



Como funcionam as implantações gradativas

Ao processar um lote, o Elastic Beanstalk separa todas as instâncias no lote do balanceador de carga, implanta a nova versão da aplicação e, depois, anexa as instâncias novamente. Se você ativar a drenagem da conexão, o Elastic Beanstalk drenará as conexões existentes das instâncias da EC2 Amazon em cada lote antes de iniciar a implantação.

Depois de anexar novamente as instâncias em um lote ao balanceador de carga, o Elastic Load Balancing espera até que elas passem em um número mínimo de verificações de integridade do Elastic Load Balancing (o valor Healthy check count threshold (Limite de contagem de verificações de integridade)) e, depois, inicia o roteamento de tráfego para elas. Se nenhum URL de verificação de integridade estiver configurado, isso poderá acontecer muito rapidamente, porque uma instância passará pela verificação de integridade assim que puder aceitar uma conexão TCP. Se um URL de verificação de integridade estiver configurado, o load balancer não roteará o tráfego para as instâncias atualizadas até que retornem um código de status `200 OK` em resposta a uma solicitação `HTTP GET` para o URL de verificação de integridade.

O Elastic Beanstalk aguarda até que todas as instâncias em um lote estejam íntegras antes de avançar para o próximo lote. Com os relatórios básicos de integridade, a integridade da instância depende do status da verificação de integridade do Elastic Load Balancing. Quando todas as instâncias no lote passarem em verificações de integridade suficientes para serem consideradas íntegras pelo Elastic Load Balancing, o lote estará concluído. Se os relatórios aprimorados de integridade estiverem habilitados, o Elastic Beanstalk considerará vários outros fatores, incluindo o

resultado de solicitações recebidas. Com os relatórios de integridade avançados, todas as instâncias devem passar por 12 verificações de integridade consecutivas com status OK dentro de dois minutos para ambientes de servidor web e 18 verificações de integridade dentro de três minutos para ambientes de operador.

Se um lote de instâncias não se tornar íntegro dentro do tempo limite do comando, a implantação falhará. Depois de uma implantação com falha, verifique a integridade das instâncias no ambiente para obter informações sobre a causa da falha. Em seguida, execute outra implantação com uma versão corrigida ou comprovadamente boa de seu aplicativo para reverter.

Se uma implantação falhar após um ou mais lotes serem concluídos com êxito, os lotes concluídos executar a nova versão de seu aplicativo enquanto os eventuais lotes pendentes continuam a executar a versão antiga. Você pode identificar a versão em execução nas instâncias em seu ambiente na página de integridade no console. Essa página exibe o ID da implantação mais recente executada em cada instância em seu ambiente. Se você encerrar instâncias a partir da implantação com falha, o Elastic Beanstalk as substituirá por instâncias executando a versão da aplicação a partir da implantação bem-sucedida mais recente.

Como funcionam as implantações de divisão de tráfego

As implantações de divisão de tráfego permitem a execução de testes canary. Você direciona algum tráfego de entrada do cliente para a nova versão do aplicativo de modo a verificar a integridade do aplicativo antes de confirmar a nova versão e direcionar todo o tráfego para ele.

Durante uma implantação de divisão de tráfego, o Elastic Beanstalk cria um conjunto de instâncias em um grupo de Auto Scaling temporário separado. Depois, o Elastic Beanstalk instrui o balanceador de carga a direcionar uma certa porcentagem do tráfego de entrada do ambiente para as novas instâncias. Depois, durante um período configurado, o Elastic Beanstalk rastreia a integridade do novo conjunto de instâncias. Se tudo estiver bem, o Elastic Beanstalk vai deslocar o tráfego restante para as novas instâncias e anexá-las ao grupo de Auto Scaling original do ambiente, substituindo as instâncias antigas. Depois, o Elastic Beanstalk limpa, ou seja, encerra as instâncias antigas e remove o grupo temporário de Auto Scaling.

> **Note**
>
> A capacidade do ambiente não muda durante uma implantação de divisão de tráfego. O Elastic Beanstalk inicia o mesmo número de instâncias no grupo temporário que há no grupo original de Auto Scaling no momento em que a implantação é iniciada. Ele mantém um número constante de instâncias nos dois grupos de Auto Scaling durante a implantação.

Leve esse fato em consideração ao definir o tempo de avaliação da divisão de tráfego do ambiente.

A reversão da implantação para a versão anterior do aplicativo é rápida e não afeta o serviço para o tráfego do cliente. Se as novas instâncias não passarem nas verificações de integridade ou se você optar por interromper a implantação, o Elastic Beanstalk moverá o tráfego de volta para as instâncias antigas e encerrará as novas. É possível cancelar implantações usando a página de visão geral do ambiente no console do Elastic Beanstalk e escolher Abort current operation (Anular operação atual) em Environment actions (Ações de ambiente). Você também pode chamar a AbortEnvironmentUpdateAPI ou o AWS CLI comando equivalente.

As implantações de divisão de tráfego exigem um Application Load Balancer. O Elastic Beanstalk usa esse tipo de balanceador de carga por padrão quando você cria seu ambiente usando o console do Elastic Beanstalk ou a CLI do EB.

Namespaces da opção de implantação

Você pode usar as opções de configuração no namespace `aws:elasticbeanstalk:command` para configurar suas implantações. Se você escolher a política de divisão de tráfego, opções adicionais para essa política estarão disponíveis no namespace `aws:elasticbeanstalk:trafficsplitting`.

Use a opção `DeploymentPolicy` para definir o tipo de implantação. Os valores a seguir são aceitos:

- `AllAtOnce`: desabilita implantações contínuas e sempre implanta em todas as instâncias simultaneamente.
- `Rolling`: habilita as implantações contínuas padrão.
- `RollingWithAdditionalBatch`: inicia um lote adicional de instâncias antes de iniciar a implantação para manter a capacidade total.
- `Immutable`: executa uma atualização imutável para cada implantação.
- `TrafficSplitting`: executa implantações de divisão de tráfego para fazer testes canary nas suas implantações de aplicações.

Quando você habilitar as implantações constantes, defina as opções `BatchSize` e `BatchSizeType` para configurar o tamanho de cada lote. Por exemplo, para implantar 25% de todas as instâncias em cada lote, especifique as opções e os valores a seguir.

Example .ebextensions/rolling-updates.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Rolling
    BatchSizeType: Percentage
    BatchSize: 25
```

Para implantar em cinco instâncias em cada lote, independentemente do número de instâncias em execução, e para abrir mais um lote de cinco instâncias executando a nova versão antes de colocar qualquer instância fora de serviço, especifique as opções e os valores a seguir.

Example .ebextensions/rolling-additionalbatch.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: RollingWithAdditionalBatch
    BatchSizeType: Fixed
    BatchSize: 5
```

Para executar uma atualização imutável para cada implantação com um limite de verificação de integridade Aviso e continuar com a implantação, mesmo se as instâncias em um lote não passarem nas verificações de integridade em um limite de tempo de 15 minutos, especifique as seguintes opções e valores.

Example .ebextensions/immutable-ignorehealth.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
    HealthCheckSuccessThreshold: Warning
    IgnoreHealthCheck: true
    Timeout: "900"
```

Para executar implantações de divisão de tráfego, encaminhando 15% do tráfego do cliente para a nova versão do aplicativo e avaliando a integridade por 10 minutos, especifique as seguintes opções e valores.

Example .ebextensions/traffic-splitting.config

```
option_settings:
  aws:elasticbeanstalk:command:
    DeploymentPolicy: TrafficSplitting
  aws:elasticbeanstalk:trafficsplitting:
    NewVersionPercent: "15"
    EvaluationTime: "10"
```

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Implantações azuis/verdes com o Elastic Beanstalk

Como AWS Elastic Beanstalk executa uma atualização no local quando você atualiza as versões do aplicativo, seu aplicativo pode ficar indisponível para os usuários por um curto período de tempo. Para evitar isso, execute uma implantação azul/verde. Para fazer isso, implante a nova versão em um ambiente separado e, em seguida, troque os CNAMEs dois ambientes para redirecionar o tráfego para a nova versão instantaneamente.

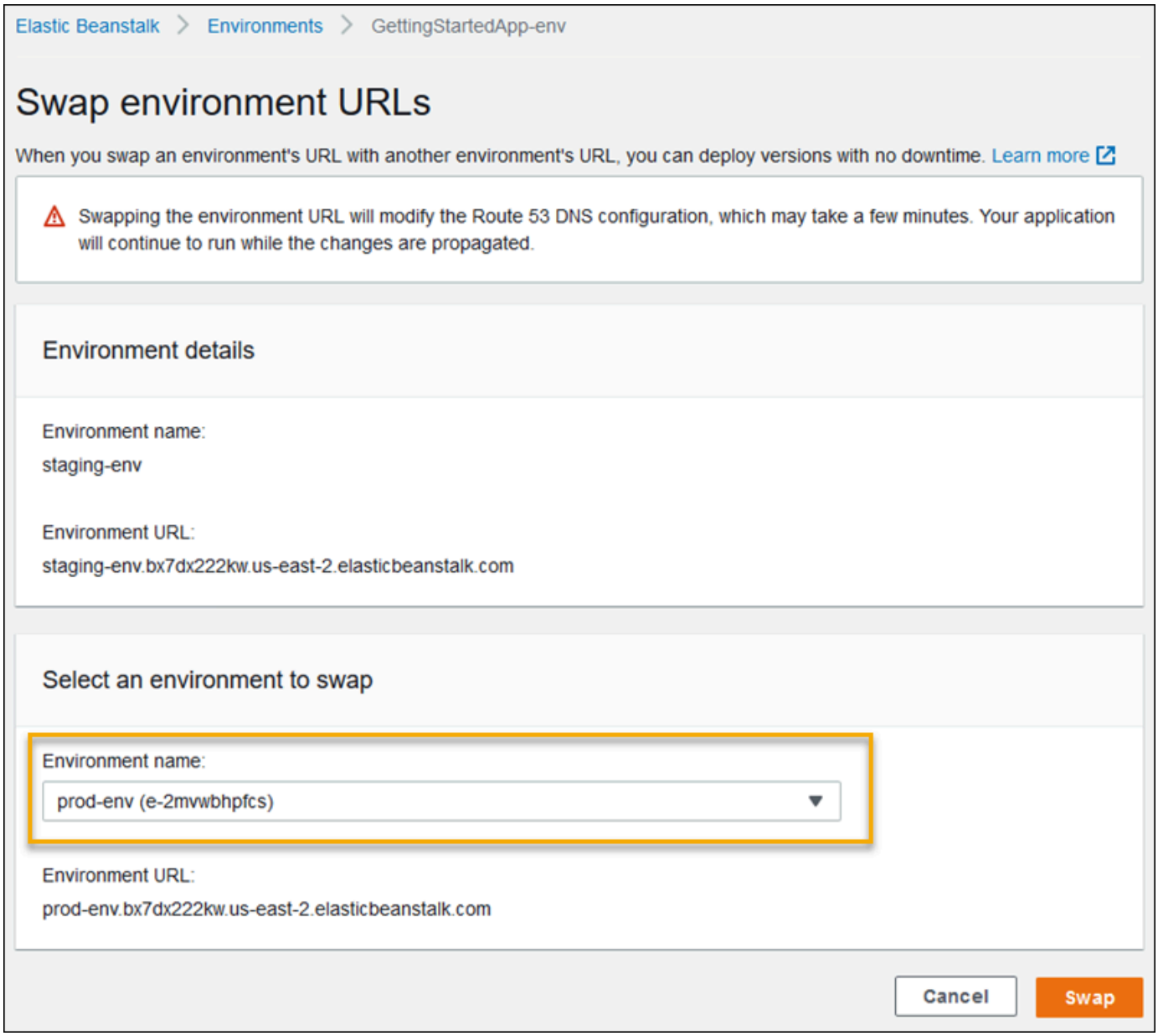
Uma implantação azul/verde também é necessária quando você deseja atualizar um ambiente para uma versão de plataforma incompatível. Para obter mais informações, consulte the section called “Atualizações da plataforma”.

As implantações azuis/verdes exigem que o ambiente seja executado de forma independente do seu banco de dados de produção, caso a aplicação use um. Se seu ambiente incluir um banco de dados criado pelo Elastic Beanstalk em seu nome, o banco de dados e a conexão do ambiente não serão preservados, a menos que você realize ações específicas. Se você tiver um banco de dados que deseja reter, use uma das opções de ciclo de vida do banco de dados do Elastic Beanstalk. Você pode escolher a opção Reter para manter o banco de dados e o ambiente operacionais depois de desacoplar o banco de dados. Para obter mais informações, consulte Ciclo de vida do banco de dados no capítulo Configuração de ambientes deste guia.

Para instruções sobre como configurar sua aplicação para se conectar a uma instância externa do Amazon RDS (não gerenciada pelo Elastic Beanstalk), consulte Usar o Elastic Beanstalk com o Amazon RDS.

Para executar uma implantação azul/verde

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. Clone seu ambiente atual ou inicie um novo ambiente para executar a versão de plataforma desejada.
3. Implante a nova versão da aplicação no novo ambiente.
4. Teste a nova versão no novo ambiente.
5. Na página de visão geral do ambiente, escolha Ações e, em seguida, escolha Ambiente de troca. URLs
6. Em Environment name (Nome do ambiente), selecione o ambiente atual.



7. Escolha Swap (Trocar).

O Elastic Beanstalk troca os registros de CNAME dos ambientes antigo e novo redirecionando o tráfego da versão antiga para a nova versão.

Depois que o Elastic Beanstalk concluir a operação de troca, verifique se o novo ambiente responde quando você tenta se conectar ao URL do ambiente antigo. No entanto, não encerre o ambiente antigo até as alterações de DNS serem propagadas e seus antigos registros DNS expirarem. Os servidores do DNS nem sempre limpam os registros antigos do cache com base na vida útil (TTL) que você define nos registros do DNS.

Alterações de configuração

Quando você modifica as definições das opções de configuração na seção Configuração do console de gerenciamento do ambiente, a alteração é AWS Elastic Beanstalk propagada para todos os recursos afetados. Esses recursos incluem o balanceador de carga que distribui o tráfego para as EC2 instâncias da Amazon que executam seu aplicativo, o grupo Auto Scaling que gerencia essas instâncias e EC2 as próprias instâncias.

Várias alterações de configuração podem ser aplicadas a um ambiente em execução sem substituir as instâncias existentes. Por exemplo, a configuração de um URL de verificação de integridade aciona uma atualização de ambiente para modificar as configurações do load balancer, mas não causa tempo de inatividade, porque as instâncias que executam seu aplicativo continuam atendendo às solicitações enquanto a atualização é propagada.

As alterações de configuração que modificam a configuração de execução ou configurações de VPC exigem o encerramento de todas as instâncias no ambiente e a substituição. Por exemplo, quando você altera o tipo de instância ou a configuração da chave SSH do seu ambiente, as EC2 instâncias devem ser encerradas e substituídas. O Elastic Beanstalk fornece várias políticas que determinam como essa substituição é feita.

- Atualizações contínuas: o Elastic Beanstalk aplica as alterações de configuração em lotes, mantendo um número mínimo de instâncias em execução e distribuindo o tráfego o tempo todo. Essa abordagem evita o tempo de inatividade durante o processo de atualização. Para obter detalhes, consulte Atualizações contínuas.
- Atualizações imutáveis: o Elastic Beanstalk inicia um grupo temporário de Auto Scaling fora do ambiente com um conjunto separado de instâncias em execução com a nova configuração. Depois, o Elastic Beanstalk coloca essas instâncias atrás do balanceador de carga do ambiente.

As instâncias antigas e novas distribuem o tráfego até que as novas instâncias passem nas verificações de integridade. Naquele momento, o Elastic Beanstalk move as novas instâncias para o grupo de Auto Scaling do ambiente e encerra o grupo temporário e as instâncias antigas. Para obter detalhes, consulte Atualizações imutáveis.

- Desativado: o Elastic Beanstalk não tenta evitar o tempo de inatividade. Ele encerra as instâncias existentes do ambiente e as substitui por novas instâncias em execução com a nova configuração.

> **Warning**
>
> Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Isso faz com que todos os saldos EC2 estourados acumulados da Amazon sejam perdidos. Isso acontece nos seguintes casos:
>
> - Atualizações de plataforma gerenciada com substituição de instância habilitada
> - Atualizações imutáveis
> - Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Tipos de atualização compatíveis

Contínua da configuração de atualização	Ambientes de carga equilibrada	Ambientes de instância única	Ambientes legados do Windows Server†
Desabilitado	✓ Sim	✓ Sim	✓ Sim
Contínua com base na integridade	✓ Sim	✗ Não	✓ Sim
Contínua com base no tempo	✓ Sim	✗ Não	✓ Sim
Imutável	✓ Sim	✓ Sim	✗ Não

† Para a finalidade dessa tabela, um ambiente legado do Windows Server é um ambiente com base em uma configuração de plataforma do Windows Server que usa uma versão anterior do IIS 8.5.

Tópicos

- Atualizações de configuração de ambiente contínuas do Elastic Beanstalk
- Atualizações de ambiente imutáveis

Atualizações de configuração de ambiente contínuas do Elastic Beanstalk

Quando uma alteração de configuração exige a substituição de instâncias, o Elastic Beanstalk pode executar a atualização em lotes para evitar tempo de inatividade enquanto a modificação é propagada. Durante uma atualização contínua, a capacidade só é reduzida pelo tamanho de um único lote, o qual você pode configurar. O Elastic Beanstalk coloca um lote de instâncias fora de serviço e as encerra, para, depois, iniciar um lote com a nova configuração. Depois que o novo lote começar a atender a solicitações, o Elastic Beanstalk passa para o próximo lote.

Os lotes de atualização contínua da configuração podem ser processados periodicamente (com base no tempo), com um atraso entre cada lote ou de acordo com a integridade. Para atualizações contínuas baseadas no tempo, você pode configurar a quantidade de tempo que o Elastic Beanstalk aguarda depois de concluir a inicialização de um lote de instâncias antes de passar para o próximo. Essa pausa permite que seu aplicativo faça o bootstrap e comece a atender às solicitações.

De acordo com as atualizações contínuas baseadas na integridade, o Elastic Beanstalk aguarda até que as instâncias em um lote passem por verificações de integridade antes de mudar para o próximo lote. A integridade de uma instância é determinada pelo sistema de relatórios de integridade, o qual pode ser básico ou aprimorado. Com a integridade básica, um lote é considerado íntegro assim que todas as suas instâncias passem nas verificações de integridade do Elastic Load Balancing (ELB).

Com os relatórios de integridade aprimorada, todas as instâncias em um lote devem passar por várias verificações de integridade consecutivas antes que o Elastic Beanstalk vá para o próximo lote. Além das verificações de integridade do ELB, que verificam somente suas instâncias, a integridade aprimorada monitora os logs do aplicativo e o estado de outros recursos do ambiente. Em um ambiente de servidor web com integridade aprimorada, todas as instâncias devem passar por 12 verificações de integridade durante o período de dois minutos (18 verificações em três minutos para ambientes de operador). Se qualquer instância falhar em uma verificação de integridade, a contagem recomeça.

Se um lote não se tornar íntegro dentro do tempo limite da atualização contínua (o padrão é 30 minutos), a atualização será cancelada. O timeout de atualização contínua é uma opção de configuração que está disponível no namespace `aws:autoscaling:updatepolicy:rollingupdate`. Caso a aplicação não passe nas verificações de integridade com o status `Ok`, mas esteja estável em outro

nível, você pode definir a opção `HealthCheckSuccessThreshold` no namespace `aws:elasticbeanstalk:healthreporting:system` para alterar o nível no qual o Elastic Beanstalk considera uma instância como íntegra.

Se o processo de atualização contínua falhar, o Elastic Beanstalk iniciará outra atualização contínua para retornar à configuração anterior. Uma atualização contínua pode falhar devido a verificações de integridade com falha ou caso a execução de novas instâncias fizer com que você exceda as cotas na sua conta. Se você atingir uma cota no número de EC2 instâncias da Amazon, por exemplo, a atualização contínua pode falhar ao tentar provisionar um lote de novas instâncias. Nesse caso, a reversão também falhará.

Uma de reversão com falha encerra o processo de atualização e deixa o ambiente em um estado não íntegro. Os lotes não processados ainda estão executando instâncias com a configuração antiga, enquanto todos os lotes concluídos com êxito têm a nova configuração. Para corrigir um ambiente após uma reversão com falha, primeiro resolva o problema subjacente que causou a falha da atualização e, depois, inicie outra atualização do ambiente.

Um método alternativo é implantar a nova versão do aplicativo em um ambiente diferente e, depois, executar uma troca do CNAME para redirecionar o tráfego sem qualquer inatividade. Consulte Implantações azuis/verdes com o Elastic Beanstalk para obter mais informações.

Atualizações contínuas versus implantações contínuas

As atualizações contínuas ocorrem quando você altera as configurações que exigem que novas EC2 instâncias da Amazon sejam provisionadas para o seu ambiente. Isso inclui alterações na configuração do grupo de Auto Scaling como as configurações do tipo de instância e do par de chaves, bem como alterações nas configurações da VPC. Em uma atualização contínua, cada lote de instâncias é encerrado antes que um novo lote seja provisionado para substituí-lo.

As implantações contínuas ocorrem sempre que você implanta a aplicação e normalmente podem ser executadas sem a substituição de instâncias no ambiente. O Elastic Beanstalk coloca cada lote fora de serviço, implanta a nova versão da aplicação e, depois, coloca-o novamente em serviço.

A exceção é se você alterar as configurações que exigem a substituição da instância enquanto implanta uma nova versão do aplicativo. Por exemplo, se você alterar as configurações de nome da chave em um arquivo de configuração em seu pacote de origem e o implantar em seu ambiente, vai acionar uma atualização contínua. Em vez de implantar a nova versão do aplicativo para cada lote de instâncias existentes, um novo lote de instâncias é provisionado com a nova configuração. Nesse caso, uma implantação separada não ocorre, pois as novas instâncias são ativadas com a nova versão do aplicativo.

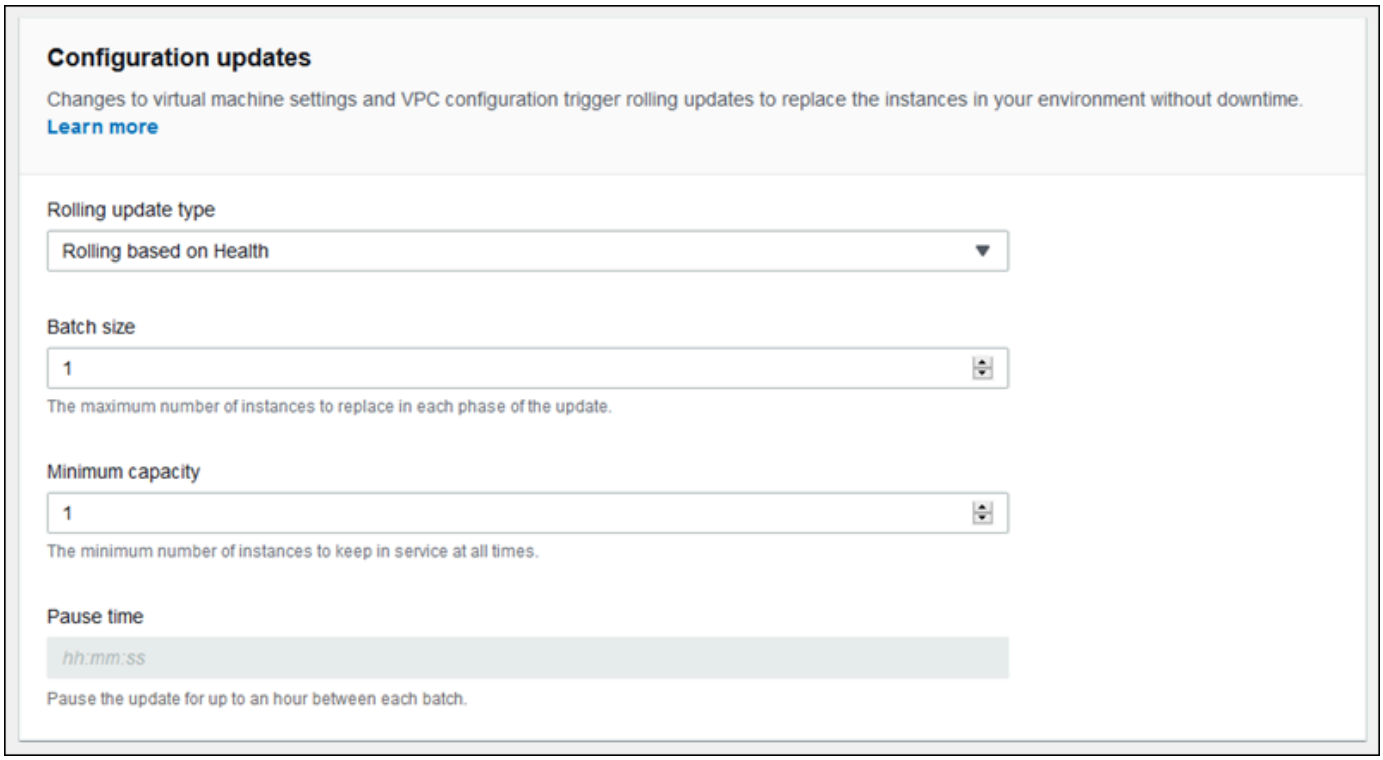
Sempre que novas instâncias são provisionadas como parte de uma atualização de ambiente, há uma fase de implantação em que o código-fonte do aplicativo é implantado nas novas instâncias, e qualquer definição da configuração que modifica o sistema operacional ou o software em instâncias é aplicada. As Configurações de verificação de integridade da implantação (Ignore health check (Ignorar verificação de integridade), Healthy threshold (Limite da integridade) Command timeout (Tempo limite do comando)) também se aplicam a atualizações contínuas baseadas na integridade e a atualizações imutáveis durante a fase de implantação.

Configurar atualizações contínuas

É possível habilitar e configurar atualizações contínuas no console do Elastic Beanstalk.

Para habilitar atualizações contínuas

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Rolling updates and deployments (Atualizações e implantações contínuas), escolha Edit (Editar).
5. Na seção Configuration updates (Atualizações da configuração), para Rolling update type (Tipo de atualização contínua), selecione uma das opções de Rolling (Contínua).



6. Selecione as configurações Batch size (Tamanho do lote), Minimum capacity (Capacidade mínima), e Pause time (Tempo de pausa).
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A seção Configuration updates (Atualizações da configuração) da página Rolling updates and deployments (Atualizações contínuas e implantações) tem as seguintes opções para atualizações contínuas:

- Rolling update type (Tipo de atualização contínua): o Elastic Beanstalk aguarda a conclusão da atualização de um lote de instâncias para passar para o próximo lote e permitir que essas instâncias concluam o bootstrapping e comecem a atender ao tráfego. Escolha uma das seguintes opções:
  - Rolling based on Health (Contínua com base na integridade): aguarde até que as instâncias no lote atual se tornem íntegras antes de colocar instâncias em serviço e iniciar o próximo lote.
  - Rolling based on Time (Contínua com base no tempo): especifique um tempo de espera entre o início de novas instâncias e sua colocação em serviço para iniciar o próximo lote.
  - Immutable (Imutável): aplique a alteração de configuração a um novo grupo de instâncias executando uma atualização imutável.
- Batch size (Tamanho do lote): o número de instâncias a serem substituídas em cada lote, entre **1** e **10000**. Por padrão, esse valor é um terço do tamanho mínimo do grupo de Auto Scaling arredondado para um número inteiro.
- Minimum capacity (Capacidade mínima): o número mínimo de instâncias para manter em execução enquanto outras instâncias estão sendo atualizadas, entre **0** e **9999**. O valor padrão é o tamanho mínimo do grupo de Auto Scaling ou um a menos do que o tamanho máximo do grupo de Auto Scaling, qualquer que seja o número menor.
- Pause time (Tempo de pausa) (somente baseado no tempo): a quantidade de tempo de espera após um lote ser atualizado antes de passar para o próximo lote para permitir que a aplicação comece a receber tráfego. Entre zero segundo e uma hora.

O namespace aws:autoscaling:updatepolicy:rollingupdate

Você também pode usar as opções de configuração no namespace `aws:autoscaling:updatepolicy:rollingupdate` para configurar as atualizações contínuas.

Use a opção `RollingUpdateEnabled` para habilitar atualizações contínuas e `RollingUpdateType` para escolher o tipo de atualização. Os valores a seguir são compatíveis com `RollingUpdateType`:

- `Health` – aguarde até que as instâncias no lote atual se tornem íntegras antes de colocá-las em serviço e iniciar o próximo lote.
- `Time` – especifique um tempo de espera entre a inicialização das novas instâncias e sua colocação em serviço para iniciar o próximo lote.
- `Immutable`: aplique a alteração de configuração a um novo grupo de instâncias executando uma atualização imutável.

Quando você habilitar as atualizações constantes, defina as opções `MaxBatchSize` e `MinInstancesInService` para configurar o tamanho de cada lote. Para atualizações contínuas baseadas no tempo, também é possível configurar `PauseTime` e `Timeout`, respectivamente.

Por exemplo, para iniciar até cinco instâncias por vez mantendo pelo menos duas instâncias em serviço e aguardar cinco minutos e 30 segundos entre cada lote, especifique as opções e os valores a seguir.

Example .ebextensions/timebased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Time
    PauseTime: PT5M30S
```

Para habilitar atualizações contínuas baseadas na integridade, com um tempo limite de 45 minutos para cada lote, especifique as opções e os valores a seguir.

Example .ebextensions/healthbased.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateEnabled: true
    MaxBatchSize: 5
    MinInstancesInService: 2
    RollingUpdateType: Health
```

```
    Timeout: PT45M
```

`Timeout`e `PauseTime` os valores devem ser especificados em ISO8601 duration:`PT#H#M#S`, onde cada `#` é o número de horas, minutos ou segundos, respectivamente.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Atualizações de ambiente imutáveis

Atualizações de ambiente imutável são uma alternativa a atualizações contínuas. As atualizações de ambiente imutável garantem que as alterações de configuração que exigem a substituição de instâncias sejam aplicadas de forma eficiente e segura. Se uma atualização de ambiente imutável falhar, o processo de reversão precisará apenas do encerramento de um grupo de Auto Scaling. Uma atualização contínua com falha, por outro lado, requer a execução de uma atualização contínua adicional para reverter as alterações.

Para executar uma atualização de ambiente imutável, o Elastic Beanstalk cria um segundo grupo temporário de Auto Scaling por trás do balanceador de carga do ambiente para conter as novas instâncias. Primeiro, o Elastic Beanstalk inicia uma única instância com a nova configuração no novo grupo. Essa instância serve o tráfego junto com todas as instâncias no grupo original de Auto Scaling que estão executando a configuração anterior.

Depois que a primeira instância passa pelas verificações de integridade, o Elastic Beanstalk executa instâncias adicionais com a nova configuração que correspondem ao número de instâncias em execução no grupo original de Auto Scaling. Quando todas as novas instâncias passam por verificações de integridade, o Elastic Beanstalk as transfere para o grupo original de Auto Scaling e encerra o grupo temporário de Auto Scaling e as instâncias antigas.

> **Note**
>
> Durante a atualização de ambiente imutável, a capacidade do ambiente dobra por um curto período quando as instâncias no novo grupo de Auto Scaling começam a atender às solicitações e antes de as instâncias do grupo original de Auto Scaling serem encerradas. Se o ambiente tiver muitas instâncias ou você tiver uma cota de instância sob demanda baixa, verifique se há capacidade suficiente para executar uma atualização de ambiente imutável. Se você estiver perto da cota, considere o uso de atualizações contínuas.

As atualizações imutáveis exigem um relatório de integridade aprimorada para avaliar a integridade de seu ambiente durante a atualização. Relatórios de integridade aprimorados combinam verificações de integridade padrão do load balancer com monitoramento de instância. Desse modo, ele verifica se as instâncias com a nova configuração estão atendendo às solicitações com êxito.

Você também pode usar as atualizações imutáveis para implantar novas versões do seu aplicativo, como uma alternativa para implantações contínuas. Quando você configura o Elastic Beanstalk para usar as atualizações imutáveis para implantações de aplicações, ele substitui todas as instâncias no ambiente cada vez que uma nova versão da aplicação é implantada. Se uma implantação de aplicação imutável falhar, o Elastic Beanstalk reverterá as alterações imediatamente ao encerrar o novo grupo de Auto Scaling. Isso pode impedir as implantações de frota parcial, que podem ocorrer quando uma implantação contínua falha após alguns lotes terem sido concluídos.

Warning

Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Isso faz com que todos os saldos EC2 estourados acumulados da Amazon sejam perdidos. Isso acontece nos seguintes casos:

- Atualizações de plataforma gerenciada com substituição de instância habilitada
- Atualizações imutáveis
- Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Se uma atualização imutável falhar, as novas instâncias farão upload de logs do pacote para o Amazon S3 antes que o Elastic Beanstalk as encerre. O Elastic Beanstalk deixa logs de uma atualização imutável com falha no Amazon S3 por uma hora antes de excluí-los, em vez de o padrão de 15 minutos para logs finais e em pacote.

Note

Se você usar as atualizações imutáveis para implantações de versão do aplicativo, mas não para a configuração, poderá encontrar um erro se tentar implantar uma versão do aplicativo com alterações de configuração que normalmente acionariam uma atualização contínua (por exemplo, configurações que alteram o tipo de instância). Para evitar isso, faça a alteração de configuração em uma atualização separada ou configure as atualizações imutáveis tanto para implantações quanto para alterações de configuração.

Não é possível realizar uma atualização imutável em conjunto com as alterações de configuração de recurso. Por exemplo, não é possível alterar as configurações que exigem substituição de instância e, ao mesmo tempo, atualizar outras configurações ou realizar uma implantação imutável com arquivos de configuração que alteram as definições de configuração ou recursos adicionais em seu código-fonte. Se você tentar alterar as configurações de recurso (por exemplo, as configurações do balanceador de carga) e simultaneamente executar uma atualização imutável, o Elastic Beanstalk retornará um erro.

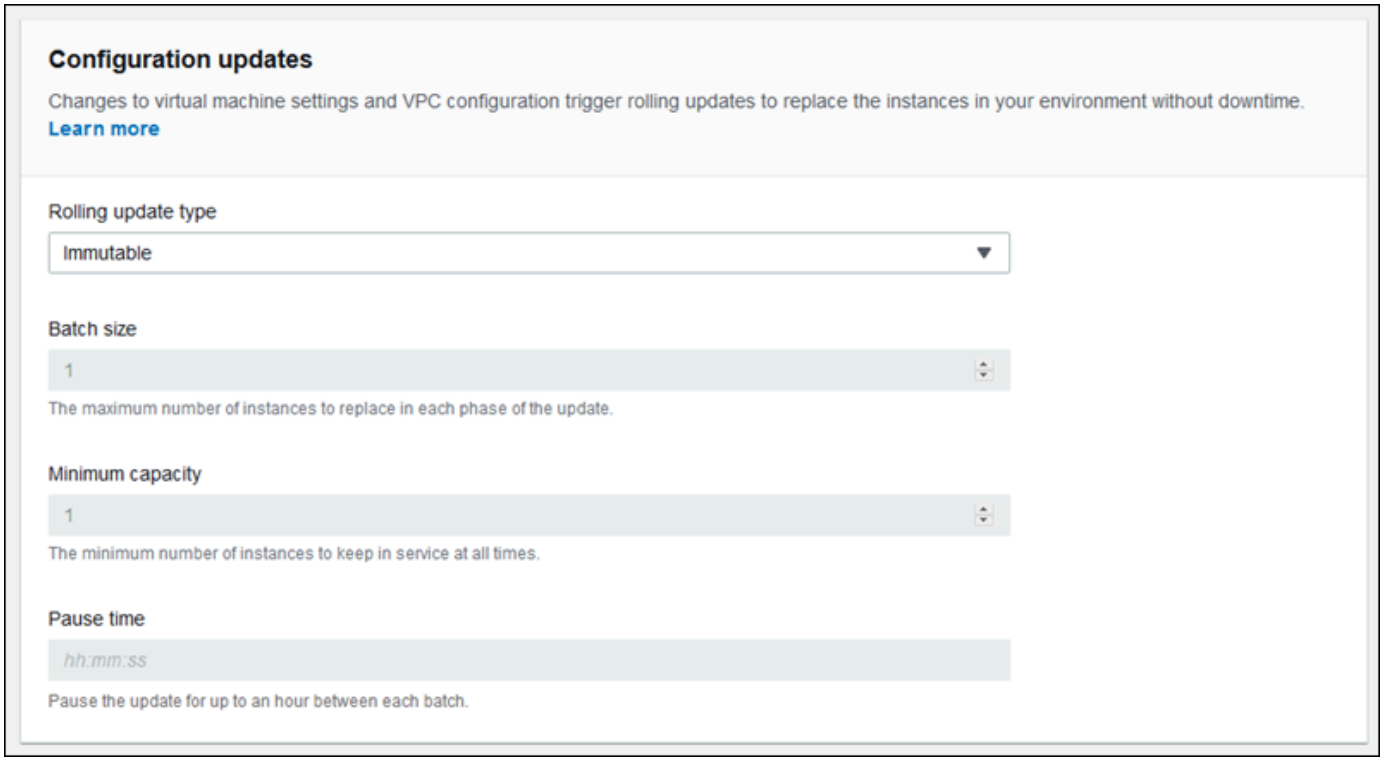
Se as alterações de configuração de recurso não são dependente de alteração em seu código-fonte ou na configuração de instância, execute-as em duas atualizações. Se elas são dependentes, realize uma implantação azul/verde.

Configurar as atualizações imutáveis

É possível habilitar e configurar atualizações imutáveis no console do Elastic Beanstalk.

Para habilitar as atualizações imutáveis (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Rolling updates and deployments (Atualizações e implantações contínuas), escolha Edit (Editar).
5. Na seção Configuration Updates, defina o Rolling update type para Immutable.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

O namespace aws:autoscaling:updatepolicy:rollingupdate

Você também pode usar as opções de configuração no namespace `aws:autoscaling:updatepolicy:rollingupdate` para configurar as atualizações imutáveis. O exemplo de arquivo de configuração a seguir permite atualizações imutáveis para alterações de configuração.

Example .ebextensions/immutable-updates.config

```
option_settings:
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
```

O exemplo a seguir permite atualizações imutáveis tanto para alterações de configuração quanto para implantações.

Example .ebextensions/immutable-all.config

```
option_settings:
```

```
  aws:autoscaling:updatepolicy:rollingupdate:
    RollingUpdateType: Immutable
  aws:elasticbeanstalk:command:
    DeploymentPolicy: Immutable
```

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Atualizar a versão de plataforma do ambiente Elastic Beanstalk

O Elastic Beanstalk libera regularmente novas versões da plataforma para atualizar todas as plataformas baseadas no Linux e no Windows Server. As novas versões de plataforma fornecem atualizações para componentes de software existentes e suporte para novos recursos e opções de configuração. Para saber mais sobre plataformas e versões de plataformas, consulte Glossário de plataformas do Elastic Beanstalk.

É possível usar o console do Elastic Beanstalk ou a CLI do EB para atualizar a versão da plataforma do ambiente. Dependendo da versão da plataforma que você deseja atualizar, o Elastic Beanstalk recomenda um dos dois métodos para realizar atualizações de plataforma.

- Método 1: Atualizar a versão da plataforma do ambiente. Recomendamos esse método quando você está atualizando para a versão mais recente da plataforma em uma ramificação, com o mesmo runtime, servidor Web, servidor de aplicações e sistema operacional e sem alterar a versão principal da plataforma. Essa é a atualização de plataforma mais comum e rotineira.
- Método 2 — Executar uma Blue/Green implantação. Recomendamos esse método quando você está atualizando para uma versão de plataforma em uma ramificação diferente, com um runtime, servidor Web, servidor de aplicações ou sistema operacional diferente, ou para outra versão principal da plataforma. Essa é uma boa abordagem quando você deseja aproveitar os novos recursos de runtime ou a funcionalidade do Elastic Beanstalk mais recente ou quando deseja sair de uma ramificação de plataforma defasada ou retirada.

  A migração de uma versão antiga da plataforma exige uma blue/green implantação, pois essas versões da plataforma são incompatíveis com as versões atualmente suportadas.

  A migração de um aplicativo Linux para o Amazon Linux 2 requer uma blue/green implantação, porque as versões da plataforma Amazon Linux 2 são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI.

Para obter mais ajuda sobre como escolher o melhor método de atualização de plataforma, expanda a seção da plataforma do ambiente.

Docker

Use o Método 1 para executar atualizações da plataforma.

Docker de vários contêineres

Use o Método 1 para executar atualizações da plataforma.

Docker pré-configurado

Considere os seguintes casos:

- Se você estiver migrando a aplicação para outra plataforma, por exemplo, do Go 1.4 (Docker) para o Go 1.11 ou do Python 3.4 (Docker) para o Python 3.6, use o Método 2.
- Se você estiver migrando a aplicação para uma versão de contêiner diferente do Docker, por exemplo, da versão Glassfish 4.1 (Docker) para a Glassfish 5.0 (Docker), use o Método 2.
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão do contêiner ou da versão principal, use o Método 1.

Go

Use o Método 1 para executar atualizações da plataforma.

Java SE

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime Java diferente, por exemplo, de Java 7 para Java 8, use o Método 2.
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime, use o Método 1.

Java com Tomcat

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime Java diferente ou uma versão do servidor de aplicações Tomcat, por exemplo, do Java 7 com Tomcat 7 para o Java 8 com Tomcat 8.5, use o Método 2.
- Se você estiver migrando a aplicação para as principais versões da plataforma Java com o Tomcat (v1.x.x, v2.x.x e v3.x.x), use o Método 2.
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime, na versão do servidor de aplicações ou na versão principal, use o Método 1.

.NET no Windows Server com IIS

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão diferente do sistema operacional Windows, por exemplo, da versão Windows Server 2008 R2 para Windows Server 2016, use o Método 2.
- Se você estiver migrando a aplicação para as principais versões da plataforma Windows Server, consulte Migrar de versões principais anteriores da plataforma do Windows Server e use o Método 2.
- Se a aplicação estiver sendo executado em uma plataforma Windows Server V2.x.x e você estiver atualizando para uma versão de plataforma mais recente, use o Método 1.

> **Note**
>
> As versões da plataforma Windows Server anteriores à v2 não são agrupadas semanticamente em versões. Só é possível iniciar a versão mais recente de cada uma das versões principais da plataforma do Windows Server e não é possível reverter após a atualização.

Node.js

Use o Método 2 para executar atualizações da plataforma.

PHP

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime PHP diferente, por exemplo, da versão PHP 5.6 para PHP 7.2, use o Método 2.
- Se você estiver migrando a aplicação para as principais versões da plataforma PHP (v1.x.x e v2.x.x), use o Método 2.
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime ou da versão principal, use o Método 1.

Python

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime diferente do Python, por exemplo, da versão Python 2.7 para Python 3.6, use o Método 2.
- Se você estiver migrando a aplicação para as principais versões da plataforma Python (v1.x.x e v2.x.x), use o Método 2.
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime ou da versão principal, use o Método 1.

Ruby

Considere os seguintes casos:

- Se você estiver migrando a aplicação para uma versão de runtime ou uma versão do servidor de aplicações Ruby diferente, por exemplo, da versão Ruby 2.3 com Puma para Ruby 2.6 com Puma, use o Método 2.
- Se você estiver migrando a aplicação para as principais versões da plataforma Ruby (v1.x.x e v2.x.x), use o Método 2.
- Se você estiver atualizando para uma versão de plataforma mais recente sem nenhuma alteração na versão de runtime, na versão do servidor de aplicações ou na versão principal, use o Método 1.

Método 1: Atualizar a versão da plataforma do ambiente

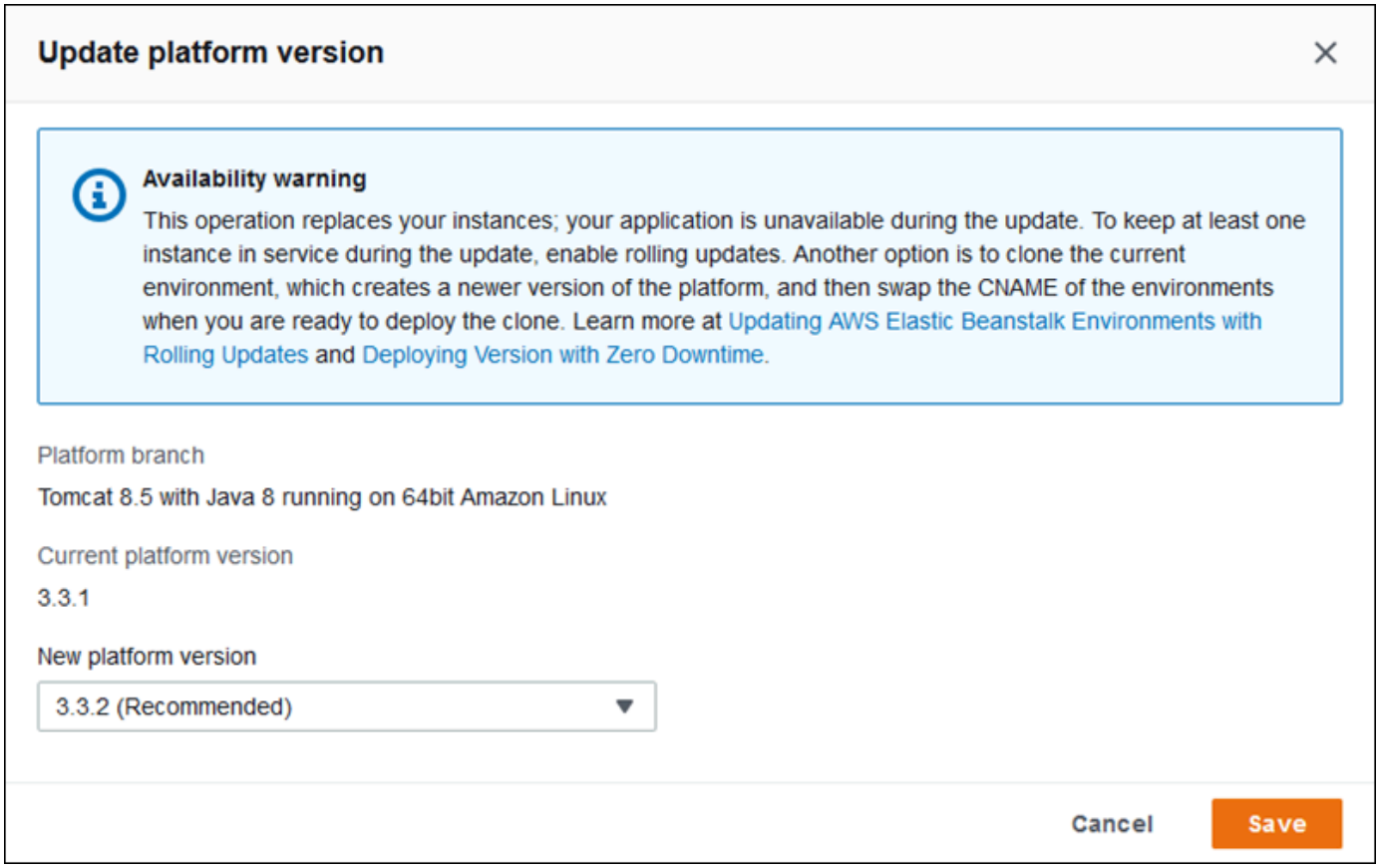
Use esse método para atualizar para a ramificação mais recente da plataforma do seu ambiente. Se você criou um ambiente anteriormente usando uma versão mais antiga da plataforma, ou atualizou o ambiente de uma versão mais antiga, também é possível esse método para reverter para uma versão anterior da plataforma, contanto que seja na mesma ramificação da plataforma.

Como atualizar a versão da plataforma de seu ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, em Platform (Plataforma), escolha Change (Alterar).



4. Na caixa de diálogo Update platform version (Atualizar versão da plataforma), selecione uma versão da plataforma. A versão mais recente (recomendada) da plataforma na ramificação é selecionada automaticamente. É possível atualizar para qualquer versão que você usou no passado.



5. Escolha Save (Salvar).

Para simplificar ainda mais as atualizações da plataforma, o Elastic Beanstalk pode gerenciá-las para você. É possível configurar o ambiente para aplicar atualizações de versão secundárias e de patch automaticamente durante uma janela de manutenção semanal configurável. O Elastic Beanstalk aplica atualizações gerenciadas sem tempo de inatividade nem redução na capacidade, e cancelará a atualização imediatamente se as instâncias que executam a aplicação na nova versão falharem nas verificações de integridade. Para obter detalhes, consulte Atualizações gerenciadas de plataforma.

Método 2 — Executar uma Blue/Green implantação

Use este método para atualizar para uma ramificação de plataforma diferente, com um runtime, servidor Web, servidor de aplicações ou sistema operacional diferente, ou para outra versão principal da plataforma. Normalmente, isso é necessário quando você deseja aproveitar os novos recursos de runtime ou a funcionalidade mais recente do Elastic Beanstalk. Também é necessário quando você está migrando para fora de uma ramificação defasada ou desativada da plataforma.

Quando você migra entre as principais versões de plataforma ou para as versões de plataforma com as principais atualizações de componentes, há uma probabilidade maior de que seu aplicativo, ou alguns aspectos dele, não funcione conforme o esperado na nova versão da plataforma e possa exigir alterações.

Antes de executar a migração, atualize sua máquina de desenvolvimento local para as versões de runtime mais novas e outros componentes da plataforma para a qual você planeja migrar. Verifique se o aplicativo ainda funciona conforme o esperado e faça as correções e alterações de código necessárias. Depois, use o procedimento de práticas recomendadas a seguir para migrar com segurança o ambiente para a nova versão da plataforma.

Para migrar o ambiente para uma versão de plataforma com atualizações importantes

1. Crie um ambiente novo usando a nova versão de plataforma de destino e implante o código do aplicativo nela. O novo ambiente deve estar na aplicação do Elastic Beanstalk que contém o ambiente para o qual você está migrando. Não encerre o ambiente existente.
2. Use o novo ambiente para migrar o aplicativo. Em particular:

   - Encontre e corrija quaisquer problemas de compatibilidade de aplicativos que você não conseguiu descobrir durante a fase de desenvolvimento.
   - Certifique-se de que as personalizações que seu aplicativo faz usando os arquivos de configuração funcionem corretamente no novo ambiente. Eles podem incluir configurações de opções, pacotes adicionais instalados, políticas de segurança personalizadas e arquivos de script ou configuração instalados em instâncias de ambiente.
   - Se o aplicativo usar uma imagem de máquina da Amazon (AMI), crie uma nova AMI personalizada com base na AMI da nova versão da plataforma. Para saber mais, consulte Usar uma imagem de máquina Amazon personalizada (AMI) no ambiente Elastic Beanstalk. Especificamente, isso é necessário se o aplicativo usar a plataforma Windows Server com uma AMI personalizada e você estiver migrando para uma versão da plataforma Windows Server V2. Neste caso, consulte também Migrar de versões principais anteriores da plataforma do Windows Server.

   Itere os testes e implante suas correções até que você esteja satisfeito com o aplicativo no novo ambiente.
3. Torne o novo ambiente em seu ambiente de produção. Para isso, troque o CNAME dele pelo CNAME do ambiente de produção existente. Para obter mais detalhes, consulte Implantações azuis/verdes com o Elastic Beanstalk.

4. Quando você estiver satisfeito com o estado do seu novo ambiente de produção, encerre o ambiente antigo. Para obter mais detalhes, consulte Encerrar um ambiente do Elastic Beanstalk.

Atualizações gerenciadas de plataforma

AWS Elastic Beanstalk lança regularmente atualizações de plataforma para fornecer correções, atualizações de software e novos recursos. Com as atualizações gerenciadas de Plataforma, você pode configurar seu ambiente para atualizar automaticamente para a versão mais recente de uma Plataforma durante uma janela de manutenção programada. Seu aplicativo permanece em serviço durante o processo de atualização, sem redução da capacidade. As atualizações gerenciadas estão disponíveis em ambos os ambientes de instância única e com balanceamento de carga.

> **Note**
>
> Esse recurso não está disponível em versões da plataforma Windows Server anteriores à versão 2 (v2).

Você pode configurar seu ambiente para aplicar automaticamente as atualizações de versão de patch ou ambas as atualizações de patch e de versão secundária. As atualizações gerenciadas da plataforma não são compatíveis com atualizações em ramificações da plataforma (atualizações para diferentes versões principais de componentes da plataforma, como sistema operacional, runtime ou componentes do Elastic Beanstalk), pois elas podem apresentar alterações incompatíveis com versões anteriores.

Você também pode configurar o Elastic Beanstalk para substituir todas as instâncias no ambiente durante a janela de manutenção, mesmo se não houver uma atualização de plataforma disponível. A substituição de todas as instâncias em seu ambiente será útil se o seu aplicativo encontrar bugs ou problemas de memória quando executado por um longo período.

Em ambientes criados em 25 de novembro de 2019 ou posteriormente usando o console do Elastic Beanstalk, as atualizações gerenciadas são habilitadas por padrão sempre que possível. As atualizações gerenciadas exigem integridade aprimorada para serem habilitadas. A integridade aprimorada é habilitada por padrão quando você seleciona uma das predefinições de configuração, e desabilitada quando você seleciona Custom configuration (Configuração personalizada). O console não pode habilitar atualizações gerenciadas para versões de plataforma mais antigas que não oferecem suporte à integridade aprimorada ou quando a integridade aprimorada está desabilitada.

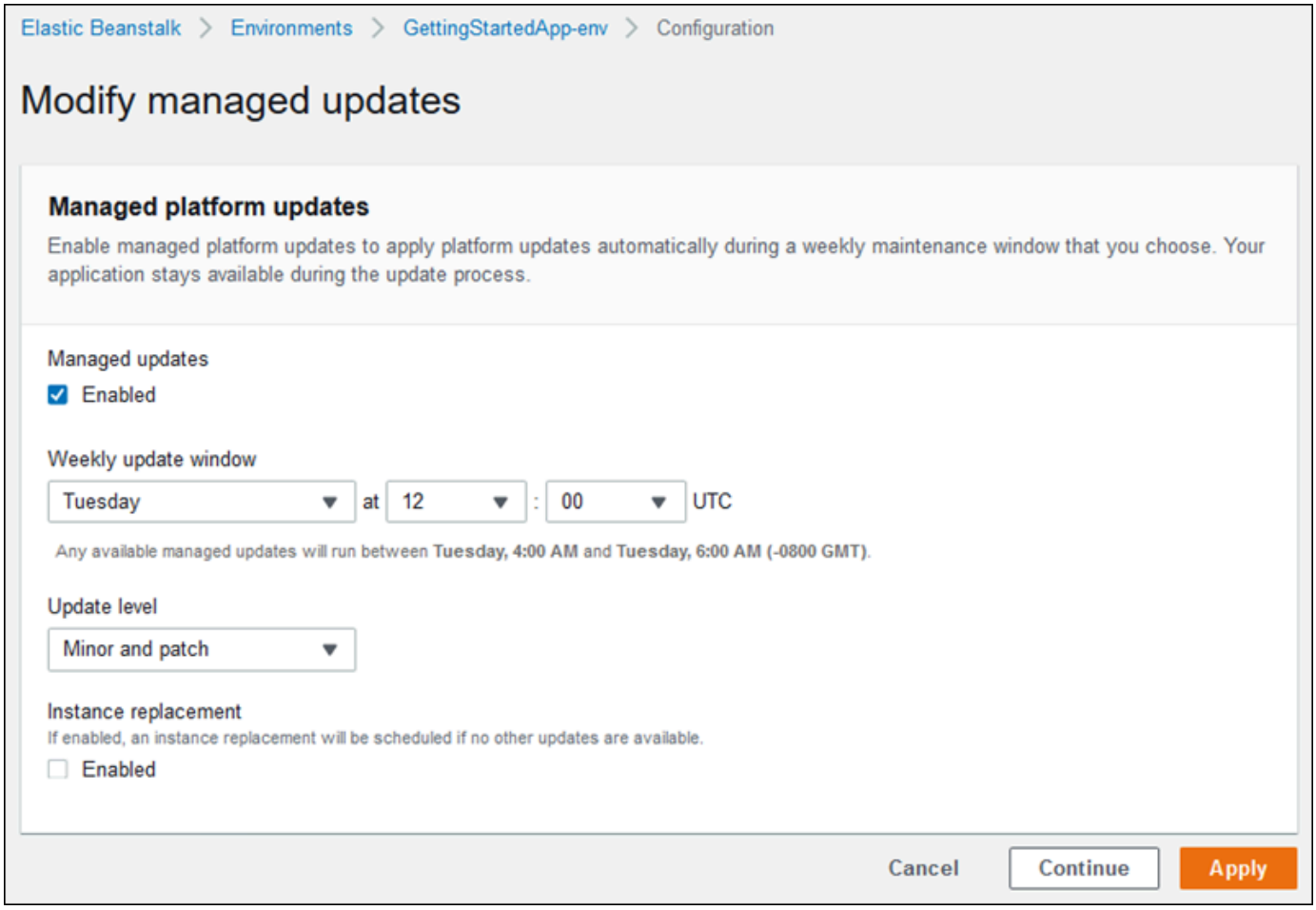
Quando o console habilita as atualizações gerenciadas para um novo ambiente, a Weekly update window (Janela de atualização semanal) é definida como um dia aleatório da semana em um horário aleatório. O Update level (Nível de atualização) é definido como Minor and patch (Menor e patch), e a Instance replacement (Substituição da instância) é desabilitada. Você pode desabilitar ou reconfigurar atualizações gerenciadas antes da etapa final de criação do ambiente.

Para um ambiente existente, use o console do Elastic Beanstalk a qualquer momento para configurar atualizações de plataforma gerenciadas.

> **Important**
>
> Um grande número de ambientes Beanstalk em uma conta AWS pode apresentar um risco de problemas de controle de utilização durante atualizações gerenciadas. Grande número é uma quantidade relativa que depende de quão rigorosamente você agenda as atualizações gerenciadas para seus ambientes. Mais de 200 ambientes em uma conta agendados de perto podem causar problemas de controle de utilização, embora um número menor também possa ser problemático.
> Para equilibrar a carga de recursos para atualizações gerenciadas, recomendamos distribuir as janelas de manutenção programadas para os ambientes em uma conta.
> Além disso, considere uma estratégia de várias contas. Para obter mais informações, consulte Organizando seu AWS ambiente usando várias contas no site AWS Whitepapers & Guides.

Como configurar atualizações de plataforma gerenciadas

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria Managed updates (Atualizações gerenciadas), escolha Edit (Editar).
5. Desabilite ou habilite as Managed Updates (Atualizações gerenciadas).
6. Se as atualizações gerenciadas estiverem habilitadas, selecione uma janela de manutenção e selecione um Update level (Nível de atualização).
7. (Opcional) Selecione Instance replacement (Substituição da instância) para habilitar a substituição semanal da instância.



8. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

As atualizações gerenciadas de Plataforma dependem de relatórios de integridade aprimorada para determinar se o seu aplicativo é íntegro o suficiente para considerar a atualização da Plataforma bem-sucedida. Para obter instruções, consulte Habilitar relatórios de integridade aprimorada do Elastic Beanstalk.

Seções

- Permissões necessárias para executar atualizações gerenciadas de plataforma
- Janela de manutenção da atualização gerenciada
- Atualizações de versão secundária e patch
- Atualizações de ambiente imutáveis
- Gerenciar atualizações gerenciadas
- Namespaces da opção de ação gerenciada

Permissões necessárias para executar atualizações gerenciadas de plataforma

O Elastic Beanstalk precisa de permissão para iniciar uma atualização de plataforma em seu nome. Para obter essas permissões, o Elastic Beanstalk assume a função de serviço de atualizações gerenciadas. Quando a função de serviço padrão é usada para o ambiente, o console do Elastic Beanstalk também a usa como função de serviço de atualizações gerenciadas. O console atribui a política gerenciada `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` à sua função de serviço. Esta política tem todas as permissões que o Elastic Beanstalk precisa para executar atualizações de plataforma gerenciada.

Para obter detalhes sobre outras formas de definir a função de serviço de atualizações gerenciadas, consulte the section called “Perfis de serviço”.

> **Note**
>
> Se você usar arquivos de configuração para estender o ambiente a fim de incluir recursos adicionais, poderá ser necessário adicionar permissões à função de serviço de atualizações gerenciadas do ambiente. Normalmente, você precisa adicionar permissões ao fazer referência a esses recursos pelo nome em outras seções ou arquivos.

Se houver falha em uma atualização, você poderá localizar o motivo na página Atualizações gerenciadas.

Janela de manutenção da atualização gerenciada

Ao AWS lançar uma nova versão da plataforma do seu ambiente, o Elastic Beanstalk agenda uma atualização gerenciada da plataforma durante a próxima janela de manutenção semanal. As janelas de manutenção são de duas horas. O Elastic Beanstalk inicia uma atualização agendada durante a janela de manutenção. Pode ser que a atualização não seja concluída antes do término das janelas.

> **Note**
>
> Na maioria dos casos, o Elastic Beanstalk programa a atualização gerenciada para que ela ocorra durante a próxima janela de manutenção semanal. O sistema considera diversos aspectos de segurança e disponibilidade do serviço de atualização ao programar as atualizações gerenciadas. Em casos raros, uma atualização pode não ser programada para a primeira janela de manutenção. Quando isso acontece, o sistema tenta novamente na

próxima janela de manutenção. Para aplicar manualmente a atualização gerenciada, escolha Apply now (Aplicar agora) como explicado em Gerenciar atualizações gerenciadas, nesta página.

Atualizações de versão secundária e patch

Você pode habilitar as atualizações gerenciadas de plataforma para aplicar somente atualizações de versão de patch ou para ambas as atualizações de patch e de versão secundária. As atualizações de versão de patch fornecem correções de bugs e melhorias de performance e podem incluir alterações de configuração secundárias no software da instância, nos scripts e nas opções de configuração. Atualizações de versões secundárias são compatíveis com novos recursos do Elastic Beanstalk. Você não pode aplicar atualizações de versão principal, que podem fazer alterações incompatíveis com a versão anterior, com as atualizações gerenciadas da plataforma.

No número da versão de um plataforma, o segundo número é a versão de atualização secundária, e o terceiro número é a versão do patch. Por exemplo, a versão 2.0.7 de uma plataforma tem 0 como versão secundária e 7 como versão de patch.

Atualizações de ambiente imutáveis

As atualizações gerenciadas de Plataforma executam atualizações de ambiente imutáveis para atualizar seu ambiente para uma nova versão de Plataforma. As atualizações imutáveis atualizam o ambiente sem retirar as instâncias de serviço ou modificar o ambiente, antes de confirmar que as instâncias que executam a nova versão foram aprovadas nas verificações de integridade.

Em uma atualização imutável, o Elastic Beanstalk implanta o mesmo número de instâncias que estão em execução no momento com a nova versão da plataforma. As novas instâncias começam a fazer solicitações juntamente com aquelas que executam a versão antiga. Se o novo conjunto de instâncias for aprovado em todas as verificações de integridade, o Elastic Beanstalk encerrará o conjunto antigo de instâncias, deixando apenas aquelas com a nova versão.

As atualizações gerenciadas de plataforma sempre realizam atualizações imutáveis, mesmo quando você as aplica fora da janela de manutenção. Se você alterar a versão da plataforma no Dashboard (Painel), o Elastic Beanstalk aplicará a política de atualização escolhida para as atualizações de configuração.

> **Warning**
>
> Algumas políticas substituem todas as instâncias durante a implantação ou a atualização. Isso faz com que todos os saldos EC2 estourados acumulados da Amazon sejam perdidos. Isso acontece nos seguintes casos:
>
> - Atualizações de plataforma gerenciada com substituição de instância habilitada
> - Atualizações imutáveis
> - Implantações com atualizações imutáveis ou divisão de tráfego habilitada

Gerenciar atualizações gerenciadas

O console do Elastic Beanstalk exibe informações detalhadas sobre atualizações gerenciadas na página Managed updates overview (Visão geral de atualizações gerenciadas).

Para visualizar informações sobre atualizações gerenciadas (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Managed updates (Atualizações gerenciadas).

A seção Visão geral de atualizações gerenciadas apresenta informações sobre as atualizações gerenciadas programadas e pendentes. A seção History lista as atualizações bem-sucedidas e as tentativas com falha.

Você pode aplicar uma atualização programada imediatamente, em vez de esperar pela janela de manutenção.

Para aplicar uma atualização gerenciada de plataforma imediatamente (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Managed updates (Atualizações gerenciadas).
4. Escolha Apply now.
5. Verifique os detalhes da atualização e escolha Apply (Aplicar).

Quando você aplica uma atualização gerenciada de plataforma fora da janela de manutenção, o Elastic Beanstalk realiza uma atualização imutável. Se você atualizar a plataforma do ambiente no Dashboard (Painel) ou usando um cliente diferente, o Elastic Beanstalk aplicará o tipo de atualização selecionado nas alterações de configuração.

Se você não tem uma atualização gerenciada programada, talvez o ambiente já esteja executando a versão mais recente. Outros motivos para não ter uma atualização programada incluem:

- Uma atualização de versão secundária está disponível, mas o ambiente está configurado para aplicar automaticamente apenas as atualizações de versão de patch.
- O ambiente não foi verificado desde que a atualização foi lançada. O Elastic Beanstalk normalmente verifica se há atualizações a cada hora.
- Uma atualização está pendente ou já está em andamento.

Quando a janela de manutenção é iniciada ou quando você seleciona Apply now (Aplicar agora), as atualizações programadas entram no status pendente antes da execução.

Namespaces da opção de ação gerenciada

Você pode usar as opções de configuração nos namespaces `aws:elasticbeanstalk:managedactions` e `aws:elasticbeanstalk:managedactions:platformupdate` para habilitar e configurar as atualizações gerenciadas de Plataforma.

A opção `ManagedActionsEnabled` ativa as atualizações gerenciadas de plataforma. Defina essa opção como `true` para habilitar as atualizações gerenciadas de plataforma e use as outras opções para configurar o comportamento de atualização.

Use `PreferredStartTime` para configurar o início da janela de manutenção semanal no *`minute`* formato *`dayhour`*::.

Defina `UpdateLevel` como `minor` ou `patch` para aplicar ambas as atualizações de versão secundária e de patch ou apenas as atualizações de versão de patch, respectivamente.

Quando as atualizações gerenciadas de plataforma são habilitadas, você pode permitir a substituição de instâncias definindo a opção `InstanceRefreshEnabled` como `true`. Quando essa configuração é habilitada, o Elastic Beanstalk executa uma atualização imutável no ambiente a cada semana, mesmo que não haja uma nova versão da plataforma disponível.

O arquivo de configuração de exemplo a seguir habilita as atualizações gerenciadas de plataforma para atualizações da versão de patch com uma janela de manutenção que começa às 9h UTC toda terça-feira.

Example .ebextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Migrar seu aplicativo de uma versão de plataforma legada

Se você tiver implantado uma aplicação do Elastic Beanstalk que usa uma versão de plataforma legada, deverá migrar a aplicação para um novo ambiente usando uma versão de plataforma não legada para ter acesso a novos recursos. Se não tiver certeza de estar executando o aplicativo usando uma versão de plataforma legada, verifique no console do Elastic Beanstalk. Para obter instruções, consulte Como verificar se você está usando uma versão de plataforma legada.

Quais novos recursos as versões de plataforma legada não disponibilizam?

Plataformas herdadas não são compatíveis com os seguintes recursos:

- Arquivos de configuração, como descrito no tópico Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`)
- Verificações de integridade do ELB, conforme descrito no tópico Relatórios de integridade básica
- Perfis de instância, conforme descrito no tópico Gerenciar perfis de instância do Elastic Beanstalk
- VPCs, conforme descrito no Usar o Elastic Beanstalk com Amazon VPC tópico
- Níveis de dados, como descrito no tópico Adição de um banco de dados ao seu ambiente do Elastic Beanstalk
- Níveis de operador, como descrito no tópico Ambientes de operador do Elastic Beanstalk
- Ambientes de instância única, como descrito no tópico Tipos de ambiente
- Tags, como descrito no tópico Marcar recursos em seus ambientes do Elastic Beanstalk
- Lançamento de atualizações, como descrito no tópico Atualizações de configuração de ambiente contínuas do Elastic Beanstalk

Por que algumas versões de plataforma são marcadas como legadas?

Algumas versões mais antigas da plataforma não são compatíveis com os recursos mais recentes do Elastic Beanstalk. Essas versões são marcadas como (legacy) (herdado) na página de visão geral do ambiente no console do Elastic Beanstalk.

Como verificar se você está usando uma versão de plataforma legada

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, visualize o nome da Platform (Plataforma).

   O aplicativo está usando uma versão de plataforma legada se você vir (legacy) (legada) ao lado do nome da plataforma.

Para migrar seu aplicativo

1. Implante o aplicativo em um novo ambiente. Para obter instruções, vá para Criar um ambiente do Elastic Beanstalk.
2. Se você tiver uma instância de banco de dados Amazon RDS, atualize seu grupo de segurança de banco de dados para permitir acesso ao seu grupo EC2 de segurança para seu novo ambiente. Para obter instruções sobre como encontrar o nome do seu grupo de EC2 segurança usando o AWS Management Console, consulteEC2 grupos de segurança. Para obter mais informações sobre como configurar seu grupo de EC2 segurança, acesse a seção “Autorizando o acesso à rede a um grupo de EC2 segurança da Amazon” em Trabalhando com grupos de segurança de banco de dados no Guia do usuário do Amazon Relational Database Service.
3. Troque o URL de seu ambiente. Para obter instruções, vá para Implantações azuis/verdes com o Elastic Beanstalk.
4. Encerre seu ambiente antigo. Para obter instruções, vá para Encerrar um ambiente do Elastic Beanstalk.

> **Note**
>
> Se você usar AWS Identity and Access Management (IAM), precisará atualizar suas políticas para incluir AWS CloudFormation o Amazon RDS (se aplicável). Para obter

mais informações, consulte Usando o Elastic Beanstalk com AWS Identity and Access Management.

Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2

Esta seção descreve como migrar sua aplicação usando um dos seguintes caminhos de migração.

- Migrar de uma ramificação da plataforma Amazon Linux 2 para uma ramificação da plataforma Amazon Linux 2023.
- Migre de uma ramificação da plataforma Amazon Linux AMI (AL1) para uma ramificação da plataforma Amazon Linux 2023 (recomendado) ou Amazon Linux 2.

Tópicos

- Migração do Amazon Linux 2 para o Amazon Linux 2023
- Migração do Amazon Linux AMI (AL1) para AL2 ou AL2 023

Migração do Amazon Linux 2 para o Amazon Linux 2023

Este tópico fornece orientações para ajudar a migrar sua aplicação de uma ramificação da plataforma Amazon Linux 2 para uma ramificação da plataforma Amazon Linux 2023.

Diferenças e compatibilidade

Entre o Elastic Beanstalk e as AL2 plataformas AL2 023

Há um alto grau de compatibilidade entre as plataformas Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Embora haja algumas diferenças a serem observadas:

- Instance Metadata Service versão 1 (IMDSv1) — A configuração da IMDSv1 opção Desativar é padronizada `true` em AL2 plataformas 023. O padrão está `false` nas AL2 plataformas.
- ferramenta de instância pkg-repo — A pkg-repo ferramenta não está disponível para ambientes executados em AL2 plataformas 023. No entanto, você pode aplicar manualmente as atualizações do pacote e do sistema operacional a uma instância AL2 023. Para obter mais informações,

consulte Gerenciamento de pacotes e atualizações do sistema operacional no Guia do usuário do Amazon Linux 2023

- HTTPd Configuração do Apache — O `httpd.conf` arquivo Apache para plataformas AL2 023 tem algumas definições de configuração que são diferentes daquelas para: AL2
  - Negar acesso a todo o sistema de arquivos do servidor por padrão. Essas configurações são descritas em Proteger arquivos do servidor por padrão na página Dicas de segurança do site Apache.
  - Impeça que os usuários substituam os recursos de segurança que você configurou. A configuração nega o acesso à configuração `.htaccess` em todos os diretórios, exceto aqueles especificamente habilitados. Essa configuração é descrita em Protegendo as configurações do sistema na página Dicas de segurança do site Apache. A página Tutorial do Apache HTTP Server: arquivos.htaccess afirma que essa configuração pode ajudar a melhorar o desempenho.
  - Negue acesso a arquivos com padrão de nome `.ht*`. Essa configuração impede que os clientes da Web exibam arquivos `.htaccess` e `.htpasswd`.

Você pode alterar qualquer uma das definições de configuração acima para seu ambiente. Para obter mais informações, consulte Configurar o Apache HTTPD.

Entre sistemas operacionais Amazon Linux

Para saber mais sobre as diferenças entre o Amazon Linux 2 e o Amazon Linux 2023, consulte Comparação entre o Amazon Linux 2 e o Amazon Linux 2023 no Guia do usuário do Amazon Linux 2023.

Para saber mais sobre o Amazon Linux 2023, consulte O que é o Amazon Linux 2023? no Guia do usuário do Amazon Linux 2023.

Processo geral de migração

Quando você estiver pronto para começar a produção, o Elastic Beanstalk precisará de uma implantação para realizar a blue/green atualização. A seguir estão as etapas gerais de melhores práticas que recomendamos para a migração com um procedimento blue/green de implantação.

Preparar para testar a migração

Antes de implantar a aplicação e começar a testá-la, revise as informações na seção Diferenças e compatibilidade anterior. Veja também a referência citada nessa seção, Comparação entre o Amazon Linux 2 e o Amazon Linux 2023 no Guia do usuário do Amazon Linux 2023. Anote as

informações específicas desse conteúdo que podem ser usadas na aplicação e na configuração definida.

Etapas de migração de alto nível

1. Crie um novo ambiente baseado em uma ramificação da plataforma AL2 023.
2. Implante seu aplicativo no ambiente AL2 023 de destino.

   O ambiente de produção existente permanecerá ativo e não será afetado, enquanto você estiver fazendo interações por meio de testes e ajustes no novo ambiente.
3. Teste a aplicação minuciosamente no novo ambiente.
4. Quando seu ambiente AL2 023 de destino estiver pronto para ser produzido, troque um CNAMEs dos dois ambientes para redirecionar o tráfego para o novo AL2 ambiente 023.

Etapas de migração mais detalhadas e práticas recomendadas

Para obter um procedimento de blue/green implantação mais detalhado, consulteImplantações azuis/verdes com o Elastic Beanstalk.

Para obter orientações mais específicas e etapas detalhadas de práticas recomendadas, consulte Método azul/verde.

Mais referências para ajudar a planejar a migração

As referências a seguir podem oferecer informações adicionais para planejar sua migração.

- Plataformas compatíveis com Elastic Beanstalk em Plataformas AWS Elastic Beanstalk
- Histórico de ramificações de plataforma retiradas
- the section called “Plataformas Linux”
- Perguntas frequentes sobre descontinuação de plataformas

Migração do Amazon Linux AMI (AL1) para AL2 ou AL2 023

Se a aplicação do Elastic Beanstalk for baseada em uma ramificação de plataforma do Amazon Linux AMI, use esta seção para saber como migrar os ambientes da aplicação para o Amazon Linux 2 ou Amazon Linux 2023. As ramificações da plataforma da geração anterior baseadas no Amazon Linux AMI foram descontinuadas.

É altamente recomendável migrar para o Amazon Linux 2023, já que ele é mais recente do que o Amazon Linux 2. O encerramento do suporte ao sistema operacional Amazon Linux 2 ocorrerá antes do encerramento do suporte ao Amazon Linux 2023. Consequentemente, você se beneficiará de um período de suporte mais longo se migrar para o Amazon Linux 2023.

Há um alto grau de compatibilidade entre as plataformas Amazon Linux 2 e Amazon Linux 2023 do Elastic Beanstalk. Embora algumas áreas tenham diferenças: a opção padrão do Instance Metadata Service Version 1 (IMDSv1), o suporte para a ferramenta de instância pkg-repo e algumas configurações do Apache. HTTPd Para obter mais informações, consulte Amazon Linux 2023.

Diferenças e compatibilidade

Não é garantido que as ramificações da plataforma AL2 baseadas em AL2 023/ sejam compatíveis com versões anteriores de seu aplicativo existente. É importante ter em mente que, mesmo que o código da aplicação seja implantado com sucesso na nova versão da plataforma, ele poderá se comportar ou ser executado de forma diferente devido às diferenças de runtime e sistema operacional.

Embora o Amazon Linux AMI e o AL2 023/ AL2 compartilhem o mesmo kernel Linux, eles diferem nos seguintes aspectos: sistema de inicialização, `libc` versões, cadeia de ferramentas do compilador e vários pacotes. Para obter mais informações, consulte Amazon Linux 2 FAQs.

O serviço Elastic Beanstalk também atualizou as versões de runtime, ferramentas de compilação e outras dependências específicas de cada plataforma.

Portanto, recomendamos que você reserve um tempo para testar seu aplicativo minuciosamente em um ambiente de desenvolvimento e faça os ajustes necessários.

Processo geral de migração

Quando você estiver pronto para começar a produção, o Elastic Beanstalk precisará de uma implantação para realizar a blue/green atualização. A seguir estão as etapas gerais de melhores práticas que recomendamos para a migração com um procedimento blue/green de implantação.

Preparar para testar a migração

Antes de a aplicação e começar a testá-la, revise as informações em Considerações para todas as plataformas Linux que serão apresentadas posteriormente neste tópico. Além disso, revise as informações aplicáveis à sua plataforma na seção Considerações específicas da plataforma a seguir. Anote as informações específicas desse conteúdo que podem ser usadas na aplicação e na configuração definida.

Etapas de migração de alto nível

1. Crie um novo ambiente baseado em uma ramificação da plataforma AL2 ou AL2 023. Recomendamos que você migre para uma ramificação da plataforma AL2 023.
2. Implante seu aplicativo no ambiente AL2 023/ AL2 de destino.

   O ambiente de produção existente permanecerá ativo e não será afetado, enquanto você estiver fazendo interações por meio de testes e ajustes no novo ambiente.
3. Teste a aplicação minuciosamente no novo ambiente.
4. Quando seu AL2 ambiente AL2 023/ de destino estiver pronto para ser produzido, troque um CNAMEs dos dois ambientes para redirecionar o tráfego para o novo ambiente.

Etapas de migração mais detalhadas e práticas recomendadas

Para obter um procedimento de blue/green implantação mais detalhado, consulteImplantações azuis/verdes com o Elastic Beanstalk.

Para obter orientações mais específicas e etapas detalhadas de práticas recomendadas, consulte Método azul/verde.

Mais referências para ajudar a planejar a migração

As referências a seguir podem oferecer informações adicionais para planejar sua migração.

- Comparação entre o Amazon Linux 2 e o Amazon Linux 2023 no Guia do usuário do Amazon Linux 2023.
- O que é o Amazon Linux 2023? no Guia do usuário Amazon Linux 2023.
- Plataformas compatíveis com Elastic Beanstalk em Plataformas AWS Elastic Beanstalk
- Histórico de ramificações de plataforma retiradas
- the section called “Plataformas Linux”
- Perguntas frequentes sobre descontinuação de plataformas

Considerações para todas as plataformas Linux

A tabela a seguir discute as considerações que você deve conhecer ao planejar a migração de um aplicativo para AL2 023/. AL2 Essas considerações se aplicam a qualquer uma das plataformas

Linux do Elastic Beanstalk, independentemente de linguagens de programação específicas ou servidores de aplicações.

Área	Alterações e informações
Arquivos de configuração	Nas AL2 plataformas AL2 023/, você pode usar arquivos de configuração como antes, e todas as seções funcionam da mesma maneira. No entanto, configurações específicas podem não funcionar da mesma forma que funcionavam em plataform as da AMI do Amazon Linux anteriores. Por exemplo: • Alguns pacotes de software que você instala usando um arquivo de configuração podem não estar disponíveis em AL2 023/AL2, ou seus nomes podem ter sido alterados. • Algumas opções de configuração específica da plataforma mudaram de seus namespaces específicos da plataforma para namespaces diferentes e independe ntes de plataforma. • Os arquivos de configuração de proxy fornecidos no diretório `.ebextensions/nginx` devem ser movidos para o diretório de hooks da plataforma `.platform/nginx` . Para obter detalhes, consulte Configuração de proxy reverso. Recomendamos o uso de hooks de plataforma para executar o código personali zado em instâncias do seu ambiente. Você ainda pode usar comandos e os comandos de contêiner em arquivos de configuração `.ebextensions` , apesar de não ser fácil trabalhar com eles. Por exemplo, escrever scripts de comando dentro de um arquivo YAML pode ser complicado e difícil de testar. Você ainda precisa usar arquivos de configuração `.ebextensions` para qualquer script que precise de uma referência a um recurso do AWS CloudForm ation .
Hooks de plataforma	AL2 as plataformas introduziram uma nova maneira de estender a plataforma do seu ambiente adicionando arquivos executáveis aos diretórios de conexão nas instâncias do ambiente. É possível que você possa ter usado hooks de plataforma personalizada com versões anteriores da plataforma Linux. Esses hooks não foram projetados para plataformas gerenciadas e não eram compatíveis, mas podiam funcionar de maneira útil em alguns casos. Nas versões AL2 AL2 023/plataforma, os ganchos de plataforma personalizados não funcionam. Você deve migrar todos

Área	Alterações e informações
	os hooks para os novos hooks de plataforma. Para obter mais detalhes, consulte Hooks de plataforma.
Servidores de proxy compatíveis	AL2As versões da AL2 plataforma 023/ oferecem suporte aos mesmos servidores proxy reversos de cada plataforma suportada nas versões da plataforma Amazon Linux AMI. Todas as versões da plataforma AL2 023/AL2; usam o nginx como seu servidor proxy reverso padrão, com exceção das plataformas ECS e Docker. As plataformas Tomcat, Node.js, PHP e Python também oferecem suporte ao Apache HTTPD como uma alternativa. Todas as plataformas habilitam a configuração do servidor de proxy de forma uniforme, conforme descrito nesta seção. No entanto, configurar o servidor de proxy é ligeiramente diferente do que estava na AMI do Amazon Linux. Estas são as diferenças para todas as plataformas: • O padrão é nginx — O servidor proxy padrão em AL2 todas as versões da plataforma AL2 023/ é nginx. Nas versões da plataforma da AMI do Amazon Linux de Tomcat, PHP e Python, o servidor de proxy padrão era Apache HTTPD. • Namespace consistente — Todas as versões da AL2 plataforma AL2 023/ usam o `aws:elasticbeanstalk:environment:proxy` namespace para configurar o servidor proxy. Nas versões da plataforma da AMI do Amazon Linux, essa foi uma decisão de acordo com a plataforma, e o Node.js usava um namespace diferente. • Localização do arquivo de configuração — Você deve colocar os arquivos de configuração do proxy nos `.platform/httpd` diretórios `.platform/nginx` e em todas as versões da plataforma AL2 023/AL2 . Nas versões da plataforma da AMI do Amazon Linux, esses locais eram `.ebextensions/nginx` e `.ebextensions/httpd` , respectivamente. Para ver as alterações de configuração de proxy específicas da plataforma, consulte the section called “Considerações específicas da plataforma”. Para obter informações sobre a configuração de proxy nas AL2 plataformas AL2 023/, consulte. Configuração de proxy reverso

Área	Alterações e informações
Alterações de configuração de proxy	Algumas alterações de configuração de proxy se aplicam uniformemente a todas as plataformas, mas há também alterações de configuração de proxy específicas de cada plataforma. É importante consultar ambas para configurar os ambientes com precisão. • Todas as plataformas: consulte Configuração de proxy reverso • Platform-specific (Específico da plataforma): consulte the section called “Considerações específicas da plataforma”.
Perfil de instância	AL2AL2 As plataformas 023/ exigem que um perfil de instância seja configurado. A criação do ambiente pode ser bem-sucedida temporariamente sem um perfil, mas o ambiente poderá mostrar erros logo após a criação, quando as ações que exigem um perfil de instância começarem a falhar. Para obter detalhes, consulte the section called “Perfis de instância”.
Integridade avançada	AL2As versões da AL2 plataforma 023/ habilitam a integridade aprimorada por padrão. Trata-se de uma alteração, se você não usa o console do Elastic Beanstalk para criar seus ambientes. O console permite a integridade aprimorada por padrão sempre que possível, independentemente da versão da plataforma. Para obter detalhes, consulte the section called “Monitoramento e relatórios de integridade aprimorada”.
AMI personalizada	Se seu ambiente usa uma AMI personalizada, crie uma nova AMI com base em AL2 023/ AL2 para seu novo ambiente usando uma plataforma Elastic Beanstalk 023/. AL2 AL2
Plataformas personalizadas	As versões gerenciadas AMIs da AL2 plataforma AL2 023/ não oferecem suporte a plataformas personalizadas.

Considerações específicas da plataforma

Esta seção aborda considerações de migração específicas para determinadas plataformas Linux do Elastic Beanstalk.

Docker

A família de ramificações da plataforma Docker baseada no Amazon Linux AMI (AL1) inclui três ramificações da plataforma. Recomendamos um caminho de migração específico para cada.

AL1 Filial da plataforma	Caminho de migração para AL2 023/ AL2
Docker de vários contêineres gerenciado pelo Amazon ECS em execução no Amazon Linux AMI () AL1	Ramificações da plataforma Docker AL2 AL2 023/ baseadas em ECS As ramificações da AL2 plataforma Docker AL2 023/ baseadas em ECS oferecem um caminho de migração simples para ambientes executados na ramificação da plataforma Docker de vários contêineres. AL1 • Assim como a AL1 filial anterior do Docker de vários contêineres, as AL2 filiais da AL2 plataforma 023/ usam o Amazon ECS para coordenar a implantação de vários contêineres do Docker em um cluster do Amazon ECS em um ambiente do Elastic Beanstalk. • As ramificações da AL2 plataforma AL2 023/ oferecem suporte a todos os recursos da ramificação anterior do Docker de vários contêineres. AL1 • As ramificações da AL2 plataforma AL2 023/ também suportam o mesmo arquivo `Dockerrun.aws.json` v2. Para obter mais informações sobre como migrar seus aplicativos executados na ramificação da plataforma Docker Amazon Linux de vários contêineres para uma ramificação do Amazon ECS executada AL2 na ramificação da plataforma AL2 023/, consulte. ???
Docker em execução no Amazon Linux AMI () AL1 Docker pré-configurado	Docker em execução na ramificação da plataforma AL2 023/ AL2 Recomendamos que você migre seus aplicativos executados em ambientes baseados no Docker pré-configurado (Glassfish 5.0) ou no Docker em execução no Amazon Linux AMI (AL1) para ambientes baseados no Docker em execução no Amazon Linux 2 ou no Docker em execução nas filiais da plataforma 023. AL2

<table>
<tr><th>AL1 Filial da plataforma</th><th>Caminho de migração para AL2 023/ AL2</th></tr>
<tr><td>(Glassfish 5.0) executando o Amazon Linux AMI () AL1</td><td>
Se o ambiente for baseado na ramificação de plataforma Docker pré-configurado (Glassfish 5.0), consulte the section called “Tutorial - GlassFish no Docker: caminho para o Amazon Linux 2023”.

A tabela a seguir lista as informações de migração específicas para a ramificação da plataforma Docker Running on AL2 AL2 023/.

<table>
<tr><th>Área</th><th>Alterações e informações</th></tr>
<tr><td>Armazenamento</td><td>O Elastic Beanstalk configura o Docker para usar drivers de armazenamento para armazenar dados de contêineres e imagens do Docker. Na AMI do Amazon Linux, o Elastic Beanstalk usou o driver de armazenamento Device Mapper. Para melhorar a performance, o Elastic Beanstalk provisionou um volume extra do Amazon EBS. Nas versões da plataforma AL2 AL2 023/Docker, o Elastic Beanstalk usa o driver de armazenamento OverlayFS e obtém um desempenho ainda melhor sem precisar mais de um volume separado.

Com a AMI do Amazon Linux, se você usou a opção <code>BlockDeviceMappings</code> do namespace <code>aws:autoscaling:launchconfiguration</code> para adicionar volumes de armazenamento personalizados a um ambiente do Docker, recomendamos que você também adicione o volume do Amazon EBS <code>/dev/xvdcz</code> provisionado pelo Elastic Beanstalk. O Elastic Beanstalk não provisiona mais esse volume, portanto, é necessário removê-lo dos arquivos de configuração. Para obter detalhes, consulte the section called “Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)”.</td></tr>
<tr><td>Autenticação de repositór</td><td>Ao fornecer um arquivo de autenticação gerado pelo Docker para se conectar a um repositório privado, você não precisa mais convertê-lo para o formato antigo exigido pelas versões da plataforma Amazon Linux AMI Docker. AL2023/ As versões da plataforma AL2 Docker</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>AL1
Filial da plataforma</td><td>Caminho de migração para AL2 023/ AL2</td></tr>
<tr><td></td><td>
<table>
<tr><th>Área</th><th>Alterações e informações</th></tr>
<tr><td>io privado</td><td>suportam o novo formato. Para obter detalhes, consulte the section called “Usar imagens de um repositório privado”.</td></tr>
<tr><td>Servidor de proxy</td><td>AL2023/ As versões da plataforma AL2 Docker não oferecem suporte a contêineres autônomos que não são executados por trás de um servidor proxy. Nas versões da plataforma da AMI do Docker do Amazon Linux, isso costumava ser possível pelo valor none da opção <code>ProxyServer</code> no namespace <code>aws:elasticbeanstalk:environment:proxy</code> .</td></tr>
</table>
</td></tr>
</table>

Go

A tabela a seguir lista as informações de migração para as versões da AL2 plataforma AL2 023/ na plataforma Go.

Área	Alterações e informações
Transmissão de porta	Nas AL2 plataformas AL2 023/, o Elastic Beanstalk não passa um valor de porta para seu processo de aplicação por meio da variável de ambiente. `PORT` Você pode simular esse comportamento para o seu processo configurando uma propriedade de ambiente `PORT` por conta própria. No entanto, se você tiver vários processos e estiver contando com a transmissão de valores incrementais de porta do Elastic Beanstalk para os seus processos (5000, 5100, 5200 etc.), deverá modificar a implementação. Para obter mais detalhes, consulte Configuração de proxy reverso.

Amazon Corretto

A tabela a seguir lista as informações de migração para as ramificações de plataforma do Corretto na plataforma Java SE.

Área	Alterações e informações
Corretto vs. OpenJDK	Para implementar a Plataforma Java, Standard Edition (Java SE), as ramificações da AL2 plataforma AL2 023/ usam o Amazon Corretto, uma AWS distribuição do Open Java Development Kit (OpenJDK). As ramificações anteriores da plataform a Java SE do Elastic Beanstalk usam os pacotes OpenJDK incluídos na AMI do Amazon Linux.
Ferrament as de compilação	AL2AL2 As plataformas 023/ têm versões mais recentes das ferramentas de construção:`gradle`, e. `maven` `ant`
Tratament o de arquivos JAR	Nas AL2 plataformas AL2 023/, se seu pacote de origem (arquivo ZIP) contiver um único arquivo JAR e nenhum outro arquivo, o Elastic Beanstalk não renomeia mais o arquivo JAR para. `application.jar` A renomeação ocorrerá somente se você enviar um arquivo JAR sozinho, e não dentro de um arquivo ZIP.
Transmiss ão de porta	Nas AL2 plataformas AL2 023/, o Elastic Beanstalk não passa um valor de porta para seu processo de aplicação por meio da variável de ambiente. `PORT` Você pode simular esse comportamento para o seu processo configurando uma propriedade de ambiente `PORT` por conta própria. No entanto, se você tiver vários processos e estiver contando com a transmissão de valores incrementais de porta do Elastic Beanstalk para os seus processos (5000, 5100, 5200 etc.), deverá modificar a implementação. Para obter mais detalhes, consulte Configuração de proxy reverso.
Java 7	O Elastic Beanstalk não é AL2 compatível com AL2 uma ramificação da plataform a 023/Java 7. Se você tiver uma aplicação Java 7, migre-a para Corretto 8 ou Corretto 11.

Tomcat

A tabela a seguir lista as informações de migração para as versões da AL2 plataforma AL2 023/ na plataforma Tomcat.

<table>
<tr><th>Área</th><th>Alterações e informações</th></tr>
<tr><td>Opções de configuração</td><td>Nas versões da AL2 plataforma AL2 023/, o Elastic Beanstalk suporta somente um subconjunto das opções de configuração e dos valores das opções no namespace . <code>aws:elasticbeanstalk:environment:proxy</code> Veja a seguir as informações sobre a migração para cada opção.

<table>
<tr><th>Opção</th><th>Informações sobre migração</th></tr>
<tr><td><code>GzipCompression</code></td><td>Não suportado nas versões da AL2 plataforma 023/AL2 .</td></tr>
<tr><td><code>ProxyServer</code></td><td>AL2023/ As versões da plataforma AL2 Tomcat suportam os servidores proxy nginx e Apache HTTPD versão 2.4. No entanto, não há suporte para o Apache versão 2.2.

Nas versões da plataforma da AMI do Amazon Linux, o proxy padrão era Apache 2.4. Se você usou a configuração de proxy padrão e adicionou arquivos de configuração de proxy personalizados, sua configuração de proxy ainda deve funcionar em AL2 023/AL2. No entanto, se você usou o valor da opção <code>apache/2.2</code> , agora você precisa migrar a configuração de proxy para o Apache versão 2.4.</td></tr>
</table>

A <code>XX:MaxPermSize</code> opção no <code>aws:elasticbeanstalk:container:tomcat:jvmoptions</code> namespace não é suportada nas versões AL2 AL2 023/ da plataforma. A configuração da JVM para modificar o tamanho da geração permanente se aplica somente ao Java 7 e versões anteriores e, portanto, não é aplicável às versões da plataforma AL2 023/AL2 .</td></tr>
<tr><td>Caminho do aplicativo</td><td>Nas AL2 plataformas AL2 023/, o caminho para o diretório do aplicativo nas EC2 instâncias Amazon do seu ambiente é. <code>/var/app/current</code> Era <code>/var/lib/tomcat8/webapps</code> em plataformas da AMI do Amazon Linux.</td></tr>
</table>

Node.js

A tabela a seguir lista as informações de migração para as versões da AL2 plataforma AL2 023/ na plataforma Node.js.

<table>
<tr><th>Área</th><th>Alterações e informações</th></tr>
<tr><td>Versões Node.js instaladas</td><td>Nas AL2 plataformas AL2 023/, o Elastic Beanstalk mantém várias ramificações da plataforma Node.js e instala somente a versão mais recente da versão principal do Node.js correspondente à ramificação da plataforma em cada versão da plataforma. Por exemplo, cada versão da plataforma na ramificação da plataforma Node.js 12 tem somente o Node.js 12.x.y instalado por padrão. Nas versões da plataforma da AMI do Amazon Linux, instalamos as várias versões do Node.js em cada versão da plataforma e mantemos apenas uma única ramificação da plataforma.

Escolha a ramificação da plataforma Node.js que corresponde à versão principal do Node.js de que seu aplicativo precisa.</td></tr>
<tr><td>Nomes de arquivos de log do Apache HTTPD</td><td>Nas AL2 plataformas AL2 023/, se você usa o servidor proxy Apache HTTPD, os nomes dos arquivos de log HTTPD são <code>access_log</code> <code>eerror_log</code> , o que é consistente com todas as outras plataformas que suportam o Apache HTTPD. Nas versões da plataforma da AMI do Amazon Linux, esses arquivos de log eram chamados de <code>access.log</code> e <code>error.log</code> , respectivamente.

Para obter detalhes sobre nomes de arquivos de log e locais de todas as plataform as, consulte the section called “Como o Elastic Beanstalk configura os registros CloudWatch ”.</td></tr>
<tr><td>Opções de configura ção</td><td>Nas AL2 plataformas AL2 023/, o Elastic Beanstalk não oferece suporte às opções de configuração no namespace. <code>aws:elasticbeanstalk:contai ner:nodejs</code> Algumas das opções têm alternativas. Veja a seguir as informaçõ es sobre a migração para cada opção.

<table><tr><th>Opção</th><th>Informações sobre migração</th></tr><tr><td><code>NodeComm</code> nd</td><td>Use um <code>scripts</code> ou a palavra-chave <code>Procfile</code> em um arquivo <code>package.json</code> para especificar o script de início.</td></tr></table></td></tr>
</table>

Área	Alterações e informações

Opção	Informações sobre migração
`NodeVersion`	Use a palavra-chave `engines` em um arquivo `package.json` para especificar a versão do Node.js. Lembre-se de que é possível especificar somente uma versão do Node.js que corresponda à sua ramificação da plataforma. Por exemplo, se você estiver usando a ramificação da plataforma Node.js 12, será possível especific ar somente uma versão 12.x.y do Node.js. Para obter detalhes, consulte the section called “Especificar as dependências do Node.js com um arquivo package.json”.
`GzipCompression`	Não suportado nas versões da AL2 plataforma 023/AL2 .

`ProxyServer`	Nas versões da plataforma AL2 023/ AL2 Node.js, essa opção foi movida para o `aws:elasticbeanstalk:environment:proxy` namespace. É possível escolher entre `nginx` (o padrão) e `apache`.
	
	AL2As versões da plataforma 023/ AL2 Node.js não oferecem suporte a aplicativos autônomos que não são executados por trás de um servidor proxy. Nas versões da plataforma Node.js da AMI do Amazon Linux, isso costumava ser possível pelo valor `none` da opção `ProxyServer` no namespace `aws:elasticbeanstalk:container:nodejs` . Se o ambiente executar uma aplicação autônoma, atualize o código para escutar a porta para a qual servidor de proxy (nginx ou Apache) encaminha o tráfego.
	
	`var port = process.env.PORT || 5000;`
	
	`app.listen(port, function() {`
	`  console.log('Server running at http://127.0.0.1:%s',`
	` port);`
	`});`

PHP

A tabela a seguir lista as informações de migração para as versões da AL2 plataforma AL2 023/ na plataforma PHP.

Área	Alterações e informações
Processamento de arquivos PHP	Nas AL2 plataformas AL2 023/, os arquivos PHP são processados usando o PHP-FPM (um gerenciador de processos CGI). Em plataformas da AMI do Amazon Linux, usamos mod_php (um módulo do Apache).
Servidor de proxy	AL2As versões da plataforma 023/ AL2 PHP suportam os servidores proxy nginx e Apache HTTPD. O padrão é nginx. As versões da plataforma PHP da AMI do Amazon Linux eram compatíveis apenas com o Apache HTTPD. Se você adicionou arquivos de configuração personalizados do Apache, poderá definir a opção `ProxyServer` no namespace `aws:elasticbeanstalk:environment:proxy` como `apache`.

Python

A tabela a seguir lista as informações de migração para as versões da AL2 plataforma AL2 023/ na plataforma Python.

Área	Alterações e informações
Servidor WSGI	Nas AL2 plataformas AL2 023/, o Gunicorn é o servidor WSGI padrão. Por padrão, o Gunicorn escuta na porta 8000. A porta pode ser diferente da utilizada pela aplicação na plataforma da AMI do Amazon Linux. Se você estiver definindo a opção `WSGIPath` do namespace `aws:elasticbeanstalk:container:python`, substitua o valor pela sintaxe de Gunicorn. Para obter detalhes, consulte the section called “Namespaces de configuração do Python”. Como alternativa, você pode usar um `Procfile` para especificar e configurar o servidor WSGI. Para obter detalhes, consulte the section called “Procfile”.

Área	Alterações e informações
Caminho do aplicativo	Nas AL2 plataformas AL2 023/, o caminho para o diretório do aplicativo nas EC2 instâncias Amazon do seu ambiente é. `/var/app/current` Era `/opt/python/current/app` em plataformas da AMI do Amazon Linux.
Servidor de proxy	AL2023/ As versões da plataforma AL2 Python suportam os servidores proxy nginx e Apache HTTPD. O padrão é nginx. As versões da plataforma Python da AMI do Amazon Linux eram compatíveis apenas com o Apache HTTPD. Se você adicionou arquivos de configuração personalizados do Apache, poderá definir a opção `ProxyServer` no namespace `aws:elasticbeanstalk:environment:proxy` como `apache`.

Ruby

A tabela a seguir lista as informações de migração para as versões da AL2 plataforma AL2 023/ na plataforma Ruby.

Área	Alterações e informações
Versões Ruby instaladas	Nas AL2 plataformas AL2 023/, o Elastic Beanstalk instala somente a versão mais recente de uma única versão do Ruby, correspondente à ramificação da plataforma, em cada versão da plataforma. Por exemplo, cada versão da plataforma na ramificação da plataforma Ruby 2.6 tem somente o Ruby 2.6.x instalado. Nas versões da plataforma da AMI do Amazon Linux, instalamos as versões mais recentes de várias versões do Ruby, por exemplo, 2.4.x, 2.5.x e 2.6.x. Se seu aplicativo usar uma versão do Ruby que não corresponda à ramificação da plataforma que você está usando, recomendamos mudar para uma ramificação de plataforma que tenha a versão do Ruby correta para seu aplicativo.
Servidor de aplicativos	Nas AL2 plataformas AL2 023/, o Elastic Beanstalk só instala o servidor de aplicativos Puma em todas as versões da plataforma Ruby. É possível usar um `Procfile` para iniciar um servidor de aplicativos diferente e um `Gemfile` para instalá-lo. Na plataforma da AMI do Amazon Linux, oferecemos suporte a dois tipos de ramificações da plataforma para cada versão Ruby: uma com o servidor de

Área	Alterações e informações
	aplicações Puma e outra com o servidor de aplicações Passenger. Se o seu aplicativo usar o Passenger, é possível configurar seu ambiente Ruby para instalar e usar o Passenger. Para ter mais informações e exemplos, consulte the section called “A Plataforma Ruby”.

Perguntas frequentes sobre descontinuação de plataformas

> **Note**
>
> O Elastic Beanstalk retirou todas as ramificações da plataforma com base no Amazon Linux AL1 AMI () em 18 de julho de 2022.

As respostas destas Perguntas frequentes referenciam os seguintes tópicos:

- Política de suporte às plataformas do Elastic Beanstalk
- Histórico de ramificações de plataforma retiradas
- Plataformas compatíveis com Elastic Beanstalk em Plataformas AWS Elastic Beanstalk
- Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2
- Amazon Linux 2 FAQs.

1. O que significa a retirada de uma ramificação de plataforma?

Após a data de descontinuação de uma ramificação de plataforma ser anunciada, você não poderá mais criar um novo ambiente baseado na ramificação de plataforma retirada, a menos que já tenha um ambiente ativo baseado nessa ramificação de plataforma. Para obter mais informações, consulte a pergunta frequentes nº 11. O Elastic Beanstalk deixará de fornecer novas atualizações de manutenção para essas ramificações de plataforma. Não é recomendável usar uma ramificação de plataforma retirada em ambientes de produção. Para obter mais informações, consulte a pergunta frequentes nº 5.

2. Por que AWS retirou as filiais de plataformas AL1 baseadas?

O Elastic Beanstalk retira ramificações de plataforma quando os componentes da plataforma ficam obsoletos ou são retirados pelos fornecedores. Nesse caso, o Amazon Linux AMI (AL1) encerrou o suporte padrão em 31 de dezembro de 2020. Embora o Elastic Beanstalk tenha continuado AL1 a oferecer plataformas baseadas até 2022, AL2 lançamos, AL2 desde então, uma versão 023 e plataformas baseadas que têm os recursos mais recentes. Para que os clientes continuem se beneficiando da segurança e dos recursos mais recentes no futuro, é fundamental que os clientes migrem para nossas plataformas AL2 ou baseadas em AL2 023.

3. Quais ramificações de plataforma serão retiradas?

Para obter uma lista de componentes de plataforma e ramificações de plataforma que foram retirados, consulte Histórico de ramificações de plataforma retiradas.

4. Quais são as plataformas compatíveis atualmente?

Consulte Plataformas compatíveis com Elastic Beanstalk em Plataformas do AWS Elastic Beanstalk .

5. O Elastic Beanstalk removerá ou encerrará algum componente do ambiente após a retirada?

Nossa política para ramificações de plataforma descontinuadas não remove o acesso a ambientes nem exclui recursos. Porém, um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível, pois o Elastic Beanstalk não pode fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradas, uma vez que o fornecedor o componente definiu que o componente atingiu o Fim da Vida Útil (EOL). Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram.

Se a aplicação encontrar problemas ao ser executada em uma ramificação de plataforma descontinuada e você não conseguir migrá-la para uma plataforma compatível, será necessário considerar outras alternativas. Soluções alternativas incluem o encapsulamento da aplicação em uma imagem do Docker para executá-la como um contêiner do Docker. Isso permitiria que um

cliente usasse qualquer uma de nossas soluções Docker, como nossas plataformas AL2 Elastic AL2 Beanstalk 023/Docker ou outros serviços baseados em Docker, como Amazon ECS ou Amazon EKS. As alternativas que não são do Docker incluem nosso AWS CodeDeploy serviço, que permite a personalização completa dos tempos de execução que você deseja.

6. Posso enviar uma solicitação para adiar a data de retirada?

Não. Após a data de retirada, os ambientes existentes continuarão a funcionar. Porém, o Elastic Beanstalk não fornecerá mais atualizações de manutenção e segurança da plataforma. Portanto, é fundamental migrar para AL2 ou AL2 023 se você ainda estiver executando aplicativos em uma plataforma AL1 baseada. Para obter mais informações sobre riscos e soluções alternativas, consulte a pergunta frequente nº 5.

7. Quais são as soluções alternativas se eu não conseguir concluir minha migração AL2 ou AL2 023 a tempo?

Os clientes podem continuar a executar o ambiente, embora seja muito recomendável planejar a migração de todos os ambientes do Elastic Beanstalk para uma versão de plataforma compatível. Isso minimizará o risco e proporcionará os benefícios contínuos das importantes melhorias de funcionalidade, performance e segurança oferecidas nas versões mais recentes. Para obter mais informações sobre riscos e soluções alternativas, consulte a pergunta frequente nº 5.

8. Qual é o processo recomendado para migrar para AL2 ou AL2 023 plataformas?

Para obter instruções abrangentes de AL2 migração AL1 para AL2 023/, consulte. Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2 Este tópico explica que o Elastic Beanstalk exige uma implantação azul/verde para realizar a atualização.

9. Se eu tiver um ambiente em execução em uma plataforma retirada, qual seria o impacto?

Um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível, pois o Elastic Beanstalk não pode fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradas, uma vez que o fornecedor definiu que o componente atingiu o Fim da Vida Útil (EOL). Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma

ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram. Para obter mais informações, consulte a pergunta frequentes nº 5.

10. O que acontece 90 dias após a data de retirada?

Nossa política para ramificações de plataforma descontinuadas não remove o acesso a ambientes nem exclui recursos. Porém, lembre-se de que um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível, pois o Elastic Beanstalk não pode fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradas, uma vez que o fornecedor definiu que o componente atingiu o Fim da Vida Útil (EOL). Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram. Para obter mais informações, consulte a pergunta frequentes nº 5.

11. Posso criar um novo ambiente baseado em uma plataforma retirada?

Você pode criar um novo ambiente baseado em uma ramificação de plataforma retirada se já tiver usado essa ramificação de plataforma para criar um ambiente existente usando a mesma conta e na mesma região. A ramificação da plataforma descontinuada não estará disponível no console do Elastic Beanstalk. Porém, para clientes que tenham ambientes existentes baseados em uma ramificação de plataforma retirada, ela estará disponível por meio da CLI do EB, da API do EB e da AWS CLI. Além disso, os clientes existentes podem usar os consoles Clone environment (Clonar ambiente) e Rebuild environment (Recriar ambiente). Porém, lembre-se de que um ambiente baseado em uma ramificação de plataforma retirada pode levar a uma situação imprevisível. Para obter mais informações, consulte a pergunta frequentes nº 5.

12. Se tiver um ambiente existente em execução em uma ramificação de plataforma descontinuada, até quando poderei criar um novo ambiente com base nessa ramificação? Posso fazer isso usando console, CLI ou API?

É possível criar o ambiente após a data de retirada. Porém, lembre-se de que uma ramificação de plataforma retirada pode levar a uma situação imprevisível. Quanto mais tempo após a retirada esse ambiente for criado ou permanecer ativo, maior será o risco de problemas inesperados. Para obter mais informações sobre a criação de um novo ambiente de desenvolvimento, consulte a pergunta frequente nº 11.

13. É possível clonar ou recriar um ambiente baseado em plataforma retirada?

Sim. Você pode fazer isso usando os consoles Clone environment (Clonar ambiente) e Rebuild environment (Recriar ambiente). Você também pode usar a CLI do EB, a API do EB e. AWS CLI Para obter mais informações sobre a criação de um novo ambiente de desenvolvimento, consulte a pergunta frequente nº 11.

Porém, é muito recomendável planejar a migração de todos os ambientes do Elastic Beanstalk para uma versão de plataforma compatível. Isso minimizará o risco e proporcionará os benefícios contínuos das importantes melhorias de funcionalidade, performance e segurança oferecidas nas versões mais recentes. Para obter mais informações sobre riscos e soluções alternativas, consulte a pergunta frequente nº 5.

14. Após a data de aposentadoria, o que aconteceria com os AWS recursos do meu ambiente do Elastic Beanstalk, baseado em uma ramificação de plataforma desativada? Por exemplo, se a EC2 instância em execução for encerrada, o Elastic Beanstalk conseguiria lançar AL1 uma EC2 nova instância baseada para manter a capacidade?

Os recursos do ambiente permaneceriam ativos e continuariam funcionando. E, sim, o Elastic Beanstalk será escalado AL1 EC2 automaticamente para instâncias no ambiente. Porém, o Elastic Beanstalk deixará de fornecer novas atualizações de manutenção da plataforma para o ambiente, o que pode, com o tempo. levar o ambiente a uma situação imprevisível. Para obter mais informações, consulte a pergunta frequentes nº 5.

15. Quais são as principais diferenças entre os sistemas operacionais AL2 023/ e AL2 Amazon Linux AMI (AL1)? Como as ramificações da plataforma Elastic AL2 Beanstalk AL2 023/ são afetadas?

Embora o Amazon Linux AMI e o AL2 023/ AL2 compartilhem o mesmo kernel Linux, eles diferem no sistema de inicialização, nas `libc` versões, na cadeia de ferramentas do compilador e em vários pacotes. Para obter mais informações, consulte Amazon Linux 2 FAQs.

O serviço Elastic Beanstalk também atualizou as versões de runtime, ferramentas de compilação e outras dependências específicas de cada plataforma. Não é garantido que as ramificações da plataforma AL2 baseadas em AL2 023/ sejam compatíveis com versões anteriores de seu aplicativo existente. Além disso, mesmo que o código do aplicativo seja implantado com sucesso na nova versão da plataforma, ele pode se comportar ou ser executado de forma diferente devido às

diferenças de runtime e sistema operacional. Para obter uma lista e a descrição das configurações e personalizações que você precisará revisar e testar, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Cancelar atualizações de configuração do ambiente e implantações de aplicativos

Você pode cancelar atualizações em andamento que são acionadas por alterações da configuração do ambiente. Também pode cancelar a implantação de uma nova versão do aplicativo em andamento. Por exemplo, pode ser necessário cancelar uma atualização caso você decida continuar usando a configuração de ambiente existente em vez de aplicar novas configurações do ambiente. Ou talvez você perceba que a nova versão do aplicativo que está sendo implantada tem problemas que impedirão sua inicialização ou execução correta. Cancelando uma atualização de ambiente ou de versão de aplicativo, você pode iniciar uma nova tentativa de atualizar o ambiente ou a versão do aplicativo sem ter que esperar até o fim do processo de atualização ou implantação.

> **Note**
>
> Durante a fase de limpeza em que recursos antigos desnecessários são removidos, após a atualização do último lote de instâncias, você não pode mais cancelar a atualização.

O Elastic Beanstalk executa a reversão do mesmo modo como executou a última atualização bem-sucedida. Por exemplo, se as atualizações contínuas baseadas no tempo estiverem habilitadas no seu ambiente, o Elastic Beanstalk aguardará o tempo de espera especificado entre a reversão das alterações em um lote de instâncias antes de reverter as alterações do próximo lote. Caso você tenha ativado recentemente as atualizações contínuas, mas a última atualização que fez nas configurações do ambiente não teve atualizações contínuas, o Elastic Beanstalk executará a reversão em todas as instâncias simultaneamente.

Não é possível impedir que o Elastic Beanstalk reverta à configuração de ambiente anterior depois que ele começa a cancelar a atualização. O processo de reversão continua até que todas as instâncias no ambiente tenham a configuração de ambiente anterior ou até que o processo de reversão falhe. Para implantações de versão do aplicativo, o cancelamento da implantação apenas interrompe a implantação. Algumas instâncias terão a nova versão do aplicativo enquanto outras continuarão a executar a versão do aplicativo existente. Você pode implantar a mesma versão do aplicativo ou implantar outra mais tarde.

Para obter mais informações sobre atualizações contínuas, consulte Atualizações de configuração de ambiente contínuas do Elastic Beanstalk. Para obter mais informações sobre as implantações de versão do aplicativo em lote, consulte Políticas e configurações de implantação.

Para cancelar uma atualização

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Actions (Ações) e depois Abort current operation (Abortar operação atual).

Recriar ambientes do Elastic Beanstalk

Seu AWS Elastic Beanstalk ambiente pode se tornar inutilizável se você não usar a funcionalidade do Elastic Beanstalk para modificar ou encerrar os recursos subjacentes do ambiente. AWS Se isso acontecer, você poderá reconstruir o ambiente para tentar restaurá-lo a um estado operacional. A reconstrução de um ambiente encerra todos os seus recursos e os substitui por novos recursos com a mesma configuração.

Você também pode reconstruir ambientes em até seis semanas (42 dias) de seu encerramento. Ao recriar, o Elastic Beanstalk tenta criar outro ambiente com o mesmo nome, ID e configuração.

Reconstrução de um ambiente em execução

É possível recriar um ambiente por meio do console do Elastic Beanstalk ou usando a API `RebuildEnvironment`.

> **Warning**
>
> Se seu ambiente tiver um banco de dados acoplado, ele será excluído como parte da reconstrução, e o novo banco de dados no ambiente reconstruído não conterá os dados anteriores. Se você quiser manter o banco de dados ou tirar um instantâneo, certifique-se de ter a política de exclusão do banco de dados configurada adequadamente para obter os resultados desejados após a reconstrução. Para obter mais informações, consulte Ciclo de vida do banco de dados.

Para reconstruir um ambiente em execução (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions e Rebuild environment.
4. Escolha Rebuild.

Para reconstruir um ambiente em execução com a API do Elastic Beanstalk, `RebuildEnvironment`use a ação com o ou com o SDK. AWS CLI AWS

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Reconstruir um ambiente encerrado

É possível recriar e restaurar um ambiente encerrado ao usar o console do Elastic Beanstalk, a CLI do EB ou a API `RebuildEnvironment`.

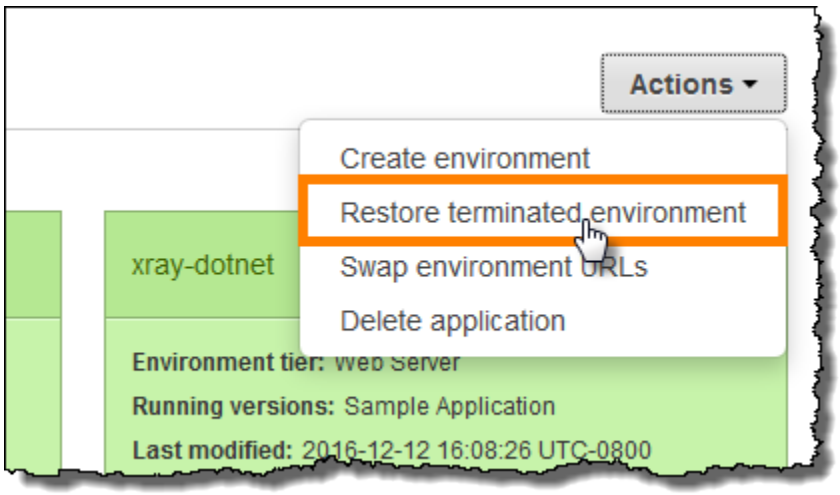
> **Note**
>
> A menos que você esteja usando seu próprio nome de domínio personalizado com o ambiente encerrado, o ambiente usará um subdomínio de elasticbeanstalk.com. Esses subdomínios são compartilhados dentro de uma região do Elastic Beanstalk. Portanto, eles podem ser usados por qualquer ambiente criado por qualquer cliente na mesma região. Embora seu ambiente esteja encerrado, um outro ambiente pode usar seu subdomínio. Nesse caso, a reconstrução não ocorrerá.
> Evite esse problema usando um domínio personalizado. Para mais detalhes, consulte Nome de domínio do ambiente do Elastic Beanstalk.

Os ambientes encerrados recentemente são exibidos na visão geral do aplicativo por até uma hora. Durante esse período, você pode visualizar eventos do ambiente em seu painel e usar a ação Restore environment para reconstruí-lo.

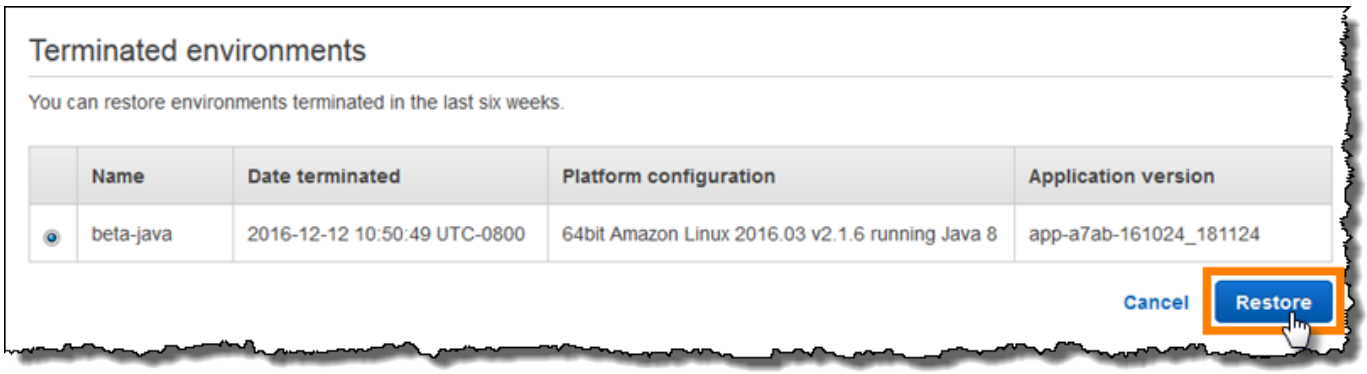
Para reconstruir um ambiente que não está mais visível, use a opção Restore terminated environment na página do aplicativo.

Para reconstruir um ambiente encerrado (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. Escolha Actions e, em seguida, Restore terminated environment.



4. Escolha um ambiente encerrado.
5. Escolha Restore.



O Elastic Beanstalk tenta criar um novo ambiente com o mesmo nome, ID e configuração. Se já existir um ambiente com o mesmo nome ou URL, quando você tentar reconstruir, a ação falhará. A exclusão da versão do aplicativo que foi implantada no ambiente também causará falha na reconstrução.

Se você usa a EB CLI para gerenciar seu ambiente, use o comando eb restore para reconstruir um ambiente encerrado.

```
$ eb restore e-vdnftxubwq
```

Consulte eb restore para obter mais informações.

Para reconstruir um ambiente encerrado com a API do Elastic Beanstalk, use `RebuildEnvironment`a ação com o ou com o SDK. AWS CLI AWS

```
$ aws elasticbeanstalk rebuild-environment --environment-id e-vdnftxubwq
```

Tipos de ambiente

Em AWS Elastic Beanstalk, você pode criar um ambiente escalável e com balanceamento de carga ou um ambiente de instância única. O tipo de ambiente necessário depende do aplicativo que você implanta. Por exemplo, é possível desenvolver e testar um aplicativo em um ambiente de instância única para reduzir custos e, depois, atualizá-lo para um ambiente escalável e com balanceamento de carga quando a aplicação estiver pronta para produção.

> **Note**
>
> Uma camada de ambiente de operador para um aplicativo web que processa tarefas em segundo plano não inclui um load balancer. No entanto, um ambiente de operador é expandido de modo eficiente adicionando instâncias ao grupo de Auto Scaling para processar dados da fila do Amazon SQS quando a carga necessita.

Ambiente escalável e com balanceamento de carga

Um ambiente escalável e com balanceamento de carga usa os serviços Elastic Load Balancing e Amazon Auto EC2 Scaling para provisionar as EC2 instâncias da Amazon que são necessárias para seu aplicativo implantado. O Amazon EC2 Auto Scaling inicia automaticamente instâncias adicionais para acomodar o aumento da carga em seu aplicativo. Se a carga em seu aplicativo diminuir, o Amazon EC2 Auto Scaling interrompe as instâncias, mas sempre deixa o número mínimo especificado de instâncias em execução. Se a aplicação exigir escalabilidade com a opção de execução em várias zonas de disponibilidade, use um ambiente escalável e com balanceamento de carga. Se não tiver certeza sobre qual tipo de ambiente selecionar, você poderá escolher um e, se necessário, mudar o tipo do ambiente mais tarde.

Ambiente de instância única

Um ambiente de instância única contém uma EC2 instância da Amazon com um endereço IP elástico. Um ambiente de instância única não tem um load balancer, o que pode ajudar a reduzir custos em comparação com um ambiente escalável e com balanceamento de carga. Embora um ambiente de instância única use o serviço Amazon EC2 Auto Scaling, as configurações para o número mínimo de instâncias, o número máximo de instâncias e a capacidade desejada estão todas definidas como 1. Dessa forma, as novas instâncias não são iniciadas para acomodar a crescente carga no aplicativo.

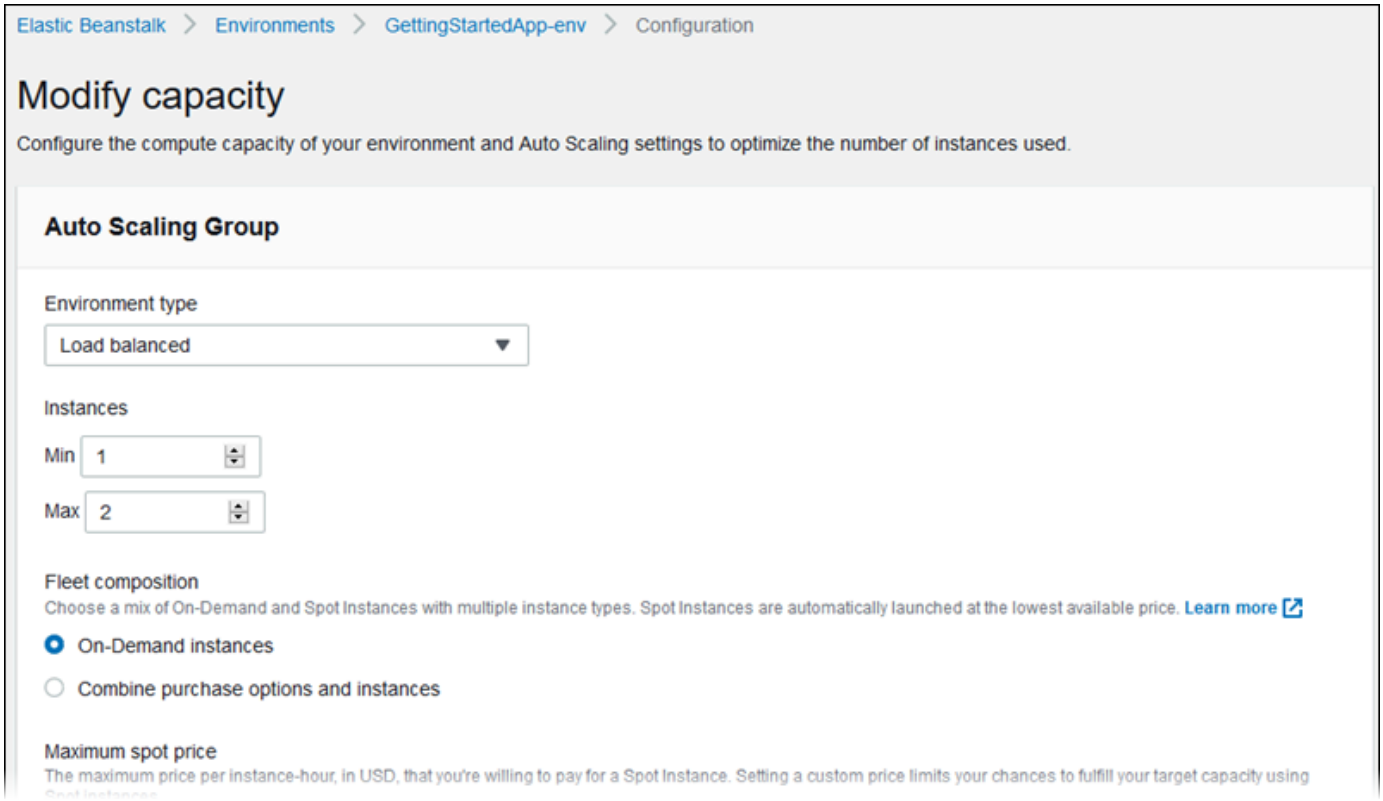
Use um ambiente de instância única se você espera que o aplicativo de produção tenha pouco tráfego ou se você está fazendo um desenvolvimento remoto. Se você não tiver certeza sobre qual tipo de ambiente selecionar, poderá escolher um e, se necessário, alterá-lo posteriormente. Para obter mais informações, consulte Alterar tipo de ambiente.

Alterar tipo de ambiente

É possível alterar o tipo de ambiente para um de instância única ou escalável e com balanceamento de carga editando a configuração do ambiente. Em alguns casos, convém alterar de um tipo de ambiente para outro. Por exemplo, suponha que você desenvolveu e testou um aplicativo em um ambiente de instância única para reduzir custos. Quando a aplicação estiver pronta para produção, você poderá alterar o tipo de ambiente para um escalável e com balanceamento de carga, de modo que ele possa ser dimensionado para atender às demandas dos clientes.

Para alterar um tipo de ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria Capacity (Capacidade), escolha Edit (Editar).
5. Na lista Environment Type (Tipo de ambiente), selecione o tipo de ambiente desejado.



6. Escolha Salvar.

   Pode levar vários minutos para que o ambiente seja atualizado enquanto o Elastic Beanstalk provisiona recursos AWS .

Se seu ambiente estiver em uma VPC, selecione sub-redes nas quais colocar as instâncias do Elastic Load Balancing e da Amazon. EC2 Cada zona de disponibilidade na qual seu aplicativo é executado deve ter as duas. Para mais detalhes, consulte Usar o Elastic Beanstalk com Amazon VPC.

Ambientes de operador do Elastic Beanstalk

Se seu AWS Elastic Beanstalk aplicativo executa operações ou fluxos de trabalho que demoram muito para serem concluídos, você pode transferir essas tarefas para um ambiente de trabalho dedicado. O desacoplamento do front-end do aplicativo web de um processo que executa operações de bloqueio é uma maneira comum de garantir que seu aplicativo mantenha a capacidade de resposta sob carga.

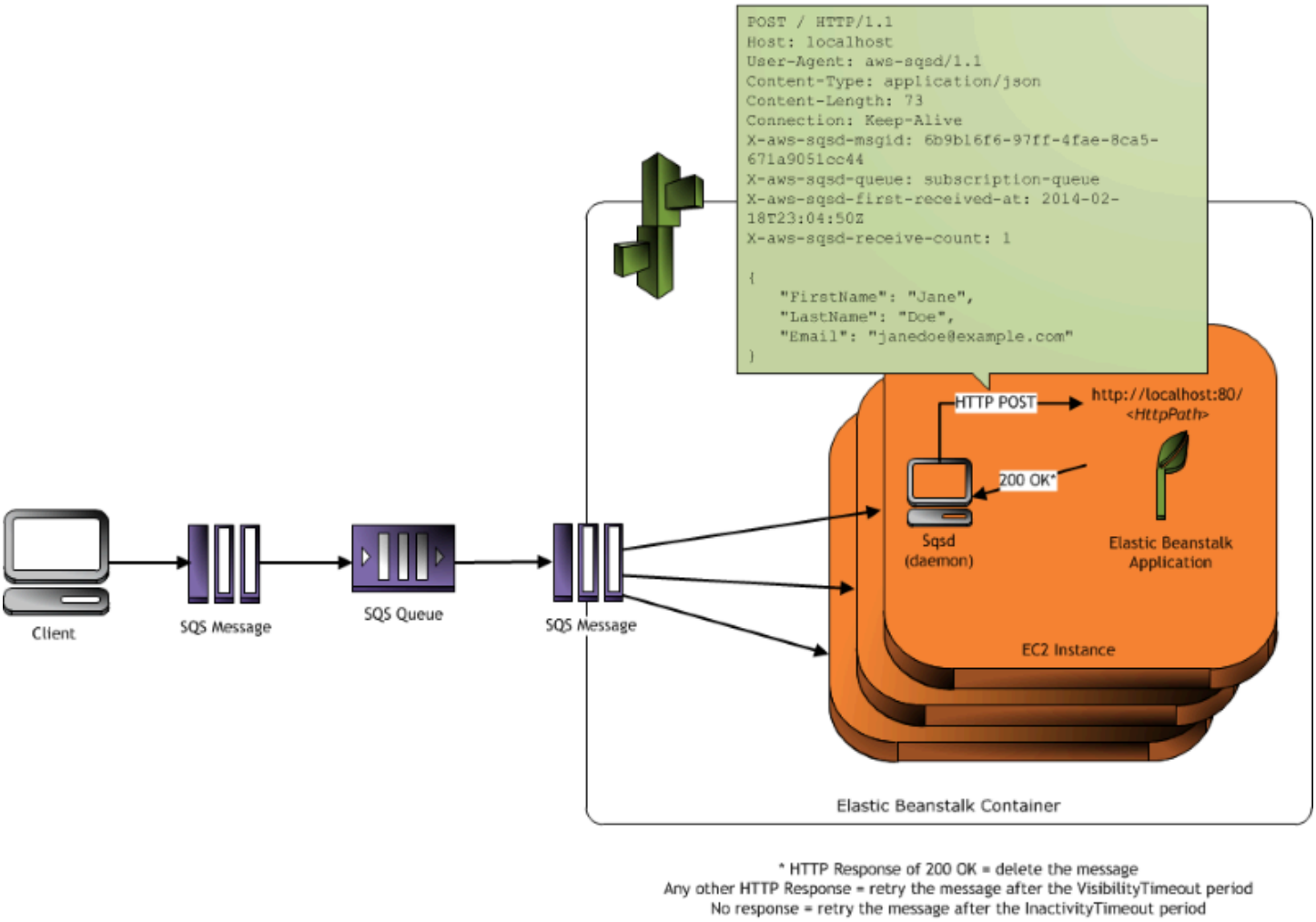
Uma tarefa de longa execução é alguma coisa que aumenta significativamente o tempo necessário para concluir uma solicitação, como o processamento de imagens ou vídeos, o envio de e-mail ou a

geração de um arquivo ZIP. Essas operações podem levar apenas um ou dois segundos para serem concluídas, mas um atraso de alguns segundos é muito para uma solicitação web que, de outra forma, seria concluída em menos de 500 ms.

Uma opção é criar um processo de operador localmente, retornar o êxito e processar a tarefa de maneira assíncrona. Isso funcionará, se a instância puder acompanhar todas as tarefas que são enviadas a ela. Sob carga elevada, no entanto, uma instância pode ficar sobrecarregada com tarefas em segundo plano e parar de responder às solicitações de prioridade mais alta. Se cada usuário puder gerar várias tarefas, o aumento na carga pode não corresponder a um aumento nos usuários, o que dificulta expandir o nível do servidor Web de modo eficaz.

Para evitar a execução local de tarefas demoradas, você pode usar o AWS SDK da sua linguagem de programação para enviá-las para uma fila do Amazon Simple Queue Service (Amazon SQS) e executar o processo que as executa em um conjunto separado de instâncias. Depois, você cria essas instâncias de operador para obter itens da fila apenas quando elas têm capacidade para executá-los, impedindo que se tornem sobrecarregadas.

Os ambientes de operador do Elastic Beanstalk simplificam esse processo gerenciando a fila do Amazon SQS e executando um processo de daemon em cada instância que lê a fila para você. Quando o daemon extrai um item da fila, envia uma solicitação HTTP POST localmente para `http://localhost/`, na porta 80, com o conteúdo da mensagem da fila no corpo. Tudo o que seu aplicativo precisa fazer é realizar a tarefa de longa execução em resposta ao POST. Você pode configurar o daemon para publicar em um caminho diferente, usar um tipo MIME que não seja aplicativo/JSON, conectar-se a uma fila existente ou personalizar conexões (máximo de solicitações simultâneas), tempos limite e novas tentativas.



Com as tarefas periódicas, você também pode configurar o daemon do operador para colocar as mensagens na fila com base em uma programação cron. Cada tarefa periódica pode executar POST em um caminho diferente. Para habilitar as tarefas periódicas, inclua um arquivo YAML em seu código-fonte que define a programação e o caminho para cada tarefa.

> **Note**
>
> O .NET na Plataforma de Windows Server não tem suporte para ambientes de operador.

Seções

- Daemon SQS do ambiente de operador
- Filas de mensagens não entregues
- Tarefas periódicas

- Use a Amazon CloudWatch para escalabilidade automática em níveis de ambiente de trabalho
- Configurar ambientes de operador

Daemon SQS do ambiente de operador

Os ambientes de operador executam um processo de daemon fornecido pelo Elastic Beanstalk. Esse daemon é atualizado regularmente para adicionar recursos e corrigir erros. Para obter a versão mais recente do daemon, atualize para a última versão da Plataforma.

Quando a aplicação no ambiente de operador retorna uma resposta `200 OK` para confirmar que ele recebeu e processou a solicitação com êxito, o daemon envia uma chamada `DeleteMessage` à fila do Amazon SQS para excluir a mensagem da fila. Se a aplicação retornar qualquer resposta diferente de `200 OK`, o Elastic Beanstalk aguardará para recolocar a mensagem na fila após o período `ErrorVisibilityTimeout` configurado. Se não houver nenhuma resposta, o Elastic Beanstalk aguardará para recolocar a mensagem na fila após o período `InactivityTimeout` para que a mensagem seja disponibilizada para outra tentativa de processamento.

> **Note**
>
> As propriedades das filas do Amazon SQS (pedido de mensagens, at-least-once entrega e amostragem de mensagens) podem afetar a forma como você projeta uma aplicação web para um ambiente de trabalho. Para obter mais informações, consulte Propriedades das filas distribuídas no Guia do desenvolvedor do Amazon Simple Queue Service.

O Amazon SQS exclui automaticamente as mensagens que estão na fila há mais tempo do que o `RetentionPeriod` configurado.

O daemon define os seguintes cabeçalhos HTTP.

Cabeçalhos HTTP

Nome	Valor
`User-Agent`	`aws-sqsd` `aws-sqsd/1.1 1`

Cabeçalhos HTTP	
`X-Aws-Sqsd-Msgid`	ID de mensagens do SQS, usado para detectar enxurradas de mensagens (um número excepcionalmente alto de novas mensagens).
`X-Aws-Sqsd-Queue`	Nome da fila do SQS.
`X-Aws-Sqsd-First-Received-At`	Horário UTC, no formato ISO 8601, quando a mensagem foi recebida pela primeira vez.
`X-Aws-Sqsd-Receive-Count`	Contagem de recebimento de mensagens do SQS.
`X-Aws-Sqsd-Attr-` *`message-attribute-name`*	Atributos de mensagem personalizados concedidos à mensagem que está sendo processada. O `message-attribute-name` é o nome do atributo de mensagem real. Todos os atributos de mensagem de string e número são adicionados ao cabeçalho. Os atributos binários são descartados e não são incluídos no cabeçalho.
`Content-Type`	Configuração de tipo Mime. Por padrão, `application/json` .

Filas de mensagens não entregues

Os ambientes de operador do Elastic Beanstalk são compatíveis com filas de mensagens mortas do Amazon Simple Queue Service (Amazon SQS). Dead letter queue (fila de mensagens mortas) é uma fila onde outras filas (origem) podem enviar mensagens que, por algum motivo, não puderam ser processadas com êxito. Um benefício importante de usar filas de mensagens mortas é a capacidade de segregar e isolar as mensagens que não foram processadas com êxito. Em seguida, você pode analisar todas as mensagens enviadas para a fila de mensagens mortas para tentar determinar por que elas não foram processadas com êxito.

Se você especificar uma fila do Amazon SQS gerada automaticamente no momento em que criar o nível de ambiente de operador, uma fila de mensagens mortas será habilitada por padrão para um

ambiente de operador. Se você escolher uma fila do SQS existente para seu ambiente de operador, deve usar o SQS para configurar uma fila de mensagens mortas de forma independente. Para obter informações sobre como usar o SQS para configurar uma fila de mensagens mortas, consulte Usar fila de mensagens mortas do Amazon SQS.

Você não pode desabilitar filas de mensagens mortas. As mensagens que não puderem ser entregues serão eventualmente enviadas a uma fila de mensagens mortas. No entanto, você pode efetivamente desabilitar esse recurso definindo a opção `MaxRetries` como o valor máximo válido de 100.

Se uma fila de mensagens mortas não estiver configurada para a fila do Amazon SQS do ambiente de operador, o Amazon SQS manterá as mensagens na fila até que o período de retenção expire. Para obter detalhes sobre como configurar o período de retenção, consulte the section called "Configurar ambientes de operador".

> **Note**
>
> A opção `MaxRetries` do Elastic Beanstalk é equivalente à opção `MaxReceiveCount` do SQS. Se o ambiente de operador não usar uma fila do SQS gerada automaticamente, use a opção `MaxReceiveCount` no SQS para desabilitar efetivamente a fila de mensagens mortas. Para obter mais informações, consulte Usar filas de mensagens mortas do Amazon SQS.

Para obter mais informações sobre o ciclo de vida de uma mensagem do SQS, acesse Ciclo de vida de mensagens.

Tarefas periódicas

Você pode definir tarefas periódicas em um arquivo denominado `cron.yaml` em seu pacote de origem para adicionar trabalhos à fila do seu ambiente de operador automaticamente em intervalos regulares.

Por exemplo, o arquivo `cron.yaml` a seguir cria duas tarefas periódicas. O primeiro é executado a cada 12 horas e o segundo é executado às 11 horas UTC todos os dias.

Example cron.yaml

```
version: 1
```

```
cron:
 - name: "backup-job"
   url: "/backup"
   schedule: "0 */12 * * *"
 - name: "audit"
   url: "/audit"
   schedule: "0 23 * * *"
```

O **`name`** deve ser exclusivo para cada tarefa. O URL é o caminho para o qual a solicitação POST é enviada para acionar o trabalho. A programação é uma expressão CRON que determina quando a tarefa é executada.

Quando uma tarefa é executada, o daemon publica uma mensagem na fila do SQS do ambiente com um cabeçalho indicando o trabalho que precisa ser realizado. Qualquer instância no ambiente pode selecionar a mensagem e processar o trabalho.

> **Note**
>
> Se você configurar o ambiente de operador com uma fila do SQS existente e escolher uma fila FIFO do Amazon SQS, as tarefas periódicas não serão compatíveis.

O Elastic Beanstalk usa a eleição de líder para determinar qual instância no ambiente de operador coloca a tarefa periódica em fila. Cada instância tenta se tornar líder gravando em uma tabela do Amazon DynamoDB. A primeira instância que for bem-sucedida será líder e deverá continuar gravando na tabela para manter o status de líder. Se o líder ficar fora de serviço, outra instância rapidamente assumirá seu lugar.

Para tarefas periódicas, o daemon do operador define os seguintes cabeçalhos adicionais.

Cabeçalhos HTTP

Nome	Valor
`X-Aws-Sqsd-Taskname`	Para tarefas periódicas, o nome da tarefa a ser executada.
`X-Aws-Sqsd-Scheduled-At`	Horário programado para a tarefa periódica

Cabeçalhos HTTP	
`X-Aws-Sqsd-Sender-Id`	AWS número da conta do remetente da mensagem

Use a Amazon CloudWatch para escalabilidade automática em níveis de ambiente de trabalho

Juntos, o Amazon EC2 Auto Scaling e o Amazon CloudWatch monitoram a utilização da CPU das instâncias em execução no ambiente de trabalho. A maneira como você configura o limite de escalação automática para a capacidade da CPU determina quantas instâncias o grupo de Auto Scaling executa para gerenciar adequadamente o throughput de mensagens na fila do Amazon SQS. Cada EC2 instância publica suas métricas de utilização da CPU em. CloudWatch O Amazon EC2 Auto Scaling recupera o uso médio CloudWatch da CPU em todas as instâncias no ambiente de trabalho. Você configura o limite superior e inferior e quantas instâncias devem ser adicionadas ou encerradas de acordo com a capacidade da CPU. Quando o Amazon EC2 Auto Scaling detecta que você atingiu o limite superior especificado na capacidade da CPU, o Elastic Beanstalk cria novas instâncias no ambiente de trabalho. As instâncias são excluídas quando a carga da CPU volta a ficar abaixo do limite.

> **Note**
>
> As mensagens que não foram processadas no momento em que uma instância foi encerrada são retornadas para a fila na qual elas podem ser processadas por outro daemon em uma instância que ainda está em execução.

Você também pode definir outros CloudWatch alarmes, conforme necessário, usando o console do Elastic Beanstalk, a CLI ou o arquivo de opções. Para obter mais informações, consulte Usando o Elastic Beanstalk com a Amazon CloudWatch e Criar um grupo de Auto Scaling com políticas de escalabilidade em etapas.

Configurar ambientes de operador

É possível gerenciar a configuração de um ambiente de operador editando a configuração Worker Configuration (Configuração do operador) na página Configuration (Configuração) no console de gerenciamento do 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**
>
> Você pode configurar o caminho do URL para publicar mensagens na fila de operadores, mas não pode configurar a porta IP. O Elastic Beanstalk sempre publica mensagens da fila do operador na porta 80. O aplicativo do ambiente de operador ou seu proxy deve ouvir na porta 80.

Para configurar o daemon do operador

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Worker (Operador), escolha Edit (Editar).

A página de configuração Modify worker (Modificar operador) tem as seguintes opções.

Na seção Queue (Fila):

- Worker queue (Fila do operador): especifique a fila do Amazon SQS de leitura do daemon. Se tiver uma, você poderá escolher uma fila existente. Se você escolher Autogenerated queue (Fila gerada automaticamente), o Elastic Beanstalk criará uma nova fila do Amazon SQS e um Worker queue URL (URL da fila do operador)correspondente.

  > **Note**
  >
  > Quando você escolhe Autogenerated queue (Fila gerada automaticamente), a fila criada pelo Elastic Beanstalk é uma fila padrão do Amazon SQS. Ao escolher uma fila existente, você pode fornecer uma fila padrão ou uma fila FIFO do Amazon SQS. Lembre-se de que, se você fornecer uma fila FIFO, as tarefas periódicas não serão compatíveis.

- Worker queue URL ((URL da fila do operador)): se você escolher uma Worker queue (Fila do operador) existente, essa configuração exibirá o URL associado a essa fila do Amazon SQS.

Na seção Messages (Mensagens):

- HTTP path (Caminho HTTP): especifique o caminho relativo para a aplicação que receberá os dados da fila do Amazon SQS. Os dados são inseridos no corpo de uma mensagem HTTP POST. O valor padrão é `/`.
- MIME type (Tipo MIME): indica o tipo MIME usado pela mensagem HTTP POST. O valor padrão é `application/json`. No entanto, qualquer valor é válido, pois você pode criar e, em seguida, especificar seu próprio tipo MIME.
- Conexões HTTP — Especifique o número máximo de conexões simultâneas que o daemon pode fazer com qualquer aplicativo dentro de uma instância da Amazon. EC2 O padrão é **`50`**. Você pode especificar de **`1`** a **`100`**.
- Visibility timeout (Tempo limite de visibilidade): indique por quanto tempo (em segundos) uma mensagem recebida da fila do Amazon SQS fica bloqueada para processamento. Após o término do tempo configurado, a mensagem ficará novamente visível na fila para outro daemon ler. Escolha um valor superior ao que seu aplicativo precisa para processar mensagens, até **`43200`** segundos.
- Error visibility timeout (Tempo limite de visibilidade de erros): indique quanto tempo (em segundos) o Elastic Beanstalk deve aguardar para retornar uma mensagem à fila do Amazon SQS após uma falha com erro explícito ao tentar processá-la. Você pode especificar de **`0`** a **`43200`** segundos.

Na seção Advanced options:

- Max retries (Máximo de tentativas): especifique o número máximo de vezes que o Elastic Beanstalk tenta enviar a mensagem à fila do Amazon SQS antes de movê-la para a fila de mensagens mortas. O valor padrão é **`10`**. Você pode especificar de **`1`** a **`100`**.

  > **Note**
  >
  > A opção Max retries (Máximo de novas tentativas) só é aplicável a filas do Amazon SQS configuradas com uma fila de mensagens mortas. Para qualquer fila do Amazon SQS que não esteja configurada com uma fila de mensagens mortas, o Amazon SQS retém as mensagens na fila e as processa até a expiração do período especificado pela opção Retention period (Período de retenção).

- Connection timeout (Tempo limite de conexão): indique por quanto tempo (em segundos) aguardar por conexões bem-sucedidas com uma aplicação. O valor padrão é **`5`**. Você pode especificar de **`1`** a **`60`** segundos.

- Inactivity timeout (Tempo limite de inatividade): indique por quanto tempo (em segundos) aguardar por uma resposta para uma conexão existente com uma aplicação. O valor padrão é **180**. Você pode especificar de **1** a **36000** segundos.
- Retention period (Período de retenção): indique por quanto tempo (em segundos) uma mensagem é válida e processada ativamente. O valor padrão é **345600**. Você pode especificar de **60** a **1209600** segundos.

Se você usa uma fila do Amazon SQS existente, as configurações que você define ao criar um ambiente de operador podem entrar em conflito com as configurações definidas diretamente no Amazon SQS. Por exemplo, se você configurar um ambiente de operador com um valor de `RetentionPeriod` maior que o valor de `MessageRetentionPeriod` definido no Amazon SQS, este excluirá a mensagem quando ela exceder o `MessageRetentionPeriod`.

Por outro lado, se o valor `RetentionPeriod` definido na configuração do ambiente de operador for menor do que o valor de `MessageRetentionPeriod` definido no Amazon SQS, o daemon excluirá a mensagem antes de o Amazon SQS poder excluí-la. Em `VisibilityTimeout`, o valor definido para o daemon nas configurações do ambiente de operador substitui a configuração `VisibilityTimeout` do Amazon SQS. Certifique-se de que as mensagens sejam excluídas adequadamente comparando suas configurações do Elastic Beanstalk às configurações do Amazon SQS.

Criar links entre ambientes do Elastic Beanstalk

À medida que seu aplicativo aumenta de tamanho e complexidade, você pode dividi-lo em componentes com diferentes ciclos de vida operacionais e de desenvolvimento. Ao executar serviços menores que interagem com os outros por uma interface bem definida, as equipes podem trabalhar de forma independente, e implantações podem ser de menor risco. O AWS Elastic Beanstalk permite vincular seus ambientes para compartilhar informações entre componentes que dependem um do outro.

> **Note**
>
> Atualmente, o Elastic Beanstalk é compatível com links de ambiente para todas as plataformas, exceto Docker de vários contêineres.

Com os links de ambiente, você pode especificar as conexões entre os ambientes de componentes do aplicativo como referências nomeadas. Quando um ambiente que define um link é criado, o Elastic Beanstalk define uma variável de ambiente com o mesmo nome do link. O valor da variável é o endpoint que você pode usar para se conectar a outro componente, que pode ser um servidor Web ou ambiente de operador.

Por exemplo, se o aplicativo é composto de um front-end que coleta endereços de e-mail e um operador que envia e-mail de boas-vindas para os endereços coletados pelo front-end, você pode criar um link para o operador do front-end e o front-end descobrir automaticamente o endpoint (URL da fila) para o operador.

Defina links para outros ambientes em um manifesto de ambiente, um arquivo em formato YAML nomeado `env.yaml` na raiz da origem do aplicativo. O seguinte manifesto define um link para um ambiente denominado operador:

`~/workspace/my-app/frontend/env.yaml`

```
AWSConfigurationTemplateVersion: 1.1.0.0
EnvironmentLinks:
  "WORKERQUEUE": "worker"
```

Quando você cria um ambiente com uma versão da aplicação que inclui o manifesto de ambiente acima, o Elastic Beanstalk procura um ambiente chamado `worker` que pertence à mesma aplicação. Se esse ambiente existir, o Elastic Beanstalk criará uma propriedade de ambiente chamada `WORKERQUEUE`. O valor de `WORKERQUEUE` é o URL da fila do Amazon SQS. O aplicativo de front-end pode ler essa propriedade da mesma maneira que uma variável de ambiente. Para mais detalhes, consulte Manifesto de ambiente (`env.yaml`).

Para usar links de ambiente, adicione um manifesto de ambiente à fonte do aplicativo e carregue-o com a CLI do EB AWS CLI ou um SDK. Se você usa o AWS CLI ou um SDK, defina o `process` sinalizador ao chamar`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"
```

Essa opção instrui o Elastic Beanstalk a validar o manifesto de ambiente e os arquivos de configuração no pacote de origem quando você cria a versão da aplicação. A EB CLI define esse sinalizador automaticamente quando você tem um manifesto de ambiente no diretório do projeto.

Crie seus ambientes normalmente usando qualquer cliente. Quando for necessário encerrar ambientes, primeiro faça isso com o vínculo. Se um ambiente estiver vinculado a qualquer outro ambiente, o Elastic Beanstalk impedirá o encerramento do ambiente vinculado. Para substituir essa proteção, use o sinalizador `ForceTerminate`. Esse parâmetro está disponível na AWS CLI como `--force-terminate`:

```
$ aws elasticbeanstalk terminate-environment --force-terminate --environment-name
 worker
```

Recuperando seu ambiente do Elastic Beanstalk de um estado inválido

Este tópico fornece algumas informações básicas e recursos que explicam como solucionar problemas em um ambiente do Elastic Beanstalk em um estado inválido.

Resolvendo o erro

As operações padrão em um ambiente em um estado inválido não serão concluídas com êxito. A operação com falha retornará um erro que inclui o seguinte texto:

```
The stack stack_id associated with environment environment-ID is in stack-status state.
```

Para solucionar esse erro, consulte o artigo do Knowledge Center Por que meu ambiente do Elastic Beanstalk está no estado inválido? .

> **Note**
>
> Antes de 16 de dezembro de 2024, a operação com falha retornava o seguinte erro: Nesse caso, você precisava entrar `Environment is in an invalid state for this operation. Must be ready.` em contato com o AWS Support para redefinir o status do ambiente depois de concluir as ações corretivas.
> Atualmente, você ainda deve resolver os problemas de pilha seguindo as instruções no artigo do Centro de Conhecimento referenciado. No entanto, depois de concluir com êxito as ações corretivas, o Elastic Beanstalk atualiza automaticamente o status do ambiente de inválido para disponível, e você pode retomar as operações padrão em seu ambiente sem mais demoras.

Por que o erro ocorre

Quando você implanta um aplicativo no Elastic Beanstalk, o serviço cria uma pilha subjacente. AWS CloudFormation O Elastic Beanstalk AWS CloudFormation chama o serviço para lançar os recursos em seu ambiente e propagar as alterações de configuração.

Se o Elastic Beanstalk realizar uma operação em um ambiente sem ter acesso a um recurso necessário, a pilha subjacente do ambiente poderá entrar em um estado CloudFormation de falha. Outros problemas também podem levar a esse estado, embora os problemas de permissão sejam a principal causa. Como resultado do estado de falha da pilha, AWS CloudFormation impede que as solicitações de operação do Elastic Beanstalk realizem mais atualizações da pilha, causando a falha das operações do Elastic Beanstalk, como e. UpdateEnvironment RetrieveEnvironmentInfo

Nesse ponto, você deve primeiro corrigir a causa raiz do problema subjacente para remediar a CloudFormation pilha. Em seguida, o serviço Elastic Beanstalk CloudFormation detecta a alteração no status da pilha e segue em frente para redefinir seu ambiente para um status disponível. Nesse ponto, outras operações podem ser concluídas com êxito.

Os problemas de permissão geralmente causam esse efeito na CloudFormation pilha e no ambiente do Elastic Beanstalk out-of-band, embora as mudanças também possam causar problemas.

> **Important**
>
> Para evitar interrupções em seu ambiente, é altamente recomendável que você inicie operações apenas para gerenciar e configurar seu ambiente a partir do serviço Elastic Beanstalk. A modificação de recursos usando o console, os comandos da CLI ou o SDK de um serviço diferente do Elastic Beanstalk é uma alteração que causa desvio de recursos. out-of-band O desvio de recursos afeta o status da CloudFormation pilha, o que, por sua vez, faz com que o ambiente do Elastic Beanstalk entre em um estado inválido.
> Para obter mais informações sobre desvio de recursos, consulte O que é desvio? no Guia do AWS CloudFormation usuário.

Configurar ambientes do Elastic Beanstalk

Este tópico se concentra nas opções de configuração disponíveis no console do Elastic Beanstalk. AWS Elastic Beanstalk fornece uma ampla variedade de opções para personalizar os recursos em seu ambiente, junto com o comportamento e as configurações da plataforma do Elastic Beanstalk.

Os tópicos a seguir mostram como configurar seu ambiente no console. Eles também descrevem os namespaces subjacentes que correspondem às opções do console a serem usadas com arquivos de configuração ou com as opções de configuração da API. Para saber mais sobre métodos de configuração avançados, consulte Configurar ambientes (avançado).

Tópicos

- Recursos provisionados
- Configuração do ambiente usando o console do Elastic Beanstalk
- As EC2 instâncias da Amazon para seu ambiente Elastic Beanstalk
- Auto Scaling de suas instâncias do ambiente do Elastic Beanstalk
- Balanceador de carga do ambiente do Elastic Beanstalk
- Adição de um banco de dados ao seu ambiente do Elastic Beanstalk
- A segurança AWS Elastic Beanstalk do seu ambiente
- Marcar recursos em seus ambientes do Elastic Beanstalk
- Variáveis de ambiente e outras configurações de software
- Notificações do ambiente do Elastic Beanstalk com o Amazon SNS
- Configurar a Amazon Virtual Private Cloud (Amazon VPC) com o Elastic Beanstalk
- Nome de domínio do ambiente do Elastic Beanstalk

Recursos provisionados

Quando você cria um ambiente de servidor web, o Elastic Beanstalk cria vários recursos para dar suporte à operação da sua aplicação. Este capítulo descreve como personalizar esses recursos para o ambiente Elastic Beanstalk.

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de

solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Configuração do ambiente usando o console do Elastic Beanstalk

Este tópico descreve as opções de configuração disponíveis no console do Elastic Beanstalk e explica como navegar pelas páginas de configuração.

Como visualizar um resumo da configuração de seu ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).

Página de configuração

A página Configuration overview (Visão geral de configuração) mostra um conjunto de categorias de configuração. Cada categoria de configuração agrupa um conjunto de opções relacionadas.

Acesso ao serviço

As opções nessa categoria selecionam a função de serviço e o perfil da EC2 instância que o Elastic Beanstalk usa para gerenciar seu ambiente. Opcionalmente, escolha um par de EC2 chaves para fazer login com segurança em suas instâncias. EC2

Redes e banco de dados

As opções nessa categoria definem configurações de VPC e sub-redes para as EC2 instâncias e o balanceador de carga do ambiente. Elas também oferecem a opção de configurar um banco de dados do Amazon RDS integrado ao ambiente.

Tráfego e dimensionamento de instâncias

Essas opções personalizam a capacidade, a escalabilidade e o balanceamento de carga das instâncias do EC2 ambiente. Você também pode configurar o Elastic Load Balancing para capturar logs com informações detalhadas sobre as solicitações enviadas ao balanceador de carga.

As seguintes opções para suas EC2 instâncias também estão disponíveis para configuração:

- Tipo, tamanho, taxa de input/output operação (IOPS) e taxa de transferência do volume raiz.
- Habilitar o serviço de metadados de instância (IMDS).
- Seleção de grupos de EC2 segurança para controlar o tráfego da instância.
- CloudWatch intervalo de monitoramento de métricas.
- Intervalo de tempo para log de métricas.

Atualizações, monitoramento e registro em log

Essa categoria configura as seguintes opções:

- Relatórios de integridade do ambiente, incluindo a opção de selecionar relatórios de integridade aprimorados.
- Atualizações gerenciadas da plataforma que definem quando e como o Elastic Beanstalk implementa alterações no ambiente.
- Ativação do serviço X-Ray para coletar dados sobre o comportamento da sua aplicação a fim de identificar problemas e oportunidades de otimização.
- Opções específicas da plataforma, incluindo o servidor proxy e as propriedades do ambiente do sistema operacional.

Navegar na página de configuração

Selecione Editar em uma categoria de configuração para abrir uma página de configuração associada, onde você pode ver os valores completos das opções e fazer alterações.

Navegação em uma categoria de configuração

Navegue em uma página de categoria de configuração com qualquer uma das seguintes ações:

- Cancelar: volta para a página de visão geral da Configuração sem aplicar as alterações de configuração. Ao escolher Cancel (Cancelar), o console perde as alterações pendentes feitas em qualquer categoria de configuração.

  Você também pode cancelar suas alterações de configuração escolhendo outro item na página de navegação à esquerda, como Eventos ou Logs.
- Continue (Continuar): voltar para a página Configuration Overview (Visão geral da configuração). Em seguida, você pode continuar a fazer alterações ou aplicar as pendentes.

- Aplicar: aplique as alterações feitas em qualquer uma das categorias de configuração ao seu ambiente. Em alguns casos, será solicitado que você confirme uma consequência de uma das suas decisões de configuração.

Navegação na página de visão geral da Configuração

Selecione Editar em uma categoria de configuração para abrir uma página de configuração relacionada, onde você pode ver os valores completos das opções e fazer alterações. Quando terminar de visualizar e modificar as opções, escolha uma das seguintes ações na página de visão geral Configuração:

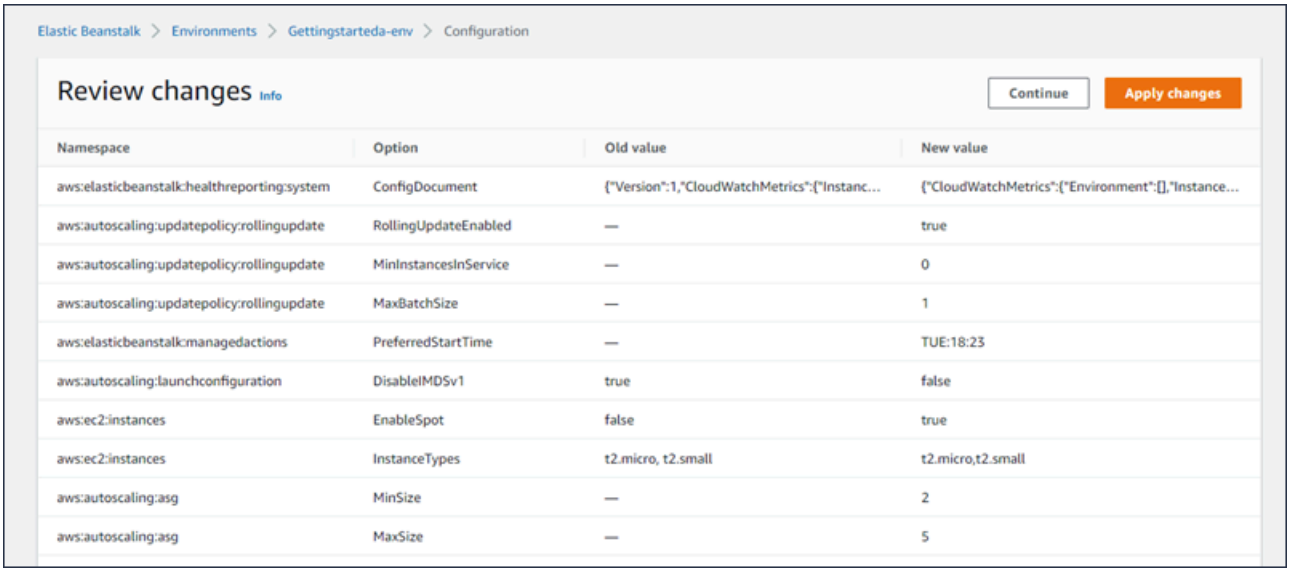
- Cancel (Cancelar): volte ao painel do ambiente sem aplicar suas alterações de configuração. Ao escolher Cancel (Cancelar), o console perde as alterações pendentes feitas em qualquer categoria de configuração.

  Você também pode cancelar suas alterações de configuração escolhendo outro item na página de navegação à esquerda, como Eventos ou Logs.
- Review changes (Revisar alterações): obtenha um resumo de todas as alterações pendentes feitas em qualquer uma das categorias de configuração. Para obter detalhes, consulte Página Revisar alterações.
- Apply changes (Aplicar alterações): aplique no seu ambiente todas as alterações feitas nas categorias de configuração. Em alguns casos, será solicitado que você confirme uma consequência de uma das suas decisões de configuração.

Página Revisar alterações

A página Review Changes (Revisar alterações) exibe uma tabela que mostra todas as alterações pendentes de opções feitas em qualquer uma das categorias de configuração e que não foram aplicadas ainda ao seu ambiente.

As tabelas listam cada opção como uma combinação de Namespace e Option (Opção) com a qual o Elastic Beanstalk a identifica. Para obter detalhes, consulte Opções de configuração.



Quando terminar de revisar suas alterações, você poderá escolher uma das seguintes ações:

- Continue (Continuar): voltar para a página Configuration Overview (Visão geral da configuração). Em seguida, você pode continuar a fazer alterações ou aplicar as pendentes.
- Apply changes (Aplicar alterações): aplique no seu ambiente todas as alterações feitas nas categorias de configuração. Em alguns casos, será solicitado que você confirme uma consequência de uma das suas decisões de configuração.

As EC2 instâncias da Amazon para seu ambiente Elastic Beanstalk

Este tópico explica as EC2 instâncias da Amazon e as opções de configuração que afetam seu ambiente do Elastic Beanstalk.

Quando você cria um ambiente de servidor web, AWS Elastic Beanstalk cria uma ou mais máquinas virtuais Amazon Elastic Compute Cloud (Amazon EC2), conhecidas como Instâncias.

As instâncias em seu ambiente são configuradas para executar aplicativos web na plataforma de sua preferência. Você pode fazer alterações em várias propriedades e comportamentos das instâncias do seu ambiente ao criar o ambiente ou depois que ele já estiver em execução. Ou, você já pode fazer essas alterações modificando o código-fonte que implanta no ambiente. Para obter mais informações, consulte the section called “Opções de configuração”.

> **Note**
>
> O grupo Auto Scaling em seu ambiente gerencia as EC2 instâncias da Amazon que executam seu aplicativo. Quando você faz alterações na configuração descritas neste tópico, a configuração de inicialização também é alterada. A configuração de lançamento é um modelo de EC2 lançamento da Amazon ou um recurso de configuração de lançamento em grupo do Auto Scaling. Essa alteração requer a substituição de todas as instâncias. Ele também aciona uma atualização contínua ou atualização imutável, dependendo de qual delas está configurada.

EC2 opções de compra por instância

O Elastic Beanstalk oferece suporte a EC2 várias opções de compra de instâncias da Amazon:

- Sob demanda — uma instância sob demanda é um pay-as-you-go recurso — não é necessário nenhum compromisso de longo prazo quando você a usa.
- Reservada: uma instância reservada é um desconto de faturamento pré-comprado aplicado automaticamente às instâncias sob demanda correspondentes no seu ambiente.
- Spot — Uma instância spot é uma EC2 instância não utilizada da Amazon que está disponível por menos do que o preço sob demanda. Você pode ativar e configurar a alocação de Instâncias Spot em seu ambiente. Para obter mais informações, consulte Auto Scaling de suas instâncias do ambiente do Elastic Beanstalk.

Tópicos

- Tipos de EC2 instância da Amazon
- Configurando EC2 instâncias da Amazon usando o console do Elastic Beanstalk
- Gerenciando grupos EC2 de segurança
- Configurando grupos de EC2 segurança e tipos de instância da Amazon usando o AWS CLI
- Configurando EC2 instâncias da Amazon com opções de namespace
- Configuração do IMDS nas instâncias do ambiente Elastic Beanstalk

Tipos de EC2 instância da Amazon

Este tópico explica o termo tipo de instância. Quando você cria um novo ambiente, o Elastic Beanstalk EC2 provisiona instâncias da Amazon com base nos tipos de instância EC2 da Amazon que você escolher. Os tipos de instância que você escolhe determinam o hardware do host que executa suas instâncias. EC2 os tipos de instância podem ser categorizados de acordo com a arquitetura de processador em que cada um se baseia. O Elastic Beanstalk suporta tipos de instância com base nas seguintes AWS arquiteturas de processador: arquitetura Arm Graviton de 64 bits (arm64), arquitetura de 64 bits (x86) e arquitetura de 32 bits (i386). O Elastic Beanstalk seleciona a arquitetura de processador x86 por padrão quando você cria um novo ambiente.

> **Note**
>
> A arquitetura i386 de 32 bits não é mais compatível com a maioria das plataformas do Elastic Beanstalk. Então recomendamos que você escolha os tipos de arquitetura x86 ou arm64. O Elastic Beanstalk fornece Opções de configuração para o tipo de instância do processador i386 no namespace da `aws:ec2:instances`.

Todos os tipos de instância na configuração de um determinado ambiente do Elastic Beanstalk devem ter o mesmo tipo de arquitetura de processador. Suponha que você adicione um novo tipo de instância a um ambiente existente que já tenha um tipo de instância t2.medium, que é baseado na arquitetura x86. Você só pode adicionar outro tipo de instância da mesma arquitetura, como t2.small. Se quiser substituir os tipos de instância existentes pelos de uma arquitetura diferente, você pode fazer isso. Porém, certifique-se de que todos os tipos de instância no comando sejam baseados no mesmo tipo de arquitetura.

O Elastic Beanstalk regularmente adiciona suporte para novos tipos de instância compatíveis depois que a Amazon os introduz. EC2 Para obter informações sobre os tipos de instância disponíveis, consulte Tipos de instância no Guia EC2 do usuário da Amazon.

> **Note**
>
> O Elastic Beanstalk agora oferece suporte ao Graviton em todas as plataformas mais recentes do Amazon Linux 2 em todas as regiões suportadas pelo Graviton. AWS Para obter mais informações sobre como criar um ambiente do Elastic Beanstalk com tipos de instâncias baseados em arm64, consulte Configurando EC2 instâncias da Amazon usando o console do Elastic Beanstalk.

Crie novos ambientes que executam EC2 instâncias da Amazon na arquitetura arm64 e migre seus aplicativos existentes para eles com as opções de implantação no Elastic Beanstalk.
Para saber mais sobre os processadores baseados em Graviton arm64, consulte estes AWS recursos:

- Benefícios — O AWS processador Graviton
- Introdução e outros tópicos, como considerações específicas do idioma — Artigo sobre como começar a usar o Graviton AWS GitHub

Configurando EC2 instâncias da Amazon usando o console do Elastic Beanstalk

Você pode criar ou modificar a configuração de instância da Amazon do seu ambiente do Elastic Beanstalk no console EC2 do Elastic Beanstalk.

> **Note**
>
> Embora o console do Elastic Beanstalk não forneça a opção para alterar a arquitetura do processador de um ambiente existente, é possível fazer isso com o AWS CLI. Para obter um exemplo de comandos, consulte Configurando grupos de EC2 segurança e tipos de instância da Amazon usando o AWS CLI.

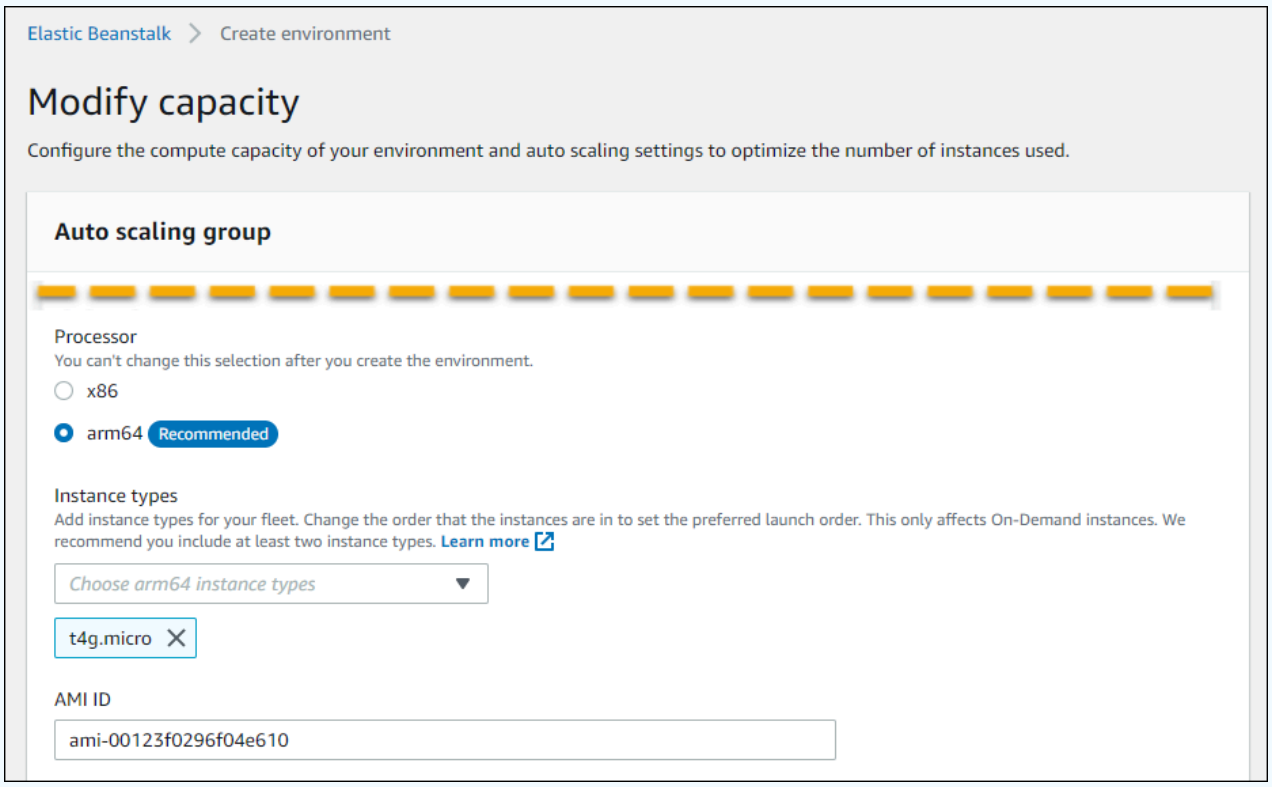
Para configurar EC2 instâncias da Amazon no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Ambientes.
3. Escolha Create a new environment (Criar um ambiente) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Apply (Aplicar). Para obter descrições de configuração, consulte a seção the section called “Configurações de categoria de instâncias” nesta página.

6. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Continue (Continuar). Para obter descrições de configuração, consulte a seção the section called “Configurações da categoria de capacidade” nesta página.

   Selecione a arquitetura do processador.

   Role para baixo até Processador para selecionar uma arquitetura de processador para suas EC2 instâncias. O console lista as arquiteturas de processador compatíveis com a plataforma que você escolheu anteriormente no painel Criar ambiente.
   Se não encontrar a arquitetura do processador que você precisa, volte à lista de categorias de configuração da lista para selecionar uma plataforma compatível com ela. No painel Modificar capacidade, escolha Cancelar. Em seguida, escolha Alterar a versão da plataforma para escolher novas configurações de plataforma. Em seguida, na categoria de configuração Capacidade, escolha Editar para ver as opções de arquitetura de processador novamente.



7. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
8. Selecione Criar ambiente.

Para configurar as EC2 instâncias da Amazon de um ambiente em execução no console do Elastic Beanstalk

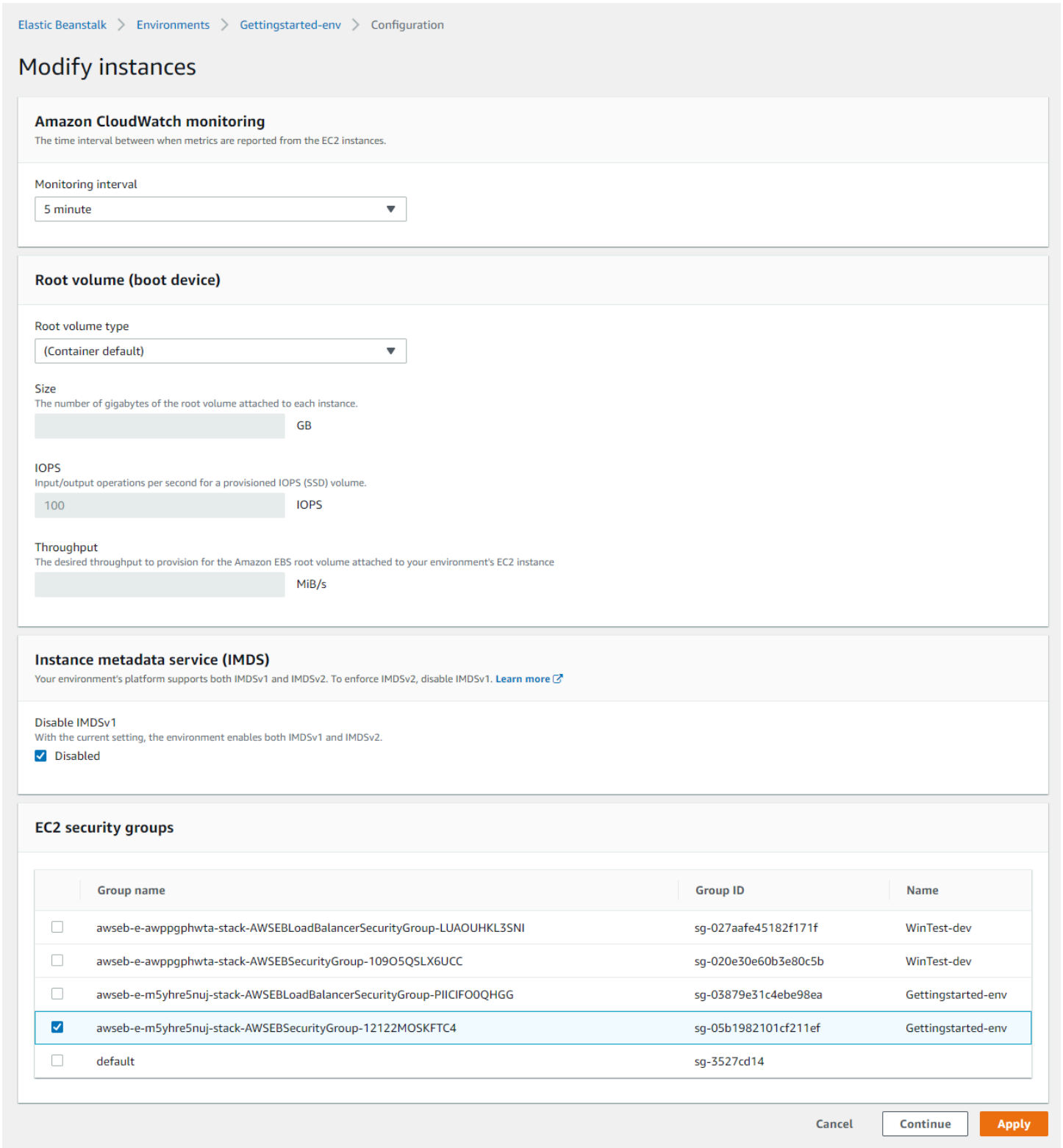
1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Apply (Aplicar). Para obter descrições de configuração, consulte a seção the section called “Configurações de categoria de instâncias” nesta página.
5. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar). Faça alterações nas configurações dessa categoria e escolha Continue (Continuar). Para obter descrições de configuração, consulte a seção the section called “Configurações da categoria de capacidade” nesta página.

Configurações de categoria de instâncias

As seguintes configurações relacionadas às EC2 instâncias da Amazon estão disponíveis na categoria Configuração de instâncias.

Opções

- Intervalo de monitoramento
- Volume raiz (dispositivo de inicialização)
- Serviço de metadados da instância
- EC2 grupos de segurança


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

Intervalo de monitoramento

Por padrão, as instâncias em seu ambiente publicam métricas básicas de saúde CloudWatch na Amazon em intervalos de cinco minutos, sem custo adicional.

Para obter relatórios mais detalhados, você pode definir o intervalo de monitoramento como 1 minuto para aumentar a frequência com que os recursos em seu ambiente publicam métricas básicas de CloudWatch saúde. CloudWatch as taxas de serviço se aplicam a métricas de intervalo de um minuto. Para obter mais informações, consulte Amazon CloudWatch.

Volume raiz (dispositivo de inicialização)

Cada instância do ambiente é configurado com um volume raiz. O volume raiz é o dispositivo de blocos do Amazon EBS anexado à instância para armazenar o sistema operacional, as bibliotecas, os scripts e o código-fonte da aplicação. Por padrão, todas as plataformas usam dispositivos de blocos SSD de uso geral para armazenamento.

Você pode modificar o Root volume type (Tipo de volume raiz) para usar armazenamento magnético ou tipos de volume SSD de IOPS provisionadas e, se necessário, aumentar o tamanho do volume. Para os volumes de IOPS provisionadas, selecione também o número de IOPS a provisionar. O throughput só é aplicável a tipos de volume gp3 SSD. Você pode inserir o throughput desejado para provisionamento. Ela pode variar entre 125 a 1000 mebibytes por segundo (MiB/s). Selecione o tipo de volume que atenda aos seus requisitos de performance e preço.

> **Important**
>
> A configuração da `RootVolumeType` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Para obter mais informações, consulte Tipos de volume do Amazon EBS no Guia do EC2 usuário da Amazon e nos detalhes do produto Amazon EBS.

Serviço de metadados da instância

O serviço de metadados da instância (IMDS) é um componente na instância que o código na instância usa para acessar metadados da instância com segurança. O código pode acessar metadados de instância em uma instância em execução usando um dos seguintes métodos: Eles são Instance Metadata Service Version 1 (IMDSv1) ou Instance Metadata Service Version 2 ()IMDSv2. IMDSv2 é mais seguro. Desative IMDSv1 para impor IMDSv2. Para obter mais informações, consulte the section called “IMDS”.

> **Note**
>
> A seção IMDS nesta página de configuração aparece somente para versões de plataforma compatíveis IMDSv2.

EC2 grupos de segurança

Os grupos de segurança vinculados às suas instâncias determinam qual tráfego tem permissão para chegar e sair das instâncias.

O grupo EC2 de segurança padrão criado pelo Elastic Beanstalk permite todo o tráfego de entrada da Internet ou dos balanceadores de carga nas portas padrão para HTTP (80) e SSH (22). Você também pode definir seus próprios grupos de segurança personalizados para designar regras de firewall para as EC2 instâncias. Os grupos de segurança podem permitir o tráfego em outras portas ou de outras fontes. Por exemplo, você pode criar um grupo de segurança para acessar SSH que permita o tráfego de entrada na porta 22 a partir de um intervalo restrito de endereços IP. Ou, para maior segurança, você pode criar um que permita o tráfego de um Bastion Host que somente você pode acessar.

Você pode optar por excluir seu ambiente do grupo de EC2 segurança padrão definindo a `DisableDefaultEC2SecurityGroup` opção no aws:autoscaling:launchconfiguration namespace como. `true` Essa opção não está disponível no console, mas você pode configurá-la com AWS CLI o. Para obter mais informações, consulte Gerenciando grupos EC2 de segurança.

Para obter mais informações sobre os grupos EC2 de segurança da Amazon, consulte Grupos EC2 de segurança da Amazon no Guia EC2 do usuário da Amazon.

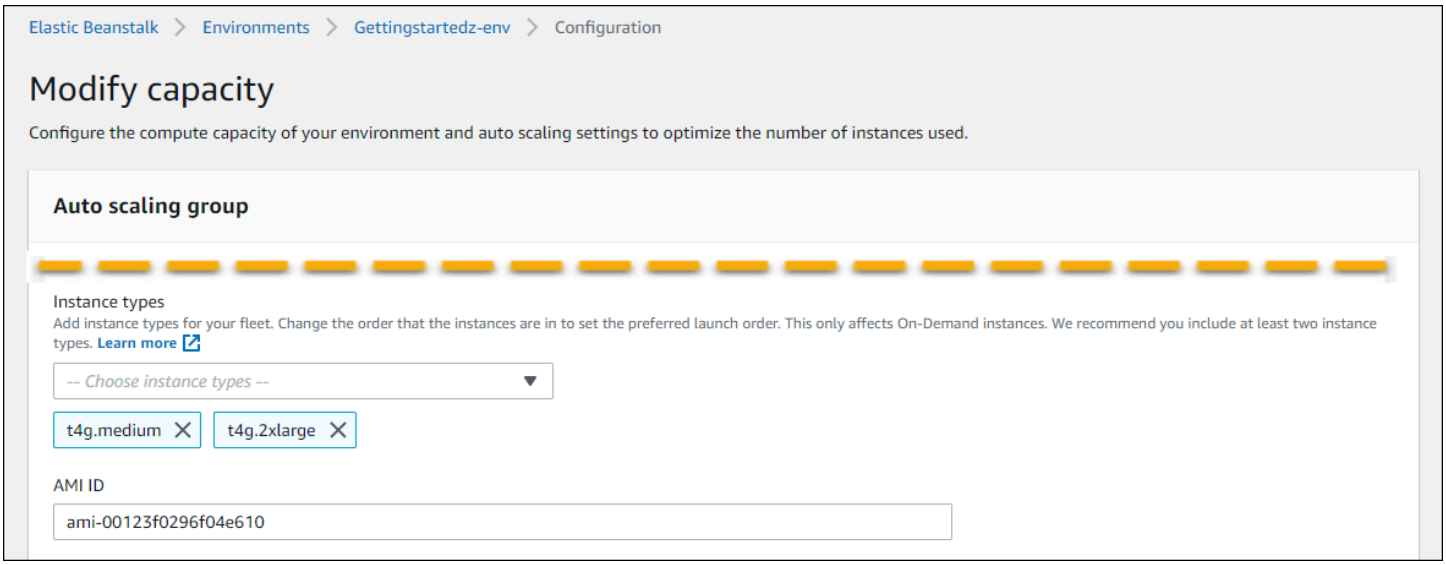
> **Note**
>
> Para permitir tráfego entre as instâncias do ambiente A e as instâncias do ambiente B, você pode adicionar uma regra ao grupo de segurança que o Elastic Beanstalk anexou ao ambiente B. Depois, você pode especificar o grupo de segurança que o Elastic Beanstalk anexou ao ambiente A. Isso permite tráfego de entrada e tráfego de saída nas instâncias do ambiente A. No entanto, isso cria uma dependência entre os dois grupos de segurança. Se mais tarde você tentar encerrar o ambiente A, o Elastic Beanstalk não poderá excluir o grupo de segurança do ambiente, pois o grupo de segurança do ambiente B depende dele.
> Por isso, recomendamos primeiro criar um grupo de segurança separado. Depois, anexe-o ao ambiente A e especifique-o em uma regra do grupo de segurança do ambiente B.

Configurações da categoria de capacidade

As seguintes configurações relacionadas às EC2 instâncias da Amazon estão disponíveis na categoria Configuração de capacidade.

Opções

- Tipos de instância
- ID de AMI



Tipos de instância

A configuração de tipos de instância determina o tipo de EC2 instância da Amazon que é lançada para executar seu aplicativo. Esta página de configuração mostra uma lista de Tipos de instância. Você pode selecionar um ou mais tipo de instância. O console do Elastic Beanstalk exibe apenas os tipos de instância com base na arquitetura do processador configurada para seu ambiente. Portanto, você só pode adicionar tipos de instância da mesma arquitetura de processador.

> **Note**
>
> Embora o console do Elastic Beanstalk não forneça a opção para alterar a arquitetura do processador de um ambiente existente, é possível fazer isso com o AWS CLI. Para obter um exemplo de comandos, consulte Configurando grupos de EC2 segurança e tipos de instância da Amazon usando o AWS CLI.

Escolha uma instância com capacidade suficiente para executar sua aplicação sob carga, mas não com tanta capacidade que ela fique ociosa a maior parte do tempo. Para fins de desenvolvimento, a família t2 de instâncias fornece poder de processamento moderado com capacidade de intermitência por curtos períodos de tempo. Para aplicações de alta disponibilidade e grande escala, use um grupo de instâncias para garantir que a capacidade não seja afetada significativamente se uma instância parar de funcionar. Comece com um tipo de instância que permita executar cinco instâncias com carga moderada durante o horário normal. Se alguma instância falhar, as demais poderão absorver o restante do tráfego. O buffer de capacidade também permite tempo para o ambiente expandir conforme o tráfego começa a subir durante horários de pico.

Para obter mais informações sobre famílias e tipos de EC2 instâncias da Amazon, consulte Tipos de instância no Guia EC2 do usuário da Amazon. Para determinar quais tipos de instância atendem aos seus requisitos e às regiões suportadas, consulte Tipos de instância disponíveis no Guia EC2 do usuário da Amazon.

ID de AMI

A Amazon Machine Image (AMI) é a imagem de máquina Amazon Linux ou Windows Server que o Elastic Beanstalk usa para EC2 iniciar instâncias da Amazon em seu ambiente. O Elastic Beanstalk fornece imagens de máquinas que contêm as ferramentas e os recursos necessários para executar a aplicação.

O Elastic Beanstalk seleciona uma AMI padrão para o ambiente com base na região, versão da plataforma e arquitetura do processador escolhida. Se você tiver criado uma AMI personalizada, substitua o ID padrão da AMI pelo seu próprio padrão personalizado.

Gerenciando grupos EC2 de segurança

Quando o Elastic Beanstalk cria um ambiente, ele atribui um grupo de segurança padrão EC2 às instâncias que são executadas com ele. Os grupos de segurança vinculados às suas instâncias determinam qual tráfego tem permissão para chegar e sair das instâncias.

O grupo EC2 de segurança padrão criado pelo Elastic Beanstalk permite todo o tráfego de entrada da Internet ou dos balanceadores de carga nas portas padrão para HTTP (80) e SSH (22). Você também pode definir seus próprios grupos de segurança personalizados para designar regras de firewall para as EC2 instâncias. Os grupos de segurança podem permitir o tráfego em outras portas ou de outras fontes. Por exemplo, você pode criar um grupo de segurança para acessar SSH que permita o tráfego de entrada na porta 22 a partir de um intervalo restrito de endereços IP. Ou, para maior segurança, você pode criar um que permita o tráfego de um Bastion Host que somente você pode acessar.

Você pode optar por excluir seu ambiente do grupo de EC2 segurança padrão definindo a `DisableDefaultEC2SecurityGroup` opção no aws:autoscaling:launchconfiguration namespace como. `true` Use os arquivos de configuração AWS CLIou para aplicar essa opção ao seu ambiente e anexar grupos de segurança personalizados às EC2 instâncias.

Gerenciamento de grupos EC2 de segurança em ambientes de várias instâncias

Se você criar um grupo de EC2 segurança personalizado em um ambiente de várias instâncias, também deverá considerar como os balanceadores de carga e as regras de tráfego de entrada mantêm suas instâncias seguras e acessíveis.

O tráfego de entrada para um ambiente com várias EC2 instâncias é gerenciado pelo balanceador de carga, que direciona o tráfego de entrada entre todas as instâncias. EC2 Quando o Elastic Beanstalk cria EC2 um grupo de segurança padrão, ele também define regras de entrada que permitem o tráfego de entrada do balanceador de carga. Sem essa regra de entrada no grupo de segurança, o tráfego de entrada não poderá entrar nas instâncias. Essa condição basicamente bloquearia as instâncias de solicitações externas.

Se você desabilitar o grupo de EC2 segurança padrão para um ambiente com balanceamento de carga, o Elastic Beanstalk valida algumas regras de configuração. Se a configuração não atender

às verificações de validação, ela emitirá mensagens instruindo você a fornecer a configuração necessária. As verificações de validação são as seguintes:

- Pelo menos um grupo de segurança deve ser atribuído ao balanceador de carga usando a `SecurityGroups` opção aws:elbv2:loadbalancer ouaws:elb:loadbalancer, dependendo se é um balanceador de carga de aplicativo ou um balanceador de carga clássico, respectivamente. Para obter AWS CLI exemplos, consulteConfigurar com o AWS CLI.
- É necessário que existam regras de tráfego de entrada que permitam que suas EC2 instâncias recebam tráfego do balanceador de carga. Tanto seus grupos EC2 de segurança quanto seus grupos de segurança do balanceador de carga devem fazer referência a essas regras de entrada. Para obter mais informações, consulte a seção Regras de entrada para tráfego abaixo.

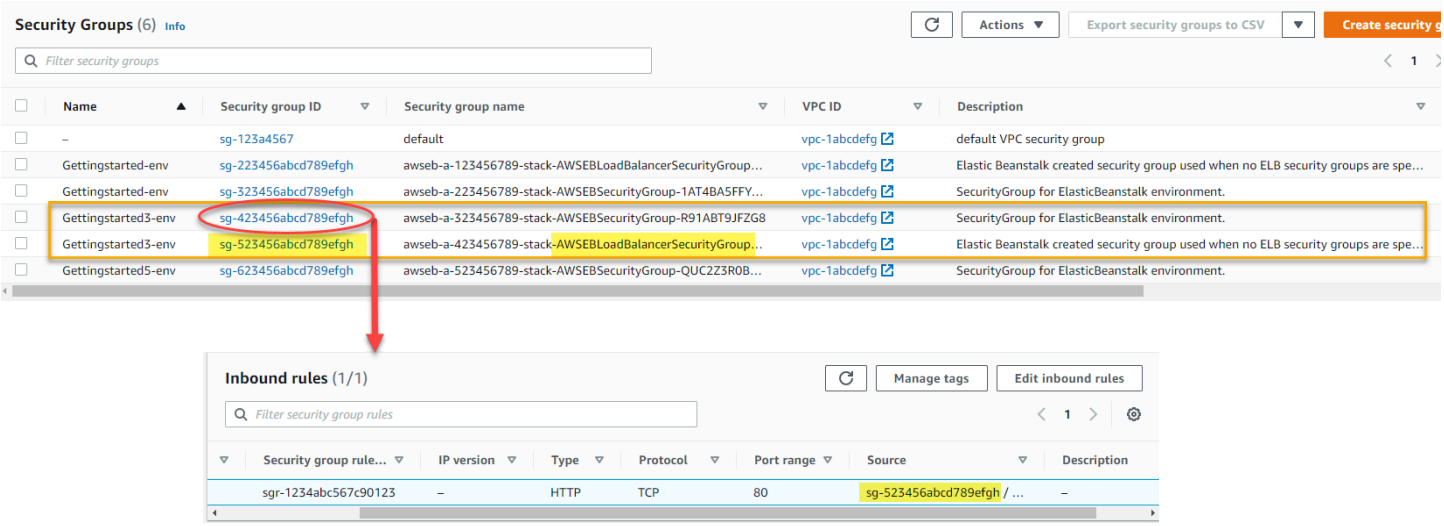
Regras de entrada para tráfego

Os grupos EC2 de segurança de um ambiente de várias instâncias devem incluir uma regra de entrada que faça referência ao grupo de segurança do balanceador de carga. Isso se aplica a ambientes com qualquer tipo de balanceador de carga, dedicado ou compartilhado, e com grupos de segurança de balanceadores de carga personalizados ou padrão.

Você pode visualizar todos os grupos de segurança que estão conectados aos componentes do seu ambiente no EC2 console. A imagem a seguir mostra a lista do EC2 console dos grupos de segurança que o Elastic Beanstalk cria por padrão durante a operação de criação do ambiente.

A tela Grupos de Segurança mostra os ambientes e seus grupos de segurança associados. Tanto GettingStarted-env quanto GettingStarted3-env são ambientes de várias instâncias com balanceadores de carga dedicados. Cada um desses ambientes tem dois grupos de segurança listados, um para as EC2 instâncias e outro para o balanceador de carga. O Elastic Beanstalk cria esses grupos de segurança ao criar os ambientes. GettingStartedO 5-env não tem um grupo de segurança de balanceador de carga, porque tem apenas uma EC2 instância e, portanto, nenhum balanceador de carga.

A tela de regras de entrada detalha o grupo de EC2 segurança das instâncias de GettingStarted 3-env. Este exemplo define as regras de entrada para o grupo EC2 de segurança. Observe que a coluna Fonte nas regras de entrada lista o ID do grupo de segurança do balanceador de carga listado na imagem anterior. Essa regra permite que as EC2 instâncias de GettingStarted3-env recebam tráfego de entrada desse balanceador de carga específico na porta 80.



Para obter mais informações, consulte Alterar grupos de segurança para sua instância e regras do Elastic Load Balancing no Guia EC2 do usuário da Amazon.

Configurando grupos de EC2 segurança e tipos de instância da Amazon usando o AWS CLI

Você pode usar a interface de linha de AWS comando (AWS CLI) para configurar as EC2 instâncias da Amazon em seus ambientes do Elastic Beanstalk.

Configurando grupos EC2 de segurança usando o AWS CLI

Este tópico fornece exemplos de diferentes configurações de grupos EC2 de segurança para ambientes de instância única e com carga balanceada (várias instâncias). Para obter mais informações sobre as opções nesses exemplos, consulte aws:autoscaling:launchconfiguration.

> **Observações**
>
> A operação de criação de ambiente fornece um grupo EC2 de segurança por padrão. Ele também cria um ambiente com um balanceador de carga de aplicativos por padrão.
> A operação de atualização do ambiente pode ser usada para desativar ou ativar o grupo de EC2 segurança padrão para seu ambiente com a opção `DisableDefaultEC2SecurityGroup` booleana. O exemplo 5 mostra como redefinir seu ambiente para a configuração de segurança padrão, caso você a tenha modificado anteriormente.

Os exemplos a seguir mostram um comando create-environment optando por não participar do grupo de EC2 segurança padrão e fornecendo, em vez disso, grupos de segurança personalizados. Como a `DisableDefaultEC2SecurityGroup` opção está definida como`true`, o grupo de EC2 segurança padrão que o Elastic Beanstalk normalmente EC2 associa às instâncias não é criado. Portanto, você deve fornecer a `SecurityGroups` opção a outros grupos de segurança.

Observe que a aws:elasticbeanstalk:environment `EnvironmentType` opção está definida como`SingleInstance`. Para criar um ambiente de instância única, você deve especificar essa opção, porque `LoadBalanced` é o padrão`EnvironmentType`. Como esse ambiente não inclui um balanceador de carga, não precisamos especificar um grupo de segurança de balanceador de carga.

Example 1 — Novo ambiente de instância única com grupos de EC2 segurança personalizados (opções de namespace embutidas)

```
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
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example 2 — Novo ambiente de instância única com grupos de EC2 segurança personalizados (opções de namespace no arquivo) **`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"
  }
]
```

O exemplo a seguir cria um ambiente com balanceamento de carga. Ele especifica a opção `LoadBalancerType` de aws:elasticbeanstalk:environment namespace definida como. `application` Como estamos desativando o grupo de EC2 segurança padrão com a `DisableDefaultEC2SecurityGroup` opção, precisamos fornecer nossos próprios grupos de segurança personalizados para as EC2 instâncias novamente, com a aws:autoscaling:launchconfiguration `SecurityGroups` opção, como no exemplo anterior. Como esse ambiente tem um balanceador de carga para rotear o tráfego, também devemos fornecer grupos de segurança para o balanceador de carga.

Para criar um ambiente com um balanceador de carga clássico, mas com a mesma configuração, atualize a configuração da opção de aws:elasticbeanstalk:environment namespace para. `LoadBalancerType` `classic`

Os diferentes tipos de balanceador de carga têm namespaces diferentes que contêm as opções para especificar os grupos de segurança:

- balanceador de carga de aplicativos — opção aws:elbv2:loadbalancer `SecurityGroups`
- balanceador de carga clássico — opção aws:elb:loadbalancer `SecurityGroups`
- balanceador de carga de rede — como os balanceadores de carga de rede não têm grupos de segurança, configure os grupos de EC2 segurança com identificadores de VPC. Para obter mais informações, consulte Atualizar os grupos de segurança do seu Network Load Balancer no Guia do usuário para Network Load Balancers.

Example 3 — Novo ambiente de várias instâncias com grupos de EC2 segurança personalizados (opções de namespace no arquivo) **`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"
  }
]
```

Você pode desativar o grupo EC2 de segurança padrão para um ambiente existente com o comando update-environment. O comando de exemplo a seguir desativa o grupo EC2 de segurança padrão e atribui grupos de EC2 segurança personalizados às EC2 instâncias do ambiente.

Use os `options.jason` arquivos de exemplo nos exemplos 4 (a), 4 (b) ou 4 (c), dependendo se o ambiente tem balanceamento de carga e do tipo de balanceador de carga. O arquivo de configuração 4 (a) especifica os grupos de segurança para um ambiente de instância única. Como ele não exige um balanceador de carga, fornecemos apenas o grupo de segurança para as EC2 instâncias. Os arquivos de configuração 4 (b) e 4 (c) especificam os grupos de segurança para um balanceador de carga de aplicativo e um balanceador de carga clássico. Para esses casos, também precisamos especificar grupos de segurança para o balanceador de carga.

Example 4 — Atualize um ambiente existente para desativar o grupo de EC2 segurança padrão (opções de namespace no **`options.json`** arquivo)

```
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) — Arquivo de configuração para ambiente de instância única (sem balanceador de carga)

```
### example options.json ###
[
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "DisableDefaultEC2SecurityGroup",
    "Value": "true"
  },
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "SecurityGroups",
    "Value": "sg-abcdef01, sg-abcdef02"
  }
]
```

Para atualizar um ambiente que usa um balanceador de carga do aplicativo, use o `aws:elbv2:loadbalancer` namespace para especificar os grupos de segurança do balanceador de carga.

Example 4 (b) — Arquivo de configuração para ambiente com um balanceador de carga de aplicativo

```
### 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"
  }
]
```

Para atualizar um ambiente que usa um balanceador de carga clássico, use o `aws:elb:loadbalancer` namespace para especificar os grupos de segurança do balanceador de carga.

Example 4 (c) — Arquivo de configuração para ambiente com um balanceador de carga clássico

```
### 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"
  }
]
```

Para retornar seu ambiente ao comportamento e à configuração padrão com o grupo de segurança padrão que o Elastic Beanstalk atribui, use o comando update-environment para definir o. `DisableDefaultEC2SecurityGroup` `false` Para um ambiente de várias instâncias, o Elastic Beanstalk também gerencia os grupos de segurança e as regras de tráfego de rede do balanceador de carga do seu ambiente.

O exemplo a seguir se aplica a um ambiente de instância única ou de várias instâncias (balanceamento de carga):

Example 5 — Atualize um ambiente de volta para usar o grupo de segurança padrão (opções de namespace no **`options.json`** arquivo)

```
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"
  }
]
```

Configurando EC2 com tipos de instância usando o AWS CLI

Este tópico fornece exemplos para configurar os tipos de instância das EC2 instâncias em seu ambiente.

Os dois primeiros exemplos criam um novo ambiente. O comando especifica um tipo de EC2 instância da Amazon, t4g.small, baseado na arquitetura do processador arm64. O Elastic Beanstalk usa como padrão o Image ID (AMI EC2) para as instâncias com base na região, na versão da plataforma e no tipo de instância. O tipo de instância corresponde a uma arquitetura de processador. O parâmetro `solution-stack-name` se aplica à versão da plataforma.

Example 1 – criar um novo ambiente baseado em arm64 (opções de namespace inline)

```
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
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example 2 – criar um novo ambiente baseado em arm64 (opções de namespace no arquivo **`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"
  }
]
```

Os próximos dois exemplos atualizam a configuração de um ambiente existente com o comando update-environment. Neste exemplo, estamos adicionando outro tipo de instância que também se baseia na arquitetura do processador arm64. Para ambientes existentes, todos os tipos de instância adicionados devem ter a mesma arquitetura de processador. Se quiser substituir os tipos de instância existentes pelos de uma arquitetura diferente, você pode fazer isso. Mas certifique-se de que todos os tipos de instância no comando tenham o mesmo tipo de arquitetura.

Example 3 – atualize um ambiente baseado em arm64 existente (opções de namespace inline)

```
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
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example 4 – atualize um ambiente baseado em arm64 existente (opções de namespace no arquivo **`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"
  }
]
```

Os próximos dois exemplos mostram mais comandos de create-environment Esses exemplos não fornecem valores para `InstanceTypes`. Quando valores `InstanceTypes` não são especificados, o Elastic Beanstalk usa como padrão a arquitetura de processador baseada em x86. A ID da imagem (AMI) das EC2 instâncias do ambiente será padronizada de acordo com a região, a versão da plataforma e o tipo de instância padrão. O tipo de instância corresponde a uma arquitetura de processador.

Example 5 – criar um novo ambiente baseado em x86 (opções de namespace inline)

```
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
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example 6 – criar um novo ambiente baseado em x86 (opções de namespace no arquivo **`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"
  }
```

```
]
```

Configurando EC2 instâncias da Amazon com opções de namespace

Você pode usar as opções de configuração no namespace `aws:autoscaling:launchconfiguration` para configurar as instâncias para o seu ambiente, inclusive opções adicionais que não estão disponíveis no console.

> **Important**
>
> A configuração da `BlockDeviceMappings` opção `DisableIMDSv1RootVolumeType`, ou pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

O exemplo de arquivo de configuração a seguir usa as opções básicas de configuração explicadas neste tópico. Para ver exemplos de opções de configuração adicionais quando você precisa especificar grupos de segurança para balanceadores de carga, consulteConfigurar com o 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
```

Os `DisableDefaultEC2SecurityGroup` e não `BlockDeviceMappings` estão disponíveis no console.

Você pode usar `BlockDeviceMappings` para configurar dispositivos de blocos adicionais para as instâncias. Para obter mais informações, consulte Mapeamento de dispositivos de blocos no Guia EC2 do usuário da Amazon.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Configuração do IMDS nas instâncias do ambiente Elastic Beanstalk

Este tópico descreve o Serviço de Metadados de Instância (IMDS).

Os metadados da instância são dados relacionados a uma instância do Amazon Elastic Compute Cloud EC2 (Amazon) que os aplicativos podem usar para configurar ou gerenciar a instância em execução. O serviço de metadados da instância (IMDS) é um componente na instância que o código na instância usa para acessar metadados da instância com segurança. Esse código pode ser o código da plataforma Elastic Beanstalk nas instâncias do seu ambiente AWS , o SDK que seu aplicativo pode estar usando ou até mesmo o próprio código do seu aplicativo. Para obter mais informações, consulte Metadados da instância e dados do usuário no Guia do EC2 usuário da Amazon.

O código pode acessar os metadados da instância de uma instância em execução usando um dos dois métodos: Instance Metadata Service Version 1 (IMDSv1) ou Instance Metadata Service Version 2 (). IMDSv2 IMDSv2 usa solicitações orientadas à sessão e mitiga vários tipos de vulnerabilidades que poderiam ser usadas para tentar acessar o IMDS. Para obter informações sobre esses dois métodos, consulte Como configurar o serviço de metadados da instância no Guia EC2 do usuário da Amazon.

Tópicos

- Suporte de plataforma ao IMDS
- Escolher métodos de IMDS
- Configurar o IMDS usando o console do Elastic Beanstalk
- O namespace aws:autoscaling:launchconfiguration

Suporte de plataforma ao IMDS

Todas as plataformas do Elastic Beanstalk executadas no Amazon Linux 2 e no Amazon Linux 2023 e no Windows Server oferecem suporte a ambos e. IMDSv1 IMDSv2 Para obter mais informações, consulte Configurar o IMDS usando o console do Elastic Beanstalk.

Escolher métodos de IMDS

Ao tomar uma decisão sobre os métodos de IMDS que você deseja que o ambiente suporte, considere os seguintes casos de uso do:

- AWS SDK — Se seu aplicativo usa um AWS SDK, certifique-se de usar a versão mais recente do SDK. Eles AWS SDKs fazem chamadas IMDS e as versões mais recentes do SDK são usadas IMDSv2 sempre que possível. Se você alguma vez desativar IMDSv1 ou se seu aplicativo usar uma versão antiga do SDK, as chamadas do IMDS poderão falhar.
- O código do seu aplicativo — Se seu aplicativo fizer chamadas IMDS, considere usar o AWS SDK para que você possa fazer as chamadas em vez de fazer solicitações HTTP diretas. Assim, não é necessário fazer alterações no código para alternar entre os métodos de IMDS. O AWS SDK é usado IMDSv2 sempre que possível.
- Código da plataforma Elastic Beanstalk — Nosso código faz chamadas IMDS AWS por meio do SDK e IMDSv2 , portanto, é usado em todas as versões de suporte da plataforma. Se seu código usa um up-to-date AWS SDK e faz todas as chamadas IMDS por meio do SDK, você pode desativá-lo com segurança. IMDSv1

Configurar o IMDS usando o console do Elastic Beanstalk

Você pode modificar a configuração da instância Amazon do seu ambiente do Elastic Beanstalk no console EC2 do Elastic Beanstalk.

> **Important**
>
> A configuração da `DisableIMDSv1` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao

atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Para configurar o IMDS em suas EC2 instâncias da Amazon no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
5. Defina Desativar IMDSv1 para impor IMDSv2. IMDSv1Desmarque a opção Desativar para ativar IMDSv1 os dois IMDSv2.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

O namespace aws:autoscaling:launchconfiguration

É possível usar uma opção de configuração no namespace `aws:autoscaling:launchconfiguration` para configurar o IMDS nas instâncias do ambiente.

Important

A configuração da `DisableIMDSv1` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

O exemplo de arquivo de configuração a seguir desativa o IMDSv1 uso da `DisableIMDSv1` opção.

```
option_settings:
  aws:autoscaling:launchconfiguration:
    DisableIMDSv1: true
```

IMDSv1Defina Desativar `true` como para desativar IMDSv1 e aplicar IMDSv2.

IMDSv1Defina Desativar como `false` para ativar ambos IMDSv1 IMDSv2 e.

Auto Scaling de suas instâncias do ambiente do Elastic Beanstalk

Este tópico descreve como você pode personalizar os recursos do Auto Scaling para gerenciar a carga de trabalho do seu ambiente Elastic Beanstalk. Você pode configurar o Auto Scaling para seu ambiente usando o console do Elastic Beanstalk, as opções de configuração de namespace ou a CLI do EB. AWS CLI

Ambientes com balanceamento de carga ou de instância única

Seu AWS Elastic Beanstalk ambiente inclui um grupo de Auto Scaling que gerencia as EC2 instâncias da Amazon em seu ambiente. Em um ambiente de instância única, o grupo de Auto Scaling garante que sempre haja uma instância em execução. Em um ambiente com balanceamento de carga, o grupo é configurado com um intervalo de instâncias a serem executadas, e o Auto Scaling adiciona ou remove instâncias conforme necessário com base na carga.

EC2 Configuração da instância

O grupo Auto Scaling também aplica suas opções de configuração para provisionar e gerenciar as EC2 instâncias em seu ambiente. Você pode modificar a EC2 configuração para alterar o tipo de instância, o par de chaves, o armazenamento do Amazon Elastic Block Store (Amazon EBS) e outras configurações que só podem ser definidas quando você executa uma instância.

Instâncias sob demanda e spot

Como opção, o Elastic Beanstalk pode incluir instâncias spot em seu ambiente e gerenciá-las em combinação com instâncias sob demanda. Você pode configurar o Amazon EC2 Auto Scaling para monitorar e responder automaticamente às mudanças que afetam a disponibilidade de suas instâncias spot ativando o rebalanceamento de capacidade. Você também pode configurar o Estratégia de alocação pontual que o serviço Auto Scaling usa para provisionar instâncias spot em seu ambiente.

Permissões necessárias ao ativar instâncias spot

A habilitação de solicitações de Instância Spot requer o uso de modelos de EC2 lançamento da Amazon. Quando você configura esse recurso durante a criação ou as atualizações do ambiente, o Elastic Beanstalk tenta configurar seu ambiente para usar os modelos de lançamento EC2 da Amazon (se o ambiente ainda não os estiver usando). Nesse caso, se a sua política de usuário não

tiver as permissões necessárias, a criação ou as atualizações do ambiente poderão falhar. Portanto, recomendamos que você use nossa política de usuário gerenciada ou adicione as permissões necessárias às suas políticas personalizadas. Para obter detalhes sobre as permissões necessárias, consulte Permissões necessárias para modelos de execução do .

Gatilhos do ajuste de escala automático

O grupo Auto Scaling usa dois CloudWatch alarmes da Amazon para acionar operações de escalabilidade. Os acionadores padrão são escalados quando a média de tráfego de rede de saída de cada instância é mais alta que 6 MiB ou mais baixa que 2 MiB durante um período de cinco minutos. Para usar o Auto Scaling com eficiência, configure acionadores que são apropriados para sua aplicação, tipo de instância e requisitos de serviço. Você pode dimensionar com base em várias estatísticas, incluindo latência, E/S de disco, utilização de CPU e a contagem de solicitações.

Agende ações de Auto Scaling

Para otimizar o uso de EC2 instâncias da Amazon em seu ambiente por meio de períodos previsíveis de pico de tráfego, configure seu grupo de Auto Scaling para alterar sua contagem de instâncias em um cronograma. Você pode programar alterações em sua configuração do grupo que sejam repetidas uma vez por dia ou uma vez por semana, ou programar as alterações de uma só vez para se preparar para eventos de marketing que irão gerar uma grande quantidade de tráfego para o seu site.

Verificação de integridade do Auto Scaling

O Auto Scaling monitora a integridade de cada EC2 instância da Amazon que ele executa. Se alguma instância for encerrada inesperadamente, o Auto Scaling detectará o encerramento e executará uma instância de substituição. Para configurar o grupo para usar o mecanismo de verificação de integridade do balanceador de carga, consulte Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk.

Tópicos

- Migrando seu ambiente do Elastic Beanstalk para modelos de lançamento
- Suporte de instância spot para seu ambiente Elastic Beanstalk
- Acionadores do Auto Scaling para seu ambiente Elastic Beanstalk
- Ações programadas de Auto Scaling para seus ambientes do Elastic Beanstalk
- Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk

Migrando seu ambiente do Elastic Beanstalk para modelos de lançamento

A partir de 1º de outubro de 2024, o Amazon EC2 Auto Scaling não oferece mais suporte às configurações de lançamento para novas contas. As contas criadas antes dessa data podem ter configurações de lançamento.

Recomendamos migrar para modelos de lançamento para obter os seguintes benefícios:

- Disponibilidade aprimorada para seus aplicativos
- Melhor otimização das cargas de trabalho em seus grupos de Auto Scaling
- Acesso aos recursos mais recentes EC2 e do Auto Scaling

Para obter mais informações, consulte as configurações de lançamento do Auto Scaling no Guia do usuário do Amazon EC2 Auto Scaling.

Configurações de opções para modelos de inicialização

Para migrar seu ambiente das configurações de lançamento para os modelos de execução, defina uma das seguintes opções de configuração:

- `RootVolumeType` definida como gp3. É possível definir essa opção com o console ou o namespace.
- A opção `BlockDeviceMappings` contém gp3. Você pode definir esta opção com o console ou o namespace.
- Opção `DisableIMDSv1` definida como true. Recomendamos definir essa opção usando o namespace.
- Opção `EnableSpot` definida como true. Para obter mais informações, consulte Habilitando instâncias spot.

> **Important**
>
> Depois que um ambiente começa a usar modelos de execução, o Elastic Beanstalk não volta às configurações de execução, mesmo que você remova as opções de configuração que originalmente acionaram o uso dos modelos de execução.

Confirme se seu ambiente tem configurações de inicialização ou modelos de inicialização

Você pode confirmar se seu ambiente já usa modelos de execução, ou se está usando configurações de execução, inspecionando o modelo de CloudFormation pilha.

Para inspecionar o modelo de CloudFormation pilha do seu ambiente

1. Abra o AWS CloudFormation console em https://console.aws.amazon.com/cloudformation.
2. Na barra de navegação na parte superior da tela, escolha a AWS região em que você criou o ambiente.
3. Na página Pilhas do CloudFormation console, inspecione a coluna Descrição.

   Localize e selecione a pilha para o ambiente do Elastic Beanstalk. CloudFormation exibe os detalhes da pilha do ambiente.
4. Nos detalhes da pilha, selecione a guia Modelo.

   Usando a pesquisa de página do navegador, você pode pesquisar o texto do modelo para launchtemplate ou launchconfiguration.

Para obter mais informações, consulte Exibir informações da pilha no Guia do Usuário do AWS CloudFormation .

Permissões necessárias para modelos de execução do

A política padrão de AWSElasticBeanstalkManagedUpdatesCustomerRolePolicyfunção de serviço gerenciado do Elastic Beanstalk fornece as permissões necessárias para criar e gerenciar modelos de lançamento. O Elastic Beanstalk deve gerenciar modelos de lançamento para concluir várias operações de ambiente, incluindo a criação de ambientes.

Se você anexar políticas personalizadas a uma função de serviço do Elastic Beanstalk, verifique se a função de serviço inclui as seguintes permissões para criar modelos de lançamento. Essas permissões permitem que o Elastic Beanstalk crie e atualize com sucesso ambientes em sua conta:

Permissões necessárias para modelos de EC2 lançamento da Amazon

- `ec2:RunInstances`
- `ec2:CreateLaunchTemplate`

- `ec2:CreateLaunchTemplateVersions`
- `ec2:DeleteLaunchTemplate`
- `ec2:DeleteLaunchTemplateVersions`
- `ec2:DescribeLaunchTemplate`
- `ec2:DescribeLaunchTemplateVersions`

O exemplo a seguir da política do IAM inclui essas permissões.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersions",
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteLaunchTemplateVersions",
        "ec2:DescribeLaunchTemplate",
        "ec2:DescribeLaunchTemplateVersions"
      ],
      "Resource": [
                "*"
            ]
    }
  ]
}
```

Para obter mais informações, consulte Gerenciar funções de serviço do Elastic Beanstalk e Gerenciar políticas de usuário do Elastic Beanstalk.

Mais informações sobre modelos de inicialização

Para saber mais sobre os modelos de lançamento, consulte os modelos de lançamento do Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling.

Para saber mais sobre a AWS transição para modelos de lançamento e os benefícios que eles oferecem, consulte O Amazon EC2 Auto Scaling não adicionará mais suporte para novos EC2 recursos às configurações de lançamento no AWS blog de computação.

> **Important**
>
> Você não precisa seguir o procedimento mencionado neste artigo do blog para fazer a transição de um ambiente antigo para modelos de inicialização. Para migrar um ambiente existente do Elastic Beanstalk para modelos de lançamento, defina uma das opções listadas em. Configurações de opções para modelos de inicialização

Suporte de instância spot para seu ambiente Elastic Beanstalk

Este tópico descreve as opções de configuração que estão disponíveis para você gerenciar a capacidade e o balanceamento de carga das Instâncias Spot em seu ambiente do Elastic Beanstalk. Ele também fornece detalhes e exemplos dos métodos que você pode usar para configurar essas opções. Você pode usar o console do Elastic Beanstalk, as opções de configuração do namespace ou o EB CLI para gerenciar AWS CLIas opções de configuração.

Minimize as interrupções de instâncias spot com o rebalanceamento de capacidade

Para ajudar a minimizar o impacto das interrupções de instâncias spot em seu aplicativo, você pode ativar a opção de rebalanceamento de capacidade incluída no Amazon Auto EC2 Scaling.

> **Important**
>
> A demanda por instâncias spot pode variar significativamente de momento a momento, e a disponibilidade de instâncias spot também pode variar significativamente, dependendo de quantas EC2 instâncias Amazon não utilizadas estão disponíveis. É sempre possível que sua instância spot seja interrompida.

Quando você ativa o rebalanceamento de capacidade, tenta substituir EC2 automaticamente as instâncias spot em um grupo de Auto Scaling antes que elas sejam interrompidas. Para habilitar esse recurso, use o console do Elastic Beanstalk para configurar o grupo do Auto Scaling. Como alternativa, é possível definir a `EnableCapacityRebalancing` opção de configuração do Elastic Beanstalk no `true` namespace aws:autoscaling:asg.

Para obter mais informações, consulte Rebalanceamento de capacidade no Guia do usuário do Amazon EC2 Auto Scaling e Spot Instance Interruptions no Guia do usuário da EC2 Amazon.

Tipos de instância mais antigos e suporte a instâncias spot

Algumas AWS contas mais antigas podem fornecer ao Elastic Beanstalk tipos de instância padrão que não são compatíveis com instâncias spot. Se você habilitar as solicitações de Instância Spot e ver o erro Nenhum dos tipos de instância que você especificou suporta Spot, atualize sua configuração com tipos de instância compatíveis com Instâncias Spot. Para escolher os tipos de instância spot, use o Spot Instance Advisor.

Tópicos

- Habilitando instâncias spot para seu ambiente
- Estratégia de alocação de instâncias spot
- Gerenciando instâncias sob demanda e instâncias spot
- Configuração de capacidade para seu ambiente Elastic Beanstalk

Habilitando instâncias spot para seu ambiente

Para aproveitar as Instâncias EC2 Spot da Amazon, defina a `EnableSpot` opção para seu ambiente. Em seguida, o grupo de Auto Scaling do seu ambiente combina as opções de EC2 compra da Amazon e mantém uma combinação de instâncias sob demanda e spot.

Você pode usar o console do Elastic Beanstalk, as opções de configuração de namespace ou AWS CLIo EB CLI para habilitar solicitações de Instância Spot para seu ambiente.

Antes de habilitar as Instâncias Spot para seu ambiente, familiarize-se com as opções de configuração de Auto Scaling, capacidade e balanceamento de carga que estão disponíveis. Os requisitos do seu aplicativo relacionados à carga de trabalho, ao impacto das interrupções de instâncias e aos preços são considerações importantes em seu planejamento para habilitar instâncias spot.

Os tópicos a seguir fornecem detalhes sobre as opções de Auto Scaling e gerenciamento de capacidade e como seu uso combinado afeta seu ambiente. Existem procedimentos e exemplos de configurações para informá-lo e orientá-lo sobre as várias opções e como configurá-las. Também oferecemos ferramentas e recursos para ajudar você a gerenciar sua configuração e responder aos eventos. Você pode programar alterações automatizadas em sua configuração com base em períodos previsíveis de tráfego, configurar acionadores para responder a fatores como volume de tráfego e configurar o monitoramento e as verificações de integridade do Auto Scaling.

Para obter informações mais detalhadas sobre instâncias spot, incluindo explicação dos principais conceitos e melhores práticas, consulte Instâncias spot no Guia EC2 do usuário da Amazon.

> **Important**
>
> A configuração da `EnableSpot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Estratégia de alocação de instâncias spot

Você pode selecionar qualquer uma das estratégias de alocação listadas neste tópico para seu ambiente do Elastic Beanstalk. Use o console do Elastic Beanstalk, as opções de configuração de namespace AWS CLIou o, para definir e configurar a estratégia de alocação de instâncias spot e os atributos relacionados ao seu ambiente.

A Amazon EC2 aplica uma estratégia de alocação para gerenciar e provisionar instâncias spot para seu ambiente. Cada estratégia de alocação otimiza as instâncias alocadas com base em como ela é definida para lidar com a capacidade disponível, o preço e a seleção dos tipos de instância.

O Amazon EC2 Auto Scaling fornece as seguintes estratégias de alocação para instâncias spot.

- Capacidade otimizada (padrão)
  - Solicita instâncias spot do pool, com capacidade ideal para o número de instâncias que estão sendo executadas.
  - Essa estratégia funciona bem para cargas de trabalho em que a possibilidade de interrupção do serviço deve ser minimizada.
- Capacidade de preço otimizada
  - Solicita instâncias spot dos pools que têm a menor chance de interrupção e o menor preço possível.
  - Essa é a opção preferível para a maioria das cargas de trabalho do Spot.
- Capacidade otimizada priorizada
  - Solicita instâncias spot com base primeiro na disponibilidade da capacidade, ao mesmo tempo em que honra sua escolha de priorização do tipo de instância com base no melhor esforço. Você

pode fornecer uma lista de tipos de instância, ordenados por prioridade, ao configurar as opções de instância spot para o Elastic Beanstalk.

  - Essa estratégia é boa para cargas de trabalho que exigem o mínimo de interrupção do serviço, e a priorização de um tipo específico de instância é importante.
- Preço mais baixo
  - Solicita instâncias spot do pool de menor preço com instâncias disponíveis.
  - É importante tomar cuidado ao usar essa estratégia, pois ela considera apenas o preço da instância e não a disponibilidade da capacidade, o que resultará em altas taxas de interrupção.

Para obter mais detalhes sobre cada estratégia de alocação, consulte Estratégias de alocação para vários tipos de instância no Guia do usuário do Amazon Auto EC2 Scaling.

Para ajudá-lo a entender qual estratégia de alocação é mais adequada para atender aos requisitos do seu ambiente, consulte Escolha a estratégia de alocação spot apropriada no Guia EC2 do usuário da Amazon.

Gerenciando instâncias sob demanda e instâncias spot

Você pode iniciar e escalar automaticamente uma frota de instâncias sob demanda e instâncias spot em um único grupo do Auto Scaling. As opções a seguir podem ser usadas em conjunto para configurar como o serviço Auto Scaling gerencia instâncias spot e instâncias sob demanda em seu ambiente.

Você pode configurar essas opções para seu ambiente usando o console do Elastic Beanstalk, as opções de configuração de namespace ou a CLI do EB. AWS CLI

Essas opções fazem parte do namespace aws:ec2:instances:

- `EnableSpot`- Quando definido com `true` essa configuração, ativa as solicitações de Instância Spot para seu ambiente.
- `SpotFleetOnDemandBase`- Define o número mínimo de instâncias sob demanda que seu grupo de Auto Scaling provisiona antes de considerar as instâncias spot à medida que seu ambiente aumenta.
- `SpotFleetOnDemandAboveBasePercentage`- A porcentagem de instâncias sob demanda como parte da capacidade adicional que seu grupo de Auto Scaling provisiona além `SpotOnDemandBase` das instâncias.

As opções listadas anteriormente se correlacionam com as seguintes opções no namespace aws:autoscaling:asg:

- `MinSize`- O número mínimo de instâncias que você deseja em seu grupo de Auto Scaling.
- `MaxSize`- O número máximo de instâncias que você deseja em seu grupo de Auto Scaling.

> **Important**
>
> A configuração da `EnableSpot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Aplicando os dois conjuntos de opções de namespace

Os pontos a seguir descrevem como a combinação dessas configurações de opção afeta o dimensionamento do seu ambiente.

- Somente `MinSize` determina a capacidade inicial do seu ambiente — o número de instâncias que você deseja executar no mínimo.
- `SpotFleetOnDemandBase` não afeta a capacidade inicial. Quando o spot está habilitado, essa opção determina quantas instâncias sob demanda são provisionadas antes que qualquer instância spot seja considerada.
- Considere quando`SpotFleetOnDemandBase` for menor que `MinSize`. Você ainda vai ter exatas `MinSize` instâncias como capacidade inicial. No mínimo `SpotFleetOnDemandBase` delas devem ser instâncias sob demanda.
- Considere quando `SpotFleetOnDemandBase` for maior que `MinSize`. À medida que o ambiente tiver aumento da escala, você terá, pelo menos,uma quantidade adicional de instâncias igual à diferença entre os dois valores. Em outras palavras, você tem a garantia de que obterá pelo menos uma `(SpotFleetOnDemandBase - MinSize)` instância adicional sob demanda antes de atender ao `SpotFleetOnDemandBase` requisito.

Ambientes de instância única

Em ambientes de produção, as instâncias spot são particularmente úteis como parte de um ambiente escalável e com balanceamento de carga. Não recomendamos o uso de spot em um ambiente de instância única. Se as instâncias spot não estiverem disponíveis, você poderá perder toda a capacidade (uma única instância) do seu ambiente. Você ainda pode usar uma instância spot em um ambiente de instância única para desenvolvimento ou teste. Neste caso, certifique-se de definir ambas `SpotFleetOnDemandBase` e `SpotFleetOnDemandAboveBasePercentage` de zerar. Quaisquer outras configurações resultam em uma instância sob demanda.

Exemplos de configurações de opções de dimensionamento

Os exemplos a seguir demonstram diferentes cenários de configurações das várias opções de escalabilidade. Todos os exemplos pressupõem um ambiente com balanceamento de carga e com solicitações de instância spot habilitadas.

Example 1: Sob demanda e spot como parte da capacidade inicial

Configurações de opção

Opção	Namespace	Valor
`MinSize`	`aws:autoscaling:asg`	`10`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

Neste exemplo, o ambiente começa com dez instâncias, das quais sete são sob demanda (quatro de base e 50% das seis acima da base) e três são spot. O ambiente pode ser expandido até 24 instâncias. Conforme ele é expandido, a quantidade sob demanda na parte da frota acima das quatro instâncias sob demanda de base é mantida em 50%, até um máximo de 24 instâncias no geral, das quais 14 são sob demanda (quatro de base e 50% das 20 acima da base) e dez são spot.

Example 2: Toda a capacidade inicial sob demanda

Configurações de opção

Opção	Namespace	Valor
`MinSize`	`aws:autoscaling:asg`	`4`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

Neste exemplo, o ambiente começa com quatro instâncias, todas sob demanda. O ambiente pode ser expandido até 24 instâncias. Conforme ele é expandido, a quantidade sob demanda na parte da frota acima das quatro instâncias sob demanda de base é mantida em 50%, até um máximo de 24 instâncias no geral, das quais 14 são sob demanda (quatro de base e 50% das 20 acima da base) e dez são spot.

Example 3: Base adicional sob demanda além da capacidade inicial

Configurações de opção

Opção	Namespace	Valor
`MinSize`	`aws:autoscaling:asg`	`3`
`MaxSize`	`aws:autoscaling:asg`	`24`
`SpotFleetOnDemandBase`	`aws:ec2:instances`	`4`
`SpotFleetOnDemandAboveBasePercentage`	`aws:ec2:instances`	`50`

Neste exemplo, o ambiente começa com três instâncias, todas sob demanda. O ambiente pode ser expandido até 24 instâncias. A primeira instância adicional acima das três iniciais é sob demanda,

para concluir as quatro instâncias sob demanda de base. À medida que ele é expandido ainda mais, a quantidade sob demanda, na parte da frota acima das quatro instâncias sob demanda de base, é mantida em 50%, até um máximo de 24 instâncias no geral, das quais 14 são sob demanda (quatro de base e 50% das 20 acima da base) e dez são spot.

Configuração de capacidade para seu ambiente Elastic Beanstalk

Este tópico descreve as diferentes abordagens para configurar a capacidade do Auto Scaling para seu ambiente do Elastic Beanstalk. Você pode usar o console do Elastic Beanstalk, a AWS CLI CLI do EB ou as opções de namespace.

> **Important**
>
> A configuração da `EnableSpot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Configurar usando o console

Você pode configurar o gerenciamento de capacidade de um grupo do Auto Scaling editando Capacidade na página de configuração do ambiente no console do Elastic Beanstalk.

Para configurar a capacidade do grupo Auto Scaling no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Grupo do Auto Scaling, defina as configurações a seguir.
   - Tipo de ambiente: selecione Carga balanceada.

- Instâncias mínimas — O número mínimo de EC2 instâncias que o grupo deve conter a qualquer momento. O grupo começa com o número mínimo de contagens e adiciona instâncias quando a condição de acionador aumentar a escala na vertical é atendida.
- Número máximo de instâncias — O número máximo de EC2 instâncias que o grupo deve conter a qualquer momento.

> **Note**
>
> Se você usar atualizações cumulativas, verifique se a contagem máxima de instância é mais alta que a configuração do Minimum instances in service (Mínimo de instâncias em serviço) para atualizações cumulativas.

- Composição da frota— O padrão é instâncias sob demanda. Para habilitar as solicitações de instância spot , selecione Opções e Instâncias de compras combinadas.

> **Important**
>
> A configuração da `EnableSpot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

As opções a seguir são habilitadas se você selecionar Habilitar as solicitações de Instância Spot:

- Estratégia de alocação spot — determina o método usado para gerenciar e provisionar as instâncias spot em seu ambiente, com base na capacidade disponível, no preço e na seleção dos tipos de instância. Selecione entre Capacidade otimizada (padrão), Capacidade de preço otimizada, Capacidade otimizada priorizada ou Preço mais baixo. Para obter uma descrição de cada estratégia de alocação e obter mais informações, consultethe section called “Estratégia de alocação pontual”.

- Preço spot máximo — Para recomendações sobre as opções de preço máximo para instâncias spot, consulte o histórico de preços de instâncias spot no Guia EC2 do usuário da Amazon.
- Base sob demanda – O número mínimo de instâncias sob demanda que o grupo do Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente tiver aumento de escala.
- Sob demanda acima da base – O percentual de instâncias sob demanda como parte da capacidade adicional que o grupo do Auto Scaling provisiona além das instâncias sob demanda.

> **Note**
>
> As opções On-Demand base (Base sob demanda) e On-Demand above base (Sob demanda acima da base) correlacionam-se com as de opções das instâncias Min (Mín.) e Max (Máx.) listadas anteriormente. Para obter mais informações sobre essas opções e exemplos, consulte the section called “Suporte para instâncias spot”.

- Rebalanceamento de capacidade — Essa opção só é relevante quando há pelo menos uma instância spot em seu grupo de Auto Scaling. Quando esse recurso é ativado, tenta substituir EC2 automaticamente as Instâncias Spot no grupo Auto Scaling antes que elas sejam interrompidas, minimizando as interrupções das Instâncias Spot em seus aplicativos. Para obter mais informações, consulte Rebalanceamento de capacidade no Guia do usuário do Amazon Auto EC2 Scaling

- Arquitetura — A arquitetura do processador para suas EC2 instâncias. A arquitetura do processador determina os tipos de EC2 instância que ficam disponíveis no próximo campo.
- Tipos de instância — Os tipos de EC2 instância da Amazon lançados para executar seu aplicativo. Para obter detalhes, consulte the section called “Tipos de instância”.
- ID AMI — A imagem da máquina que o Elastic Beanstalk usa para EC2 iniciar instâncias da Amazon em seu ambiente. Para obter detalhes, consulte the section called “ID de AMI”.
- Availability Zones (Zonas de disponibilidade): escolha o número de zonas de disponibilidade nas quais distribuir as instâncias do ambiente. Por padrão, o grupo de Auto Scaling executa instâncias uniformemente em todas as regiões utilizáveis. Para concentrar suas instâncias em um número menor de zonas, escolha o número de zonas a serem usadas. Para ambientes de produção, use pelo menos duas zonas para garantir que seu aplicativo está disponível caso uma zona de disponibilidade fique fora.

- Placement (Posicionamento) (opcional): escolha as zonas de disponibilidade a serem usadas. Use esta configuração se suas instâncias precisarem se conectar a recursos em regiões específicas ou se você tiver adquirido instâncias reservadas, que são específicas da região. Se você executar seu ambiente em uma VPC personalizada, não poderá configurar essa opção. Em uma VPC personalizada, é possível escolher as zonas de disponibilidade para as sub-redes que você atribui para o seu ambiente.
- Scaling cooldown (Desaquecimento da escalabilidade): o tempo de espera, em segundos, para que as instâncias sejam executadas ou encerradas depois da escalabilidade, antes de continuar a avaliar triggers. Para obter mais informações, consulte Desaquecimento da escalabilidade.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configuração usando opções de namespace

O Elastic Beanstalk fornece opções de configuração para as configurações do Auto Scaling em dois namespaces: `aws:autoscaling:asg` e `aws:ec2:instances`.

O namespace aws:autoscaling:asg

O namespace `aws:autoscaling:asg` fornece opções para escalabilidade e disponibilidade geral.

O arquivo de configuração demonstrativo a seguir configura o grupo de Auto Scaling para usar de duas a quatro instâncias, zonas de disponibilidade específicas e um período de desaquecimento de 12 minutos (720 segundos). Ele permite o rebalanceamento de capacidade para instâncias spot. Essa `EnableCapacityRebalancing` opção só tem efeito se `EnableSpot` estiver definida como `true` no `aws:ec2:instances`namespace, conforme mostrado no exemplo do arquivo de configuração a seguir a este.

```
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
```

O namespace aws:ec2:instances

> **Note**
>
> Quando você atualiza a configuração do seu ambiente e remove um ou mais tipos de instância da `InstanceTypes` opção, o Elastic Beanstalk encerra EC2 todas as instâncias da Amazon em execução em qualquer um dos tipos de instância removidos. O grupo d Auto Scaling do seu ambiente executa novas instâncias, conforme necessário, para completar a capacidade desejada, usando os tipos de instância especificados atuais.

O namespace `aws:ec2:instances` fornece opções relacionadas às instâncias do ambiente, incluindo o gerenciamento de instância spot. Ele complementa `aws:autoscaling:launchconfiguration` e `aws:autoscaling:asg`.

O exemplo a seguir de arquivo de configuração configura o grupo do Auto Scaling para habilitar solicitações de instância spot para seu ambiente. Ele designa três tipos de instância possíveis que podem ser usados. Pelo menos uma instância sob demanda é usadapara capacidade de linha de base, e um percentual sustentado de 33% de instâncias sob demanda é usado para qualquer capacidade adicional.

A configuração define a estratégia de alocação spot como`capacity-optimized-prioritized`. Essa estratégia de alocação específica prioriza as execuções da instância a partir do pool com base na ordem dos tipos de instância especificados na opção. `InstanceTypes` Se não `SpotAllocationStrategy` for especificado, o padrão é. `capacity-optimized`

```
option_settings:
  aws:ec2:instances:
    EnableSpot: true
    InstanceTypes: 't2.micro,t3.micro,t3.small'
    SpotAllocationStrategy: capacity-optimized-prioritized
    SpotFleetOnDemandBase: '1'
    SpotFleetOnDemandAboveBasePercentage: '33'
```

Para escolher os tipos de instância spot, use o Spot Instance Advisor.

> **Important**
>
> A configuração da `EnableSpot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer

as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Configuração usando o AWS CLI

Esta seção fornece exemplos de como você pode usar o comando AWS CLI create-environment para configurar seu ambiente com as opções Auto Scaling e Capacity descritas nessas seções. Você notará que as configurações de namespace para `aws:autoscaling:asg`e `aws:ec2:instances`, conforme descrito na seção anterior de opções de configuração de namespace, também estão definidas com este exemplo.

A interface de linha de AWS comando fornece comandos para criar e configurar ambientes do Elastic Beanstalk. Com a `--option-settings` opção, você pode transmitir opções de namespace compatíveis com o Elastic Beanstalk. Isso significa que as opções de configuração do namespace descritas anteriormente podem ser passadas para AWS CLI comandos aplicáveis para configurar seu ambiente do Elastic Beanstalk.

> **Note**
>
> Você também pode usar o comando update-environment com `--option-settings` para adicionar ou atualizar opções de namespace. Se você precisar remover qualquer opção de namespace do seu ambiente, use o update-environment comando com. `--options-to-remove`

O exemplo a seguir cria um novo ambiente. Consulte o tópico anterior Opções de configuração de namespace para obter mais contexto sobre as opções que são passadas.

A primeira opção listada, `IamInstanceProfile` no aws:autoscaling:launchconfiguration namespace, é o perfil da instância do Elastic Beanstalk. É necessário quando você cria um novo ambiente.

Example — criar ambiente com opções de Auto Scaling (opções de namespace embutidas)

```
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**
>
> A configuração da `EnableSpot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulte Migrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example —criar ambiente com opções de Auto Scaling (opções de namespace no arquivo) **`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"
    }
]
```

Configuração usando o EB CLI

Ao criar um ambiente usando o comando eb create, você pode especificar algumas opções relacionadas ao grupo do Auto Scaling no seu ambiente. Essas são algumas das opções que ajudam você a controlar a capacidade do seu ambiente.

`--single`

Cria o ambiente com uma EC2 instância da Amazon e sem balanceador de carga. Se você não usa essa opção, um balanceador de carga é adicionado ao ambienteque é criado.

`--enable-spot`

Habilita solicitações de instância spot para seu ambiente.

> **Important**
>
> A configuração da `enable-spot` opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .

As opções a seguir para o comando eb create podem ser utilizadas somente com `--enable-spot`.

`--instance-types`

Lista os tipos de EC2 instância da Amazon que você deseja que seu ambiente use.

`--spot-max-price`

O preço máximo por hora, em USD, que você está disposto a pagar por uma instância spot. Para recomendações sobre as opções de preço máximo para instâncias spot, consulte o histórico de preços de instâncias spot no Guia EC2 do usuário da Amazon.

`--on-demand-base-capacity`

O número mínimo de instâncias sob demanda que o grupo de Auto Scaling provisiona antes de considerar as instâncias spot à medida que o ambiente é dimensionado.

`--on-demand-above-base-capacity`

A porcentagem de instâncias sob demanda como parte da capacidade adicional que o grupo de Auto Scaling provisiona além do número de instâncias especificado pela opção `--on-demand-base-capacity`.

O exemplo a seguir cria um ambiente e configura o grupo de Auto Scaling para habilitar solicitações de instância spot para o novo ambiente. Para este exemplo, três tipos de instância podem ser usados.

```
$ eb create --enable-spot --instance-types "t2.micro,t3.micro,t3.small"
```

> **Important**
>
> Há outra opção nomeada de forma semelhante chamada `--instance-type` (sem "s") que a EB CLI reconhece somente ao processar instâncias sob demanda. Não use `--instance-type` (sem "s") com a opção `--enable-spot`. Se você fizer isso, a EB CLI o ignorará. Em vez disso, use `--instance-types` (com "s") com a opção `--enable-spot`.

Acionadores do Auto Scaling para seu ambiente Elastic Beanstalk

O grupo Auto Scaling em seu ambiente do Elastic Beanstalk usa dois CloudWatch alarmes da Amazon para acionar operações de escalabilidade. Os triggers padrão são dimensionados quando a média de tráfego de rede de saída de cada instância é mais alta que 6 MB ou mais baixa que 2 MB durante um período de cinco minutos. Para usar o Amazon EC2 Auto Scaling de forma eficaz, configure gatilhos apropriados para seu aplicativo, tipo de instância e requisitos de serviço. Você pode dimensionar com base em várias estatísticas, incluindo latência, E/S de disco, utilização de CPU e a contagem de solicitações.

Para obter mais informações sobre CloudWatch métricas e alarmes, consulte Amazon CloudWatch Concepts no Amazon CloudWatch User Guide.

Configurar triggers de Auto Scaling

É possível configurar os triggers que ajustam o número de instâncias no grupo de Auto Scaling do seu ambiente no console do Elastic Beanstalk.

Como configurar triggers no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Scaling triggers (Escalabilidade de triggers), defina as seguintes configurações:
   - Metric (Métrica): métrica usada para o trigger de Auto Scaling.
   - Statistic (Estatística): cálculo de estatísticas que o trigger deve usar, como `Average`.
   - Unit (Unidade): unidade da métrica do trigger, como Bytes.

- Período — Especifica com que frequência a Amazon CloudWatch mede as métricas do seu gatilho.
- Breach duration (Duração da ruptura): quantidade de tempo, em minutos, em que uma métrica pode estar fora dos limites superiores e inferiores antes de acionar uma operação de escalabilidade.
- Upper threshold (Limite superior): se a métrica ficar acima desse número pela duração da violação, uma operação de escalabilidade será acionada.
- Incremento de escalabilidade — O número de EC2 instâncias da Amazon a serem adicionadas ao realizar uma atividade de escalabilidade.
- Lower threshold (Limite inferior): se a métrica ficar abaixo desse número pela duração da violação, uma operação de escalabilidade será acionada.
- Incremento de redução — O número de EC2 instâncias da Amazon a serem removidas ao realizar uma atividade de escalabilidade.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

O namespace aws:autoscaling:trigger

O Elastic Beanstalk fornece opções de configuração para configurações de Auto Scaling no namespace `aws:autoscaling:trigger`. As configurações neste namespace estão organizadas pelo recurso ao qual elas se aplicam.

```
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
```

Ações programadas de Auto Scaling para seus ambientes do Elastic Beanstalk

Para otimizar o uso das EC2 instâncias da Amazon em seu ambiente por meio de períodos previsíveis de pico de tráfego, configure seu grupo do Amazon EC2 Auto Scaling para alterar sua contagem de instâncias de acordo com uma programação. Você pode configurar seu ambiente com uma ação recorrente para expandir todos os dias de manhã e reduzir à noite, quando o tráfego é baixo. Por exemplo, se você tiver um evento de marketing que irá direcionar o tráfego para seu site por um período limitado, você poderá programar um evento único para expandir quando ele começar e outro para reduzir quando ele terminar.

Você pode definir até 120 ações programadas ativas por ambiente. O Elastic Beanstalk também retém até 150 ações programadas expiradas que você pode reutilizar atualizando suas configurações.

Configurar ações programadas

Você pode criar ações programadas para o grupo de Auto Scaling do ambiente no console do Elastic Beanstalk.

Como configurar ações programadas no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Time-based Scaling (Escalabilidade baseada em tempo), escolha Add scheduled action (Adicionar ação programada).
6. Preencha as seguintes configurações de ações programadas:

   - Name (Nome): especifique um nome exclusivo de até 255 caracteres alfanuméricos sem espaços.
   - Instances (Instâncias): selecione a contagem de instâncias mínimas e máximas para se inscrever no grupo de Auto Scaling.
   - Desired capacity (Capacidade desejada) (opcional): defina a capacidade inicial desejada para o grupo de Auto Scaling. Depois que a ação programada for aplicada, os triggers ajustarão a capacidade desejada com base em suas configurações.

- Occurrence (Ocorrência): escolha Recurring (Recorrente) para repetir a ação de escalabilidade em uma programação.
- Start time (Hora de início): para ações únicas, selecione a data e a hora para executar a ação.

  Para ações recorrentes, é opcional definir um horário de início. Especifique-o para escolher a hora de início em que a ação é executada. Após esse horário, a ação se repete de acordo com a expressão Recurrence (Recorrência).
- Recurrence (Recorrência): Use uma expressão Cron para especificar a frequência com a qual você deseja executar a ação programada. Por exemplo, `30 6 * * 2` executa a ação todas as terças-feiras às 6:30 horas UTC.
- End time (Hora de término) (opcional): opcional para ações recorrentes. Se especificado, a ação se repete de acordo com a expressão Recurrence (Recorrência) e não será executada novamente após esse horário.

  Quando uma ação programada é finalizada, o Auto Scaling não volta automaticamente às suas configurações anteriores. Configure uma segunda ação programada para retornar o Auto Scaling para as configurações originais, conforme necessário.

7. Escolha Adicionar.
8. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

> **Note**
>
> Ações agendadas não serão salvas até serem aplicadas.

O namespace aws:autoscaling:scheduledaction

Se você precisar configurar um grande número de ações programadas, poderá usar arquivos de configuração ou a API do Elastic Beanstalk para aplicar alterações da opção de configuração de um arquivo YAML ou JSON. Esses métodos também permitem que você acesse a opção `Suspend` para desativar uma ação programada recorrente temporariamente.

> **Note**
>
> Ao trabalhar com as opções de configuração de ações programadas fora do console, use o formato de tempo da ISO 8601 para especificar horários de início e término em UTC. Por exemplo, 2015-04-28T04:07:02Z. Para obter mais informações sobre o formato de hora da

ISO 8601, consulte Formatos de data e hora. As datas devem ser exclusivas em todas as ações programadas.

O Elastic Beanstalk oferece opções de configuração para definições de ações programadas no namespace `aws:autoscaling:scheduledaction`. Use o campo `resource_name` para especificar o nome da ação programada.

Example S cheduled-scale-up-specific -time-long.config

Este arquivo de configuração instrui o Elastic Beanstalk a aumentar de cinco instâncias para 10 instâncias em 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

Para usar a sintaxe abreviada com a EB CLI ou com arquivos de configuração, coloque o nome do recurso antes do namespace.

```
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

Esse arquivo de configuração instrui o Elastic Beanstalk a reduzir em 2015-12-12T07:00:00Z.

```
option_settings:
  ScheduledScaleDownSpecificTime.aws:autoscaling:scheduledaction:
    MinSize: '1'
    MaxSize: '1'
    DesiredCapacity: '1'
    StartTime: '2015-12-12T07:00:00Z'
```

Example S cheduled-periodic-scale-up .config

Este arquivo de configuração instrui o Elastic Beanstalk a fazer escalonamento horizontal todos os dias às 9h. A ação foi programada para começar em 14 de maio de 2015 e terminar em 12 de janeiro de 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 cheduled-periodic-scale-down .config

Este arquivo de configuração instrui o Elastic Beanstalk a reduzir para nenhuma instância em execução todos os dias às 18h. Se você sabe que seu aplicativo está ocioso principalmente fora do horário comercial, você pode criar uma ação programada semelhante. Se o aplicativo tiver de ser desativado fora do horário comercial, altere `MaxSize` para `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 cheduled-weekend-scale-down .config

Este arquivo de configuração instrui o Elastic Beanstalk a reduzir todas as sextas-feiras às 18h. Se você sabe que seu aplicativo não recebe tanto tráfego durante o fim de semana, pode criar uma ação programada semelhante.

```
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
```

Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk

O Amazon EC2 Auto Scaling monitora a integridade de cada instância do Amazon Elastic Compute Cloud (Amazon EC2) que ele lança. Se alguma instância for encerrada inesperadamente, o Auto Scaling detectará o encerramento e executará uma instância de substituição. Por padrão, o grupo Auto Scaling criado para seu ambiente usa as verificações de EC2 status da Amazon. Se uma instância em seu ambiente falhar na verificação de EC2 status da Amazon, o Auto Scaling a desativa e a substitui.

As verificações de EC2 status da Amazon cobrem apenas a integridade de uma instância, não a integridade do seu aplicativo, servidor ou qualquer contêiner Docker em execução na instância. Se a aplicação falhar, mas a instância na qual ela é executada ainda estiver íntegra, ela poderá ser retirada do balanceador de carga, mas não será substituída automaticamente pelo Auto Scaling. O comportamento padrão é adequado para resolução de problemas. Se o Auto Scaling substituir a instância logo após a falha na aplicação, talvez você não perceba que ocorreu um erro, mesmo se ele falhar imediatamente após a inicialização.

Se você quiser que o Auto Scaling substitua instâncias cuja aplicação parou de responder, poderá usar um arquivo de configuração para configurar o grupo de Auto Scaling para usar verificações de integridade do Elastic Load Balancing. O exemplo a seguir define o grupo para usar as verificações

de saúde do balanceador de carga, além da verificação de EC2 status da Amazon, para determinar a integridade de uma instância.

Example .ebextensions/autoscaling.config

```
Resources:
  AWSEBAutoScalingGroup:
    Type: "AWS::AutoScaling::AutoScalingGroup"
    Properties:
      HealthCheckType: ELB
      HealthCheckGracePeriod: 300
```

Para obter mais informações sobre as `HealthCheckGracePeriod` propriedades `HealthCheckType` e, consulte AWS::AutoScaling::AutoScalingGroup no Guia do AWS CloudFormation usuário e Health Checks for Auto Scaling Instances no Amazon Auto EC2 Scaling User Guide.

Por padrão, a verificação de integridade do Elastic Load Balancing é configurada para tentar uma conexão TCP com sua instância na porta 80. Isso confirma que o servidor web em execução na instância está aceitando conexões. No entanto, você pode desejar personalizar a verificação da integridade do load balancer para garantir que seu aplicativo, e não apenas o servidor web, está em bom estado. A configuração do período de carência define por quantos segundos uma instância pode ser reprovada na verificação de integridade sem ser encerrada e substituída. As instâncias podem ser recuperadas depois de serem retiradas do load balancer, portanto, forneça um tempo adequado suficiente para seu aplicativo.

Balanceador de carga do ambiente do Elastic Beanstalk

Um load balancer distribui o tráfego entre as instâncias do ambiente. Quando você ativa o balanceamento de carga, AWS Elastic Beanstalk cria um balanceador de carga do Elastic Load Balancing dedicado ao seu ambiente. O Elastic Beanstalk gerencia totalmente esse balanceador de carga, cuidando das configurações de segurança e encerrando o balanceador de carga quando o ambiente é encerrado.

Também há a opção de compartilhar um balanceador de carga em vários ambientes do Elastic Beanstalk. Com um load balancer compartilhado, você economiza no custo operacional evitando um load balancer dedicado para cada ambiente. Você também assume mais responsabilidade de gerenciamento pelo load balancer compartilhado usado pelos ambientes.

O Elastic Load Balancing tem estes tipos de balanceador de carga:

- Classic Load Balancer: o balanceador de carga da geração anterior. Rotas de tráfego de solicitação HTTP, HTTPS ou TCP para portas diferentes em instâncias de ambiente.
- Application Load Balancer: um balanceador de carga da camada de aplicação. Rotas de tráfego de solicitação HTTP ou HTTPS para portas diferentes em instâncias de ambiente com base no caminho da solicitação.
- Network Load Balancer: um balanceador de carga de camada de rede. Rotas de tráfego de solicitação TCP para portas diferentes em instâncias de ambiente. Oferece suporte a verificações de integridade ativas e passivas.

O Elastic Beanstalk é compatível com todos os três tipos de balanceador de carga. A tabela a seguir mostra quais tipos você pode usar com os dois padrões de uso:

Tipo de load balancer	Dedicado	Compartilhado
Classic Load Balancer	✓ Sim	X Não
Application Load Balancer	✓ Sim	✓ Sim
Network Load Balancer	✓ Sim	X Não

> **Note**
>
> A opção Classic Load Balancer (CLB) está desabilitada no assistente do console Criar ambiente. Se você tiver um ambiente existente configurado com um Classic Load Balancer, poderá criar um novo clonando o ambiente existente usando o console do Elastic Beanstalk ou a CLI do EB. Você também tem a opção de usar o EB CLI ou AWS CLIcriar um novo ambiente configurado com um Classic Load Balancer. Essas ferramentas de linha de comando criarão um novo ambiente com um CLB mesmo que ainda não exista um em sua conta.

Por padrão, o Elastic Beanstalk cria um Application Load Balancer para seu ambiente quando você habilita o balanceamento de carga com o console Elastic Beanstalk ou a CLI do EB. Ele configura o load balancer para ouvir o tráfego HTTP na porta 80 e encaminhar esse tráfego para

instâncias na mesma porta. Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Posteriormente, você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo.

> **Note**
>
> Seu ambiente deve estar em uma VPC com sub-redes em pelo menos duas zonas de disponibilidade para criar um Application Load Balancer. Todas as novas AWS contas incluem contas padrão VPCs que atendem a esse requisito.

Consulte os tópicos a seguir para saber mais sobre cada tipo de balanceador de carga compatível com o Elastic Beanstalk, sua funcionalidade, como configurá-lo e gerenciá-lo em um ambiente do Elastic Balanceador e como configurar um balanceador de carga para fazer upload de logs de acesso para o Amazon S3.

Tópicos

- Configuração de um Classic Load Balancer
- Configuração de um Application Load Balancer
- Configurar um Application Load Balancer compartilhado
- Configurar um Network Load Balancer
- Configurar logs de acesso

Configuração de um Classic Load Balancer

Quando você ativa o balanceamento de carga, seu AWS Elastic Beanstalk ambiente é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias em seu ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o Guia do usuário do Elastic Load Balancing. O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um Classic Load Balancer que o Elastic Beanstalk cria e dedica ao seu ambiente. Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte Balanceador de carga do ambiente do Elastic Beanstalk.

> **Note**
>
> Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Posteriormente, você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo.

Introdução

Um Classic Load Balancer é o balanceador de carga da geração anterior do Elastic Load Balancing. Oferece suporte ao roteamento de tráfego de solicitação HTTP, HTTPS ou TCP para portas diferentes em instâncias de ambiente.

Quando seu ambiente usa um Classic Load Balancer, o Elastic Beanstalk o configura por padrão para ouvir o tráfego HTTP na porta 80 e encaminhá-lo para instâncias na mesma porta. Embora não seja possível excluir o ouvinte padrão da porta 80, você pode desabilitá-lo, o que obtém a mesma funcionalidade ao bloquear o tráfego. Observe que você pode adicionar ou excluir outros receptores. Para oferecer suporte a conexões seguras, configure o load balancer com um listener na porta 443 e um certificado TLS.

O balanceador de carga usa uma verificação de saúde para determinar se as EC2 instâncias da Amazon que executam seu aplicativo estão íntegras. A verificação de integridade faz uma solicitação para um URL especificado em um intervalo de tempo definido. Se o URL retornar uma mensagem de erro ou não retornar dentro do tempo limite especificado, essa verificação falhará.

Se a aplicação tiver melhor performance atendendo a várias solicitações do mesmo cliente em um único servidor, configure o balanceador de carga para usar sticky sessions. Com sessões fixas, o balanceador de carga adiciona um cookie às respostas HTTP que identifica a EC2 instância da Amazon que atendeu à solicitação. Quando uma solicitação subsequente é recebida do mesmo cliente, o load balancer usa o cookie para enviar a solicitação para a mesma instância.

Com o balanceamento de carga entre zonas, cada nó do balanceador de carga do seu Classic Load Balancer distribui solicitações uniformemente a todas as instâncias registradas em todas as zonas de disponibilidade habilitadas. Se o balanceamento de carga entre zonas estiver desabilitado, cada nó do balanceador de carga distribuirá solicitações uniformemente às instâncias registradas somente em sua zona de disponibilidade.

Quando uma instância é removida do balanceador de carga por não ser íntegra ou por redução do ambiente, a diminuição de conexão dá à instância tempo de concluir as solicitações antes de fechar

a conexão entre a instância e o balanceador de carga. Você pode alterar o tempo concedido às instâncias para enviar uma resposta ou pode desabilitar completamente a drenagem de conexão.

> **Note**
>
> A diminuição de conexão é habilitada por padrão quando um ambiente é criado com o console do Elastic Beanstalk ou a CLI do EB. Para os outros clientes, é possível ativá-la com opções de configuração.

Você pode usar configurações avançadas do balanceador de carga para configurar ouvintes em portas arbitrárias, modificar configurações adicionais de sessão fixa e configurar o balanceador de carga para se conectar às instâncias com segurança. EC2 Essas configurações estão disponíveis por meio das opções de configuração que podem ser definidas com os arquivos de configuração no código-fonte ou diretamente em um ambiente usando a API do Elastic Beanstalk. Muitas dessas configurações também estão disponíveis no console do Elastic Beanstalk. Além disso, é possível configurar um balanceador de carga para fazer upload de logs de acesso no Amazon S3.

Configurar um Classic Load Balancer usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar portas, certificado HTTPS e outras configurações de um Classic Load Balancer durante a criação do ambiente ou mais tarde, quando o ambiente estiver em execução.

> **Note**
>
> A opção Classic Load Balancer (CLB) está desabilitada no assistente do console Criar ambiente. Se você tiver um ambiente existente configurado com um Classic Load Balancer, poderá criar um novo clonando o ambiente existente usando o console do Elastic Beanstalk ou a CLI do EB. Você também tem a opção de usar o EB CLI ou AWS CLIcriar um novo ambiente configurado com um Classic Load Balancer. Essas ferramentas de linha de comando criarão um novo ambiente com um CLB mesmo que ainda não exista um em sua conta.

Como configurar o Classic Load Balancer de um ambiente em execução no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS

2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

> **Note**
>
> Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer. Para aprender como configurar um, consulte Alterar tipo de ambiente.

5. Faça as alterações de configuração do Classic Load Balancer necessárias ao seu ambiente.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurações do Classic Load Balancer

- Listeners
- Sessões
- Balanceamento de carga entre zonas
- Drenagem da conexão
- Verificação de integridade

Listeners

Use esta lista para especificar listeners para seu load balancer. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada usando um protocolo especificado para suas instâncias. Inicialmente, a lista mostra o listener padrão, que roteia o tráfego HTTP de entrada na porta 80 para os servidores de instância do seu ambiente que estão observando o tráfego HTTP na porta 80.

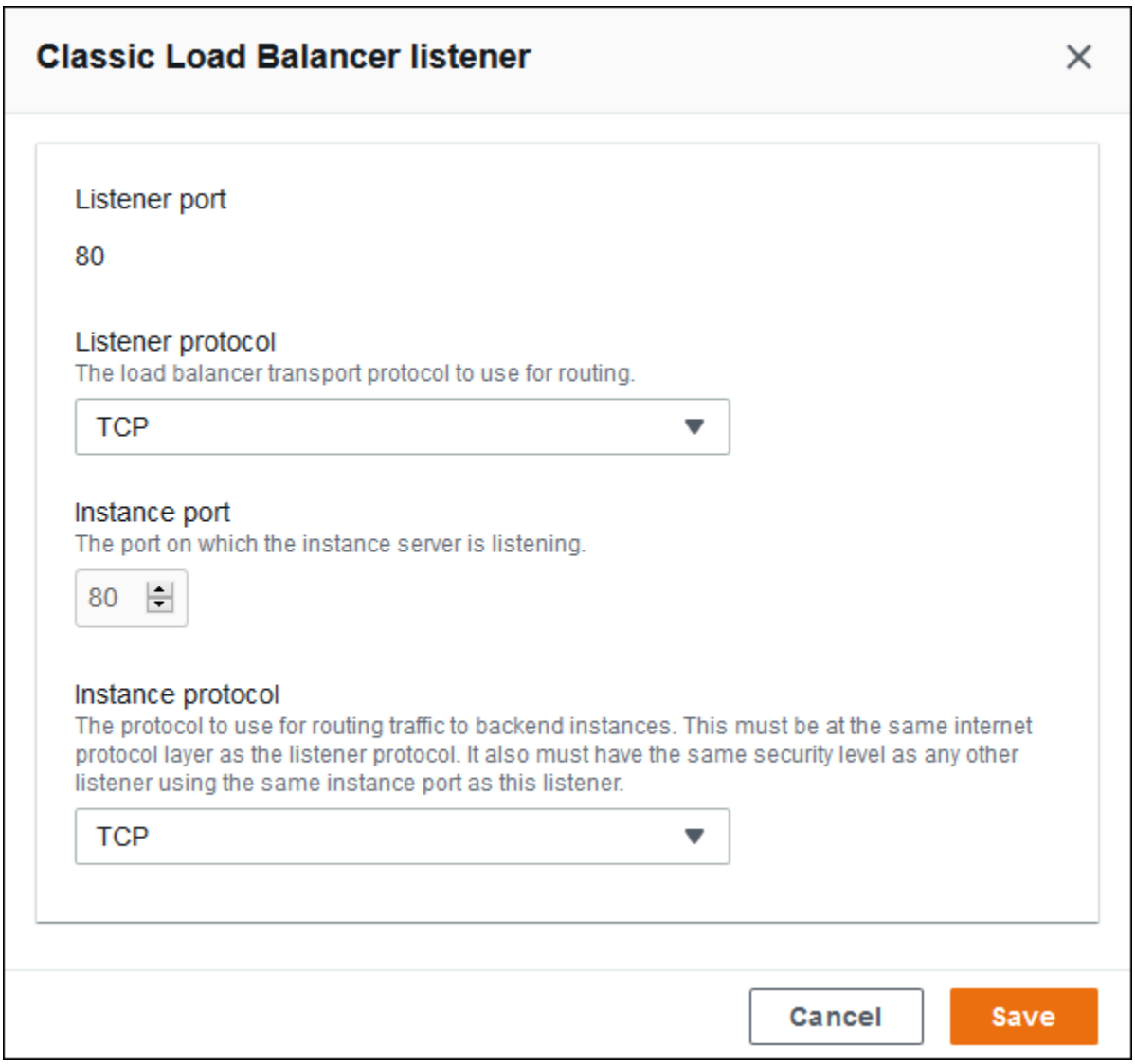
> **Note**
>
> Embora não seja possível excluir o ouvinte padrão da porta 80, você pode desabilitá-lo, o que obtém a mesma funcionalidade ao bloquear o tráfego.



Para configurar um listener existente

1. Marque a caixa de seleção ao lado da entrada da tabela, escolha Actions, e depois escolha a ação desejada.
2. Se escolher Edit, use a caixa de diálogo Classic Load Balancer listener para editar as configurações e, em seguida, escolha Save.

Por exemplo, você pode editar o listener padrão e alterar Protocol de HTTP para TCP se você quiser que o load balancer encaminhe uma solicitação como está. Isso impede que o load balancer reescreva cabeçalhos (incluindo `X-Forwarded-For`). A técnica não funciona com sticky sessions.



Para adicionar um listener

1. Escolha Add listener.
2. Na caixa de diálogo Classic Load Balancer listener, defina as configurações desejadas e, em seguida, escolha Add.

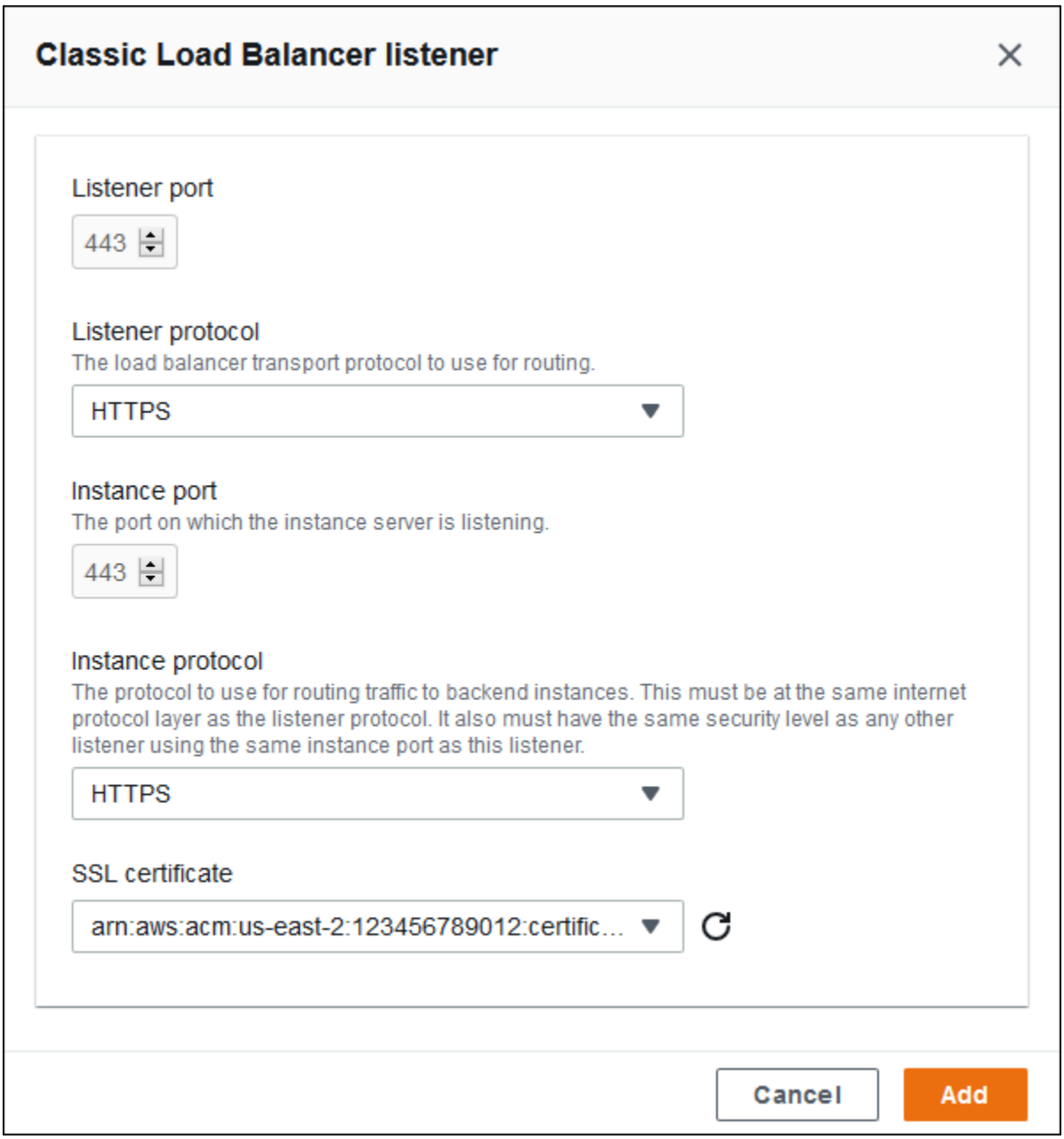
Adicionar um listener seguro é um caso de uso comum. O exemplo na imagem a seguir adiciona um listener ao tráfego HTTPS na porta 443. Esse listener direciona o tráfego de entrada para os servidores de instância do ambiente que observam o tráfego HTTPS na porta 443.

Antes de configurar um listener HTTPS, verifique se você tem um certificado SSL válido. Execute um destes procedimentos:

- Se AWS Certificate Manager (ACM) estiver disponível em sua AWS região, crie ou importe um certificado usando o ACM. Para obter mais informações sobre como solicitar um certificado do ACM, consulte Solicitação de um certificado no Manual do usuário do AWS Certificate Manager . Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte Importação de certificados no Manual do usuário do AWS Certificate Manager .
- Se o ACM não estiver disponível na sua AWS região, faça o upload do certificado e da chave existentes para o IAM. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte Trabalhar com certificados de servidor no Guia do usuário do IAM.

Para obter mais detalhes sobre como configurar o HTTPS e trabalhar com os certificados no Elastic Beanstalk, consulte Configurar HTTPS para o seu ambiente do Elastic Beanstalk.

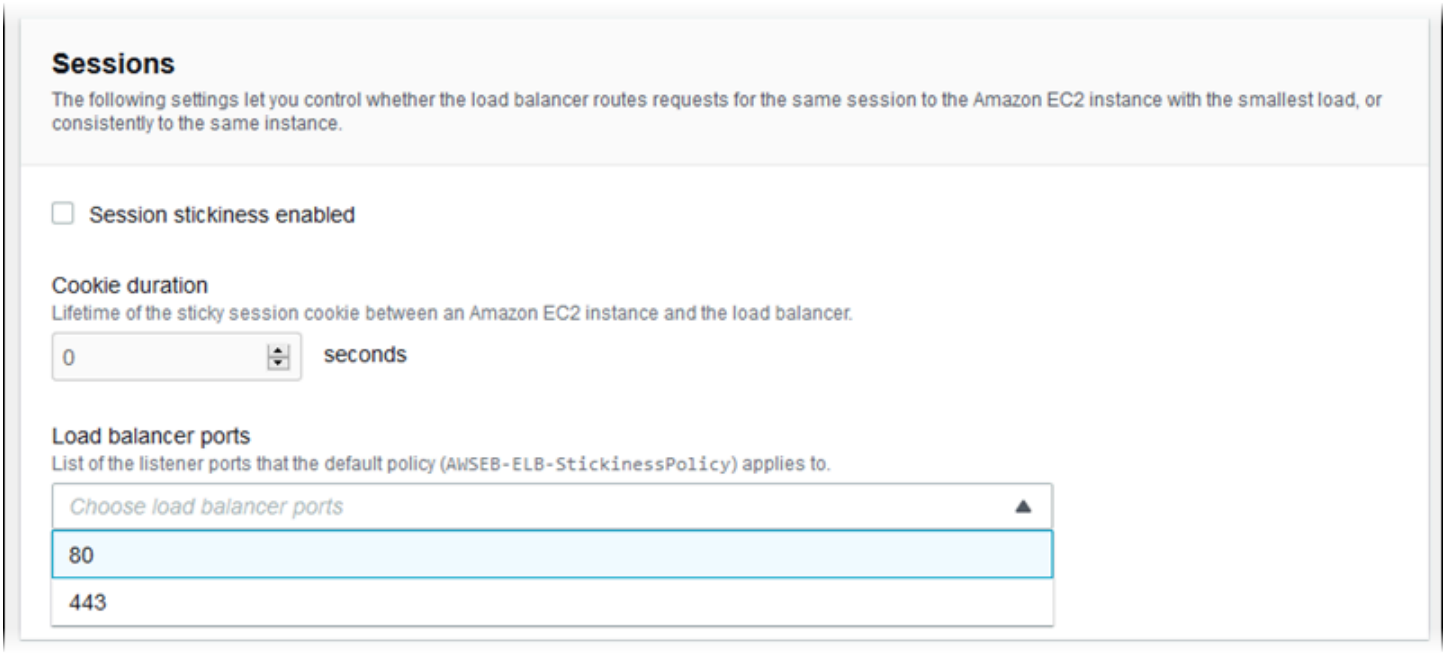
Para SSL certificate, escolha o ARN do seu certificado SSL. Por exemplo, `arn:aws:iam::123456789012:server-certificate/abc/certs/build` ou `arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`.



Para obter detalhes sobre como configurar o HTTPS e trabalhar com os certificados no Elastic Beanstalk, consulte Configurar HTTPS para o seu ambiente do Elastic Beanstalk.

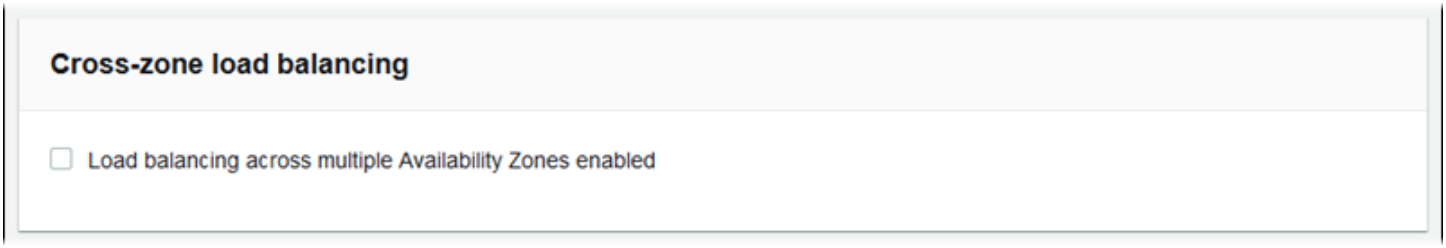
Sessões

Selecione ou desmarque a caixa Perdurabilidade da sessão habilitadapara habilitar ou desabilitar as sticky sessions. Use Duração do cookie para configurar a duração de uma sticky session até **`1000000`** segundos. Na lista Load balancer ports (Portas do load balancer), selecione portas do listener às quais a política padrão (`AWSEB-ELB-StickinessPolicy`) se aplica.



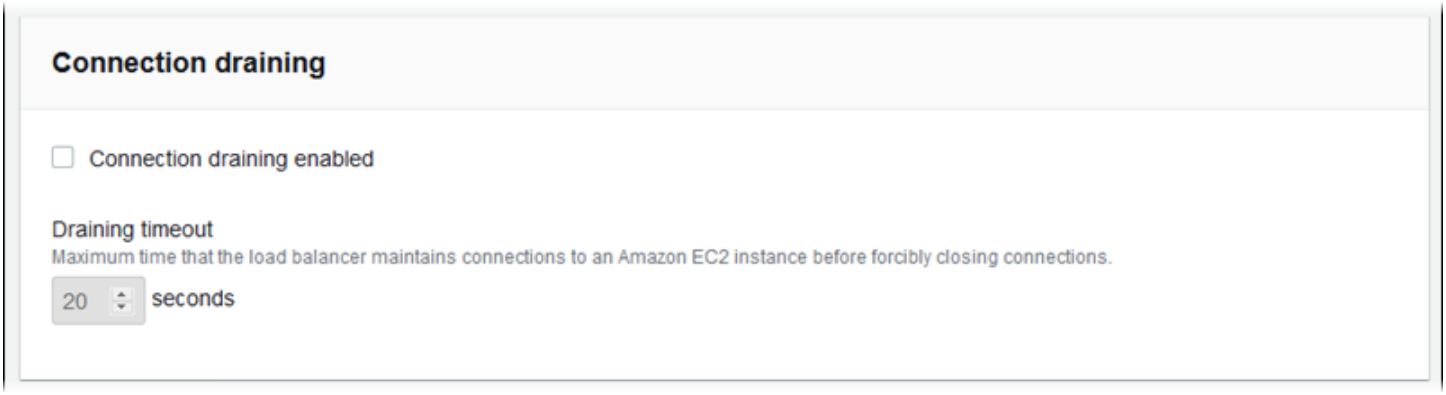
Balanceamento de carga entre zonas

Selecione ou desmarque a caixa Load balancing across multiple Availability Zones enabled para habilitar ou desabilitar o balanceamento de carga entre zonas.



Drenagem da conexão

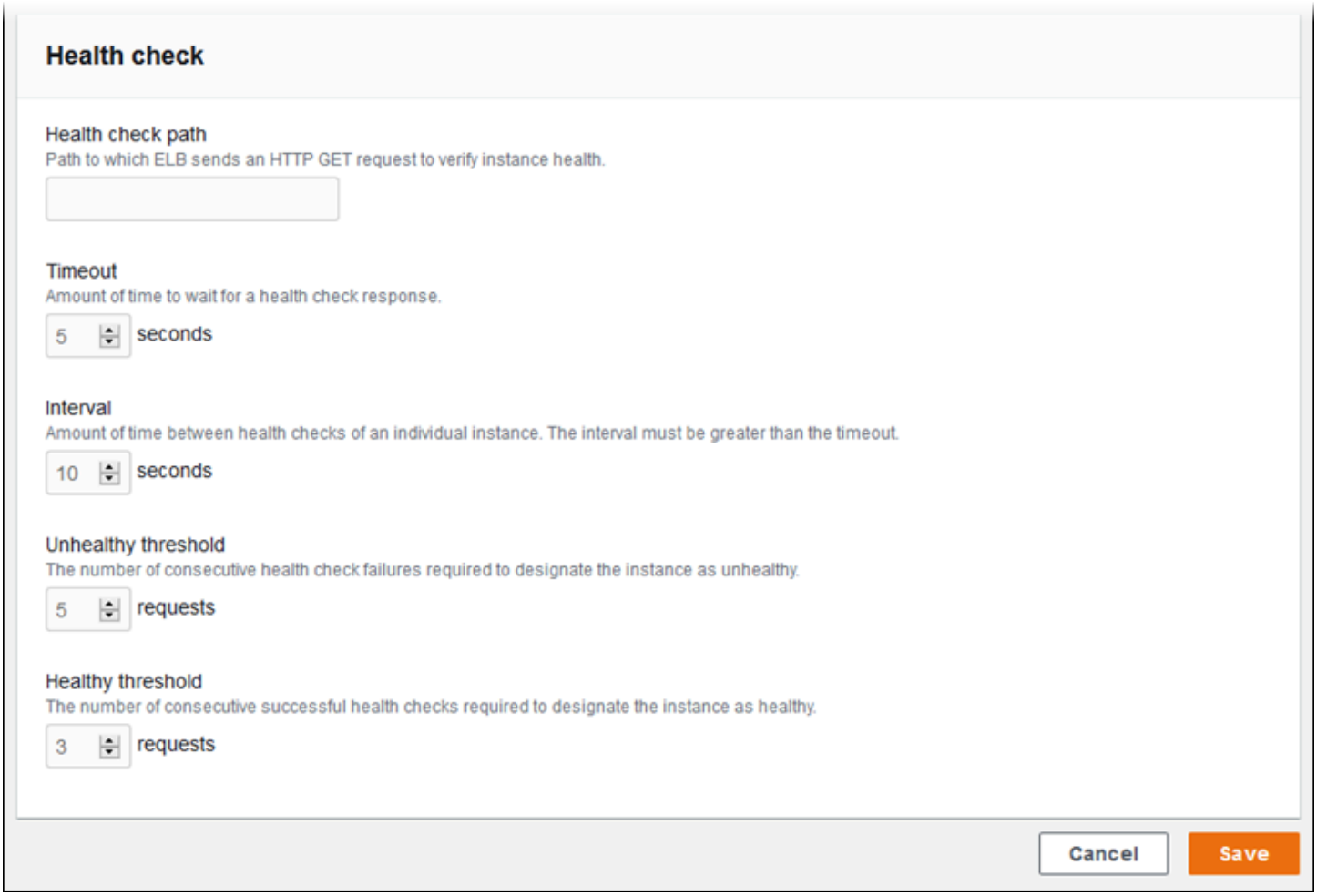
Selecione ou desmarque a caixa Diminuição de conexão habilitada para habilitar ou desabilitar a diminuição de conexão. Defina Tempo limite de drenagem até **`3600`** segundos.



Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do load balancer:

- Health check path (Caminho da verificação de integridade): o caminho para o qual o balanceador de carga envia solicitações de verificação de integridade. Se você não definir o caminho, o load balancer tentará fazer uma conexão TCP na porta 80 para verificar a integridade.
- Timeout (Tempo limite): a quantidade de tempo, em segundos, para aguardar uma resposta de verificação de integridade.
- Interval (Intervalo): a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual. O intervalo deve ser maior que o tempo limite.
- Unhealthy threshold (Limite não íntegro), Healthy threshold (Limite íntegro): o número de verificações de integridade que devem falhar ou passar, respectivamente, antes que o Elastic Load Balancing altere o estado de integridade de uma instância.



> **Note**
>
> A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de saúde do Elastic Load Balancing não são substituídas automaticamente pelo Amazon Auto EC2 Scaling, a menos que você configure manualmente o Amazon Auto EC2 Scaling para fazer isso. Para mais detalhes, consulte Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk.

Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte Relatórios de integridade básica.

Configurar um Classic Load Balancer usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o eb create.

```
$ 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):
```

Pressione Enter para selecionar `classic`.

Você também pode especificar um tipo de load balancer usando a opção `--elb-type`.

```
$ eb create test-env --elb-type classic
```

Namespaces de configuração do Classic Load Balancer

É possível encontrar configurações relacionadas aos Classic Load Balancers nos seguintes namespaces:

- `aws:elb:healthcheck`: configure os limites, o intervalo de verificação e o tempo limite das verificações de integridade do balanceador de carga.
- `aws:elasticbeanstalk:application`: configure o URL da verificação de integridade.
- `aws:elb:loadbalancer`: habilitar balanceamento de carga entre zonas. Atribua security groups ao load balancer e substitua o security group padrão que o Elastic Beanstalk cria. Esse namespace também inclui opções obsoletas de configuração de ouvintes padrão e seguros que foram substituídas pelas opções no namespace `aws:elb:listener`.
- `aws:elb:listener`: configure o listener padrão na porta 80, um listener seguro na porta 443 ou outros listeners para qualquer protocolo em qualquer porta. Se você especificar `aws:elb:listener` como o namespace, as configurações se aplicarão ao listener padrão na porta 80. Se você especificar uma porta (por exemplo, `aws:elb:listener:443`), um listener será configurado nessa porta.
- `aws:elb:policies`: defina configurações adicionais para o balanceador de carga. Use as opções nesse namespace para configurar ouvintes em portas arbitrárias, modificar configurações adicionais de sessão fixa e configurar o balanceador de carga para se conectar com segurança às instâncias da Amazon. EC2

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Example .ebextensions/loadbalancer-terminatehttps.config

O exemplo de arquivo de configuração a seguir cria um listener HTTPS na porta 443, atribui um certificado que o load balancer usa para encerrar a conexão segura e desabilita o listener padrão na porta 80. O balanceador de carga encaminha as solicitações descriptografadas para as EC2 instâncias em seu ambiente em HTTP:80.

```
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
```

Configuração de um Application Load Balancer

Quando você ativa o balanceamento de carga, seu AWS Elastic Beanstalk ambiente é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias em seu ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o Guia do usuário do Elastic Load Balancing. O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um Application Load Balancer que o Elastic Beanstalk cria e dedica ao seu ambiente. Consulte também the section called “Application Load Balancer compartilhado”. Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte the section called “Load balancer”.

> **Note**
>
> Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo. Também não é possível alternar de um load balancer dedicado para um compartilhado ou vice-versa.

Introdução

Um Application Load Balancer inspeciona o tráfego na camada de protocolo de rede da aplicação para identificar o caminho do pedido de modo que ele possa direcionar solicitações de caminhos diferentes para destinos diferentes.

Quando seu ambiente usa um Application Load Balancer, o Elastic Beanstalk o configura por padrão para executar a mesma função que um Classic Load Balancer. O listener padrão aceita solicitações de HTTP na porta 80 e as distribui para as instâncias no ambiente. É possível adicionar um listener seguro na porta 443 com um certificado para descriptografar o tráfego de HTTPS, configurar o comportamento de verificação de integridade e enviar os registros de acesso do balanceador de carga para um bucket do Amazon Simple Storage Service (Amazon S3).

> **Note**
>
> Diferentemente de um Classic Load Balancer ou de um Network Load Balancer, um Application Load Balancer não pode ter TCP ou ouvintes na camada de transporte (camada

4). SSL/TLS Suporta apenas listeners HTTP e HTTPS. Além disso, não é possível usar a autenticação de back-end para autenticar conexões HTTPS entre o load balancer e as instâncias de back-end.

Em um ambiente do Elastic Beanstalk, é possível usar um Application Load Balancer para direcionar o tráfego de determinados caminhos para um processo diferente em suas instâncias do servidor Web. Com um Classic Load Balancer, todo o tráfego para o listener é roteado para um único processo nas instâncias de back-end. Com um Application Load Balancer, é possível configurar várias regras no listener para rotear as solicitações para determinados caminhos para um processo de back-end diferente. Configure cada processo com a porta na qual o processo escuta.

Por exemplo, é possível executar um processo de login separadamente do aplicativo principal. Embora o aplicativo principal nas instâncias do seu ambiente aceite a maioria das solicitações e observe a porta 80, seu processo de login observará a porta 5000 e aceitará solicitações para o caminho `/login`. Todas as solicitações de entrada dos clientes vêm na porta 80. Com um Application Load Balancer, é possível configurar um único listener para o tráfego de entrada na porta 80, com duas regras que direcionam o tráfego para dois processos separados, dependendo do caminho na solicitação. Adicione uma regra personalizada que direcione o tráfego para `/login` para o processo de logon que escuta na porta 5000. A regra padrão encaminha todos os outros tráfegos para o processo principal do aplicativo, observando a porta 80.

Uma regra do Application Load Balancer mapeia uma solicitação para um grupo de destino. No Elastic Beanstalk, um grupo de destino é representado por um processo. Você pode configurar um processo com configurações de protocolo, porta e verificação de integridade. O processo representa a execução nas instâncias em seu ambiente. O processo padrão é um listener na porta 80 do proxy reverso (nginx ou Apache) que é executado na frente de seu aplicativo.

> **Note**
>
> Fora do Elastic Beanstalk, um grupo de destino é mapeado para um grupo de instâncias. Um listener pode usar regras e grupos de destino para rotear o tráfego para instâncias diferentes com base no caminho. Dentro do Elastic Beanstalk, todas as instâncias no ambiente são idênticas, então a distinção é feita entre os processos escutando em portas diferentes.

Um Classic Load Balancer usa um único caminho de verificação de integridade para todo o ambiente. Com um Application Load Balancer, cada processo tem um caminho de verificação

de integridade separado que é monitorado pelo balanceador de carga e pelo monitoramento de integridade aprimorado do Elastic Beanstalk.

Para usar o Application Load Balancer, seu ambiente deve estar em uma VPC personalizada ou padrão e deve ter uma função de serviço com o conjunto padrão de permissões. Se você tiver uma função de serviço mais antiga, talvez seja necessário atualizar as permissões dela para incluir `elasticloadbalancing:DescribeTargetHealth` e `elasticloadbalancing:DescribeLoadBalancers`. Para obter mais informações sobre os Application Load Balancers, consulte O que é Application Load Balancer?.

> **Note**
>
> A verificação de integridade do Application Load Balancer não usa o caminho da verificação de integridade do Elastic Beanstalk. Em vez disso, ele usa o caminho específico configurado para cada processo separadamente.

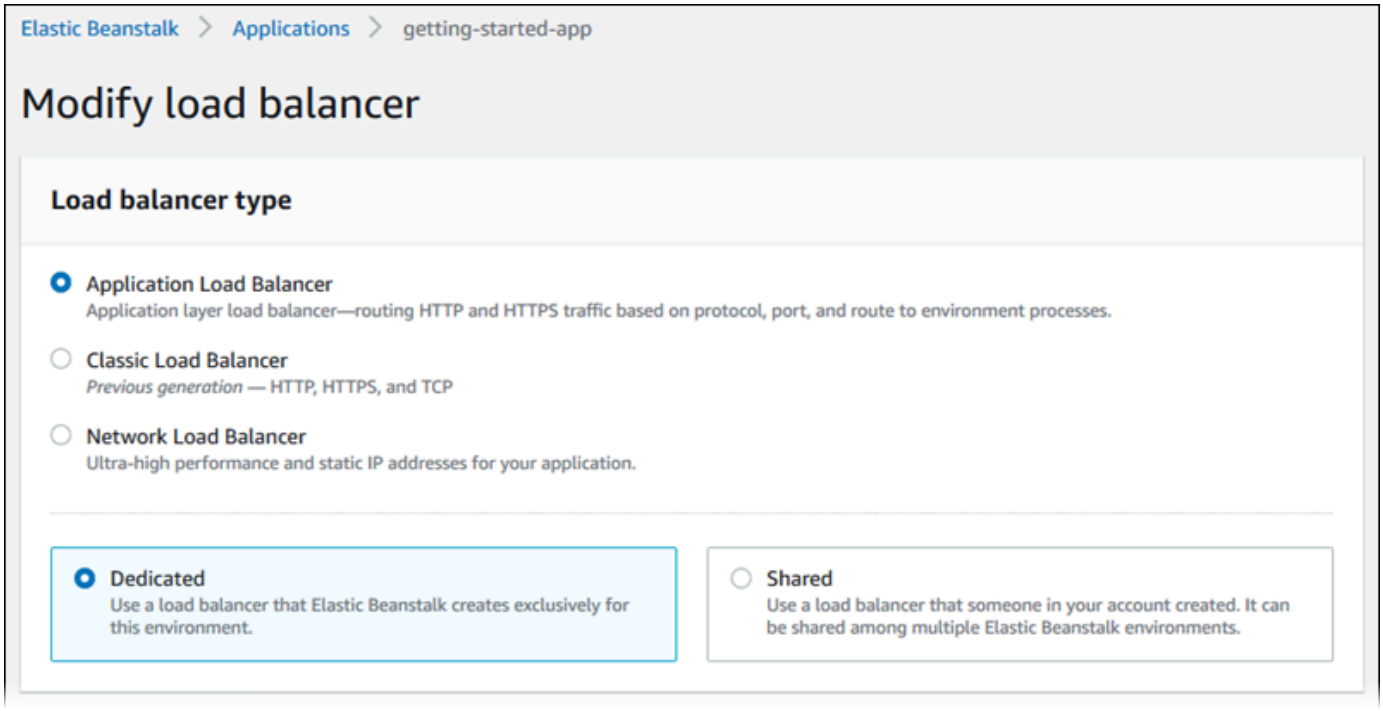
Configurar um Application Load Balancer usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar os listeners, processos e regras de um Application Load Balancer durante a criação do ambiente ou posterior quando seu ambiente estiver em execução.

Como configurar um Application Load Balancer no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Ambientes.
3. Escolha Create a new environment (Criar um ambiente) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Escolha a predefinição de configuração High availability (Alta disponibilidade).

   Se preferir, na categoria de configuração Capacity (Capacidade), configure o tipo de ambiente Load balanced (Carga equilibrada). Para obter detalhes, consulte Capacidade.
6. Na categoria de configuração Load balancer, escolha Edit (Editar).
7. Selecione as opções Application Load Balancer e Dedicated (Dedicado) se elas ainda não estiverem selecionadas.



8. Faça todas as alterações de configuração do Application Load Balancer necessárias ao seu ambiente.
9. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
10. Selecione Criar ambiente.

Como configurar o Application Load Balancer de um ambiente em execução no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

> **Note**
>
> Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer. Para aprender como configurar um, consulte Alterar tipo de ambiente.

5. Faça as alterações de configuração do Application Load Balancer necessárias ao seu ambiente.

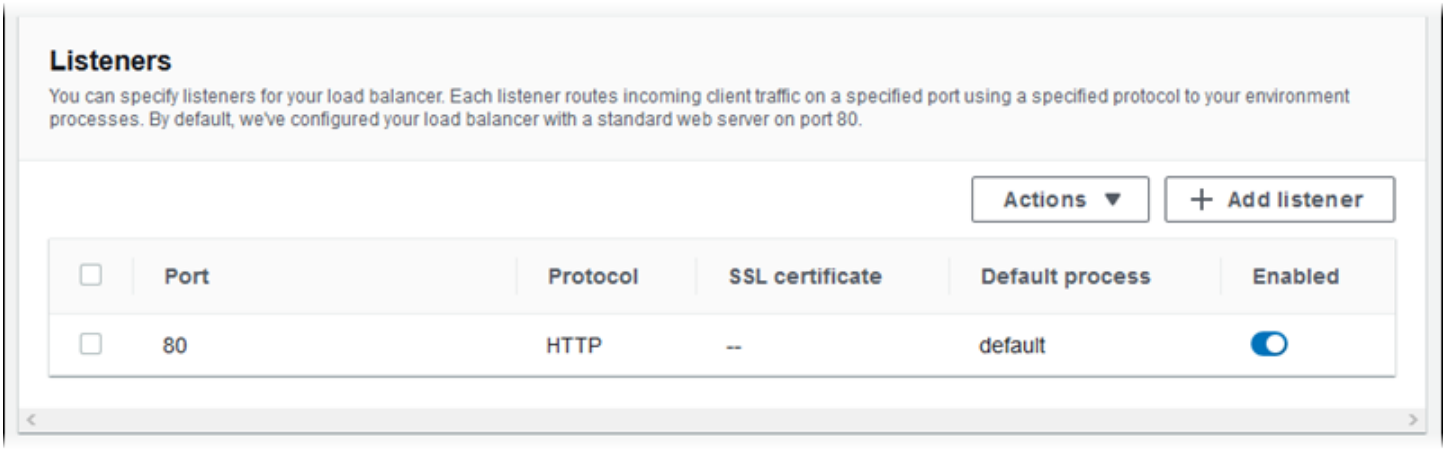
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurações do Application Load Balancer

- Listeners
- Processos
- Regras
- Captura de log de acesso

Listeners

Use esta lista para especificar listeners para seu load balancer. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada usando um protocolo especificado para um ou mais processos nas suas instâncias. Inicialmente, a lista mostra o listener padrão, que encaminha o tráfego HTTP de entrada na porta 80 para um processo chamado padrão.



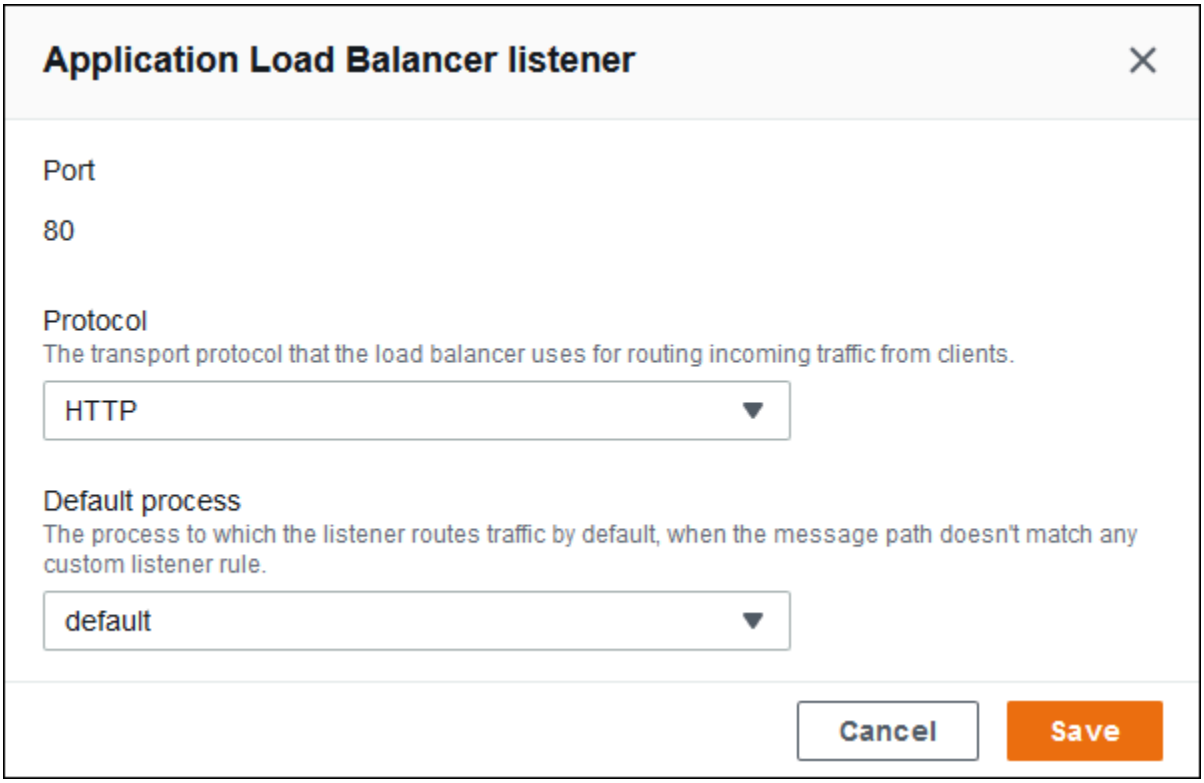
Para configurar um listener existente

1. Marque a caixa de seleção ao lado da entrada da tabela e, em seguida, escolha Actions, Edit.
2. Use a caixa de diálogo Application Load Balancer listener para editar as configurações e, em seguida, escolha Save.

Para adicionar um listener

1. Escolha Add listener.
2. Na caixa de diálogo Application Load Balancer listener (Listener do Application Load Balancer), defina as configurações desejadas e selecione Add (Adicionar).

Use as configurações da caixa de diálogo Listener do Application Load Balancer para escolher a porta e o protocolo nos quais o ouvinte escuta o tráfego, bem como o processo para encaminhar o tráfego. Se você escolher o protocolo HTTPS, defina as configurações de SSL.



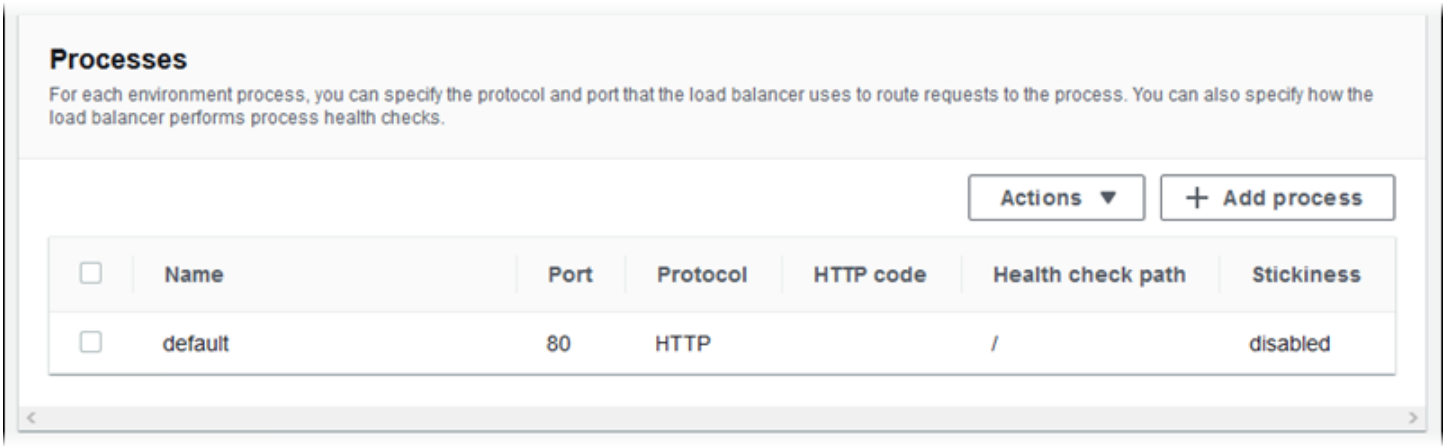
Antes de configurar um listener HTTPS, verifique se você tem um certificado SSL válido. Execute um destes procedimentos:

- Se AWS Certificate Manager (ACM) estiver disponível em sua AWS região, crie ou importe um certificado usando o ACM. Para obter mais informações sobre como solicitar um certificado do ACM, consulte Solicitação de um certificado no Manual do usuário do AWS Certificate Manager . Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte Importação de certificados no Manual do usuário do AWS Certificate Manager .
- Se o ACM não estiver disponível na sua AWS região, faça o upload do certificado e da chave existentes para o IAM. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte Trabalhar com certificados de servidor no Guia do usuário do IAM.

Para obter mais detalhes sobre como configurar o HTTPS e trabalhar com os certificados no Elastic Beanstalk, consulte Configurar HTTPS para o seu ambiente do Elastic Beanstalk.

Processos

Use esta lista para especificar processos para seu load balancer. Um processo é um destino para os listeners direcionarem o tráfego. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada usando um protocolo especificado para um ou mais processos nas suas instâncias. Inicialmente, a lista mostra o processo padrão, que observa o tráfego HTTP de entrada na porta 80.



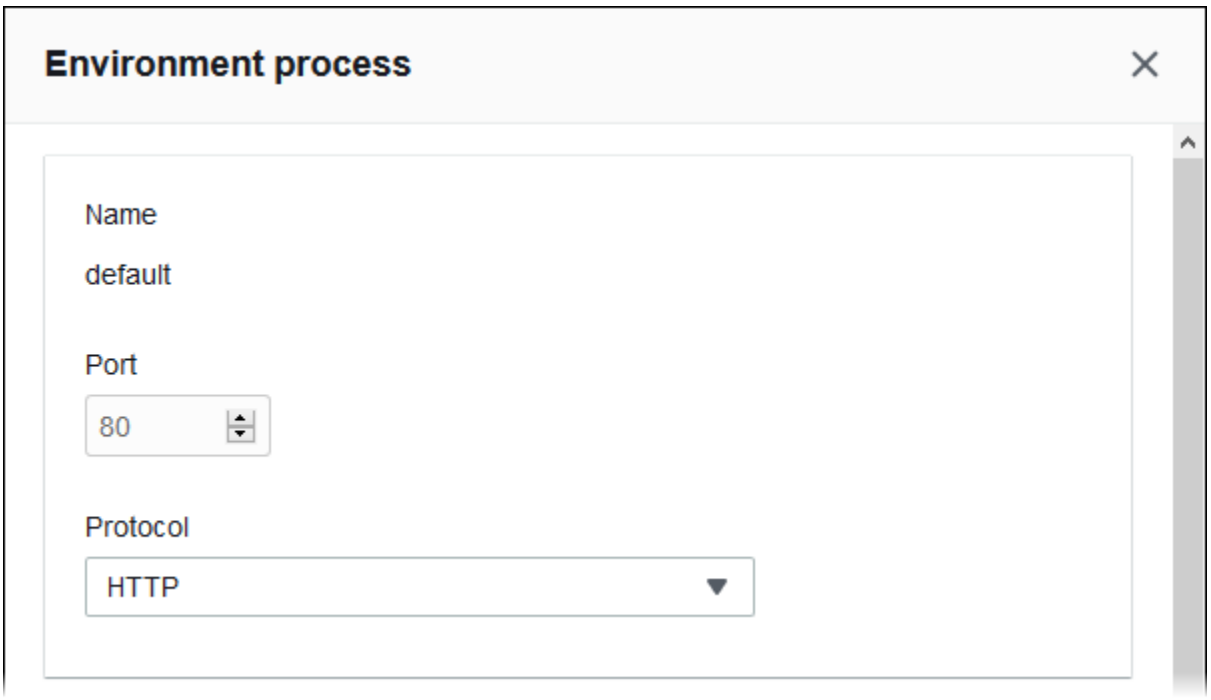
Você pode editar as configurações de um processo existente ou adicionar um novo processo. Para começar a editar um processo na lista ou adicionar um processo a ele, use as mesmas etapas listadas para lista de listener. A caixa de diálogo Environment process é exibida.

Configurações da caixa de diálogo Environment process (Processo do ambiente) do Application Load Balancer

- Definição
- Verificação de integridade
- Sessões

Definição

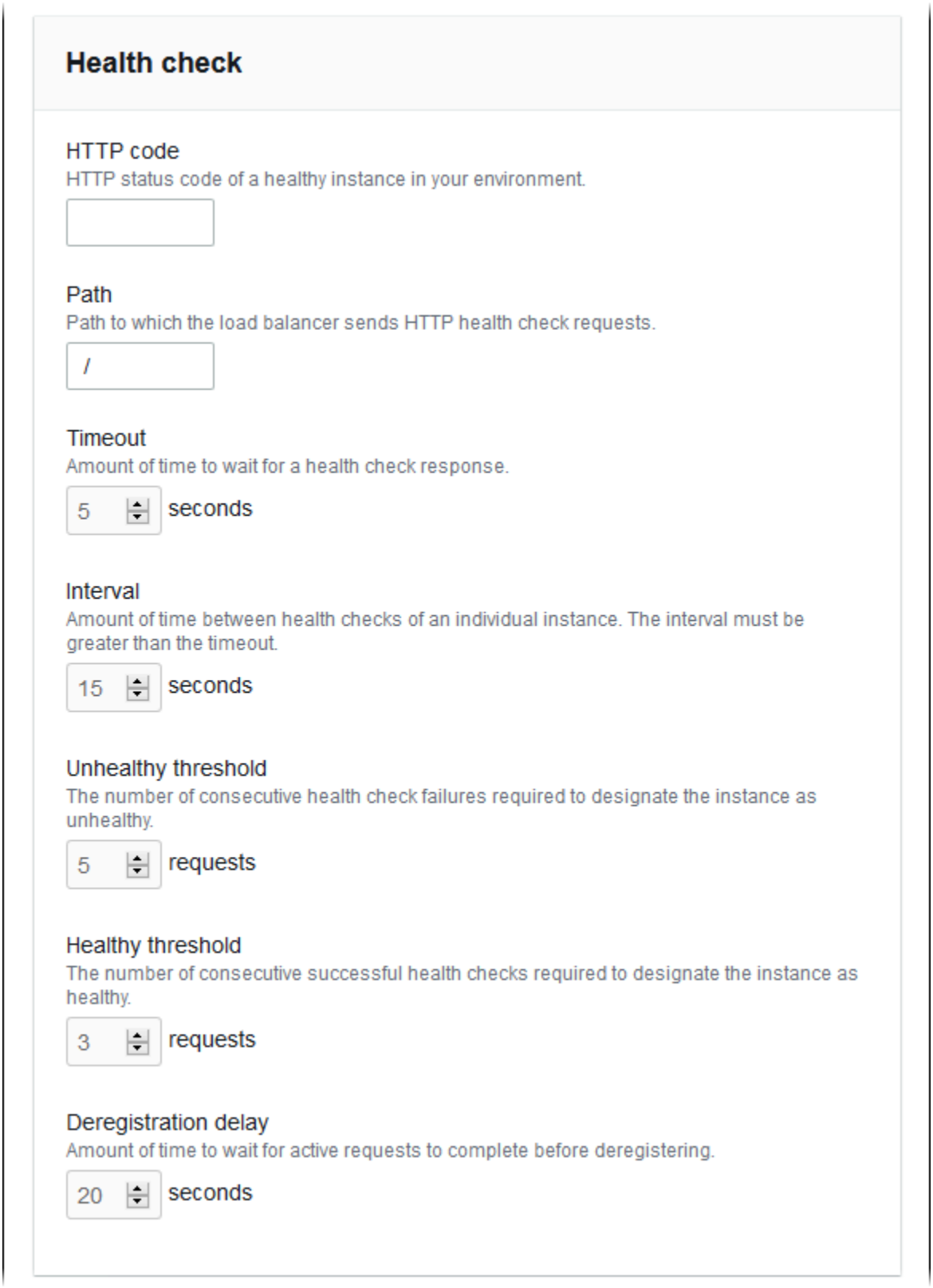
Use estas configurações para definir o processo Name, Port e Protocol em que as solicitações são observadas.



Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do processo:

- HTTP code (Código de HTTP): o código de status HTTP que designa um processo íntegro.
- Path (Caminho): o caminho da solicitação de verificação de integridade para o processo.
- Timeout (Tempo limite): a quantidade de tempo, em segundos, para aguardar uma resposta de verificação de integridade.
- Interval (Intervalo): a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual. O intervalo deve ser maior que o tempo limite.
- Unhealthy threshold (Limite não íntegro), Healthy threshold (Limite íntegro): o número de verificações de integridade que devem falhar ou passar, respectivamente, antes que o Elastic Load Balancing altere o estado de integridade de uma instância.
- Deregistration delay (Atraso do cancelamento de registro): a quantidade de tempo, em segundos, para aguardar que as solicitações ativas sejam concluídas antes de cancelar o registro de uma instância.

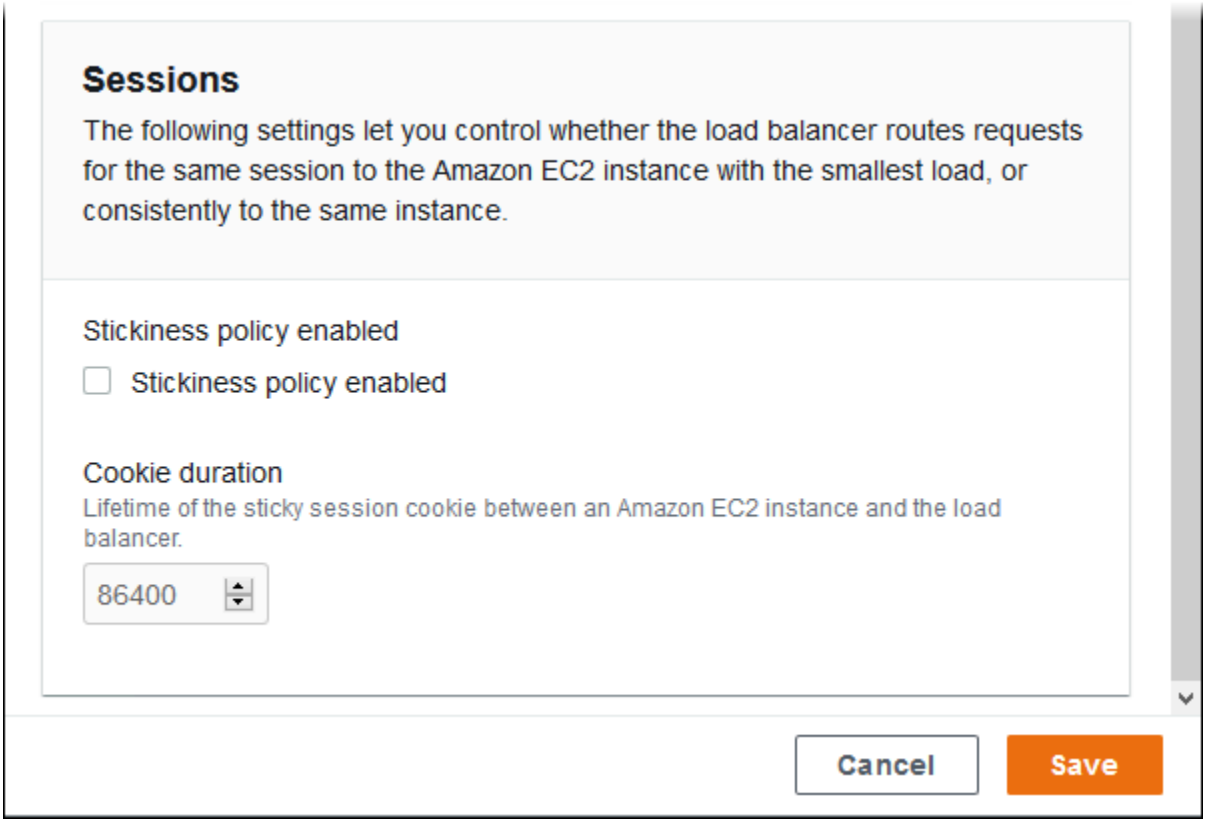

Health check
HTTP code
HTTP status code of a healthy instance in your environment.
Path
Path to which the load balancer sends HTTP health check requests.
/
Timeout
Amount of time to wait for a health check response.
5 seconds
Interval
Amount of time between health checks of an individual instance. The 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**
>
> A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de saúde do Elastic Load Balancing não são substituídas automaticamente pelo Amazon Auto EC2 Scaling, a menos que você configure manualmente o Amazon Auto EC2 Scaling para fazer isso. Para mais detalhes, consulte Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk.

Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte Relatórios de integridade básica.

Sessões

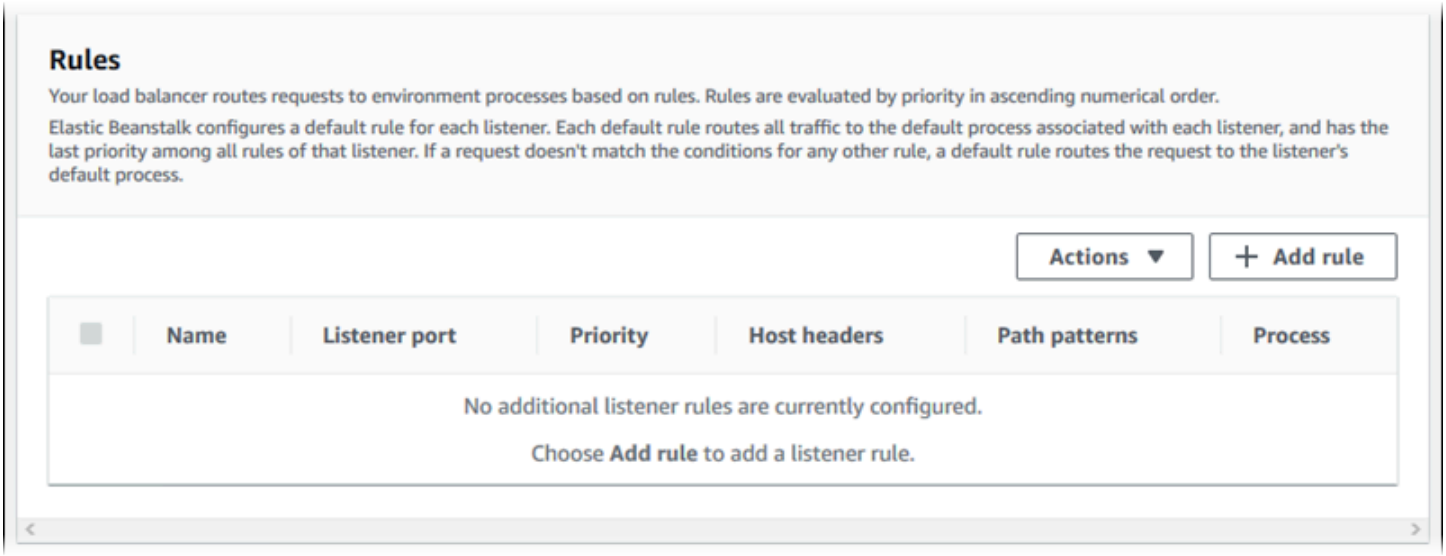
Selecione ou desmarque a caixa Política de perdurabilidade habilitada para habilitar ou desabilitar as sticky sessions. Use Duração do cookie para configurar a duração de uma sticky session até **`604800`** segundos.



Regras

Use esta lista para especificar regras de listener para seu load balancer. Uma regra mapeia solicitações que o listener recebe em um padrão de caminho específico para um processo de destino. Cada listener pode ter várias regras, roteando solicitações em diferentes caminhos para diferentes processos nas suas instâncias.

As regras têm prioridades numéricas que determinam a precedência em que são aplicadas às solicitações recebidas. Para cada novo listener adicionado, o Elastic Beanstalk adiciona uma regra padrão que direciona todo o tráfego do listener para o processo padrão. A precedência da regra padrão é a mais baixa. Ela é aplicada se nenhuma outra regra para o mesmo listener corresponder à solicitação recebida. Inicialmente, se você não tiver adicionado regras personalizadas, a lista estará vazia. As regras padrão de todos os listeners não são exibidas.



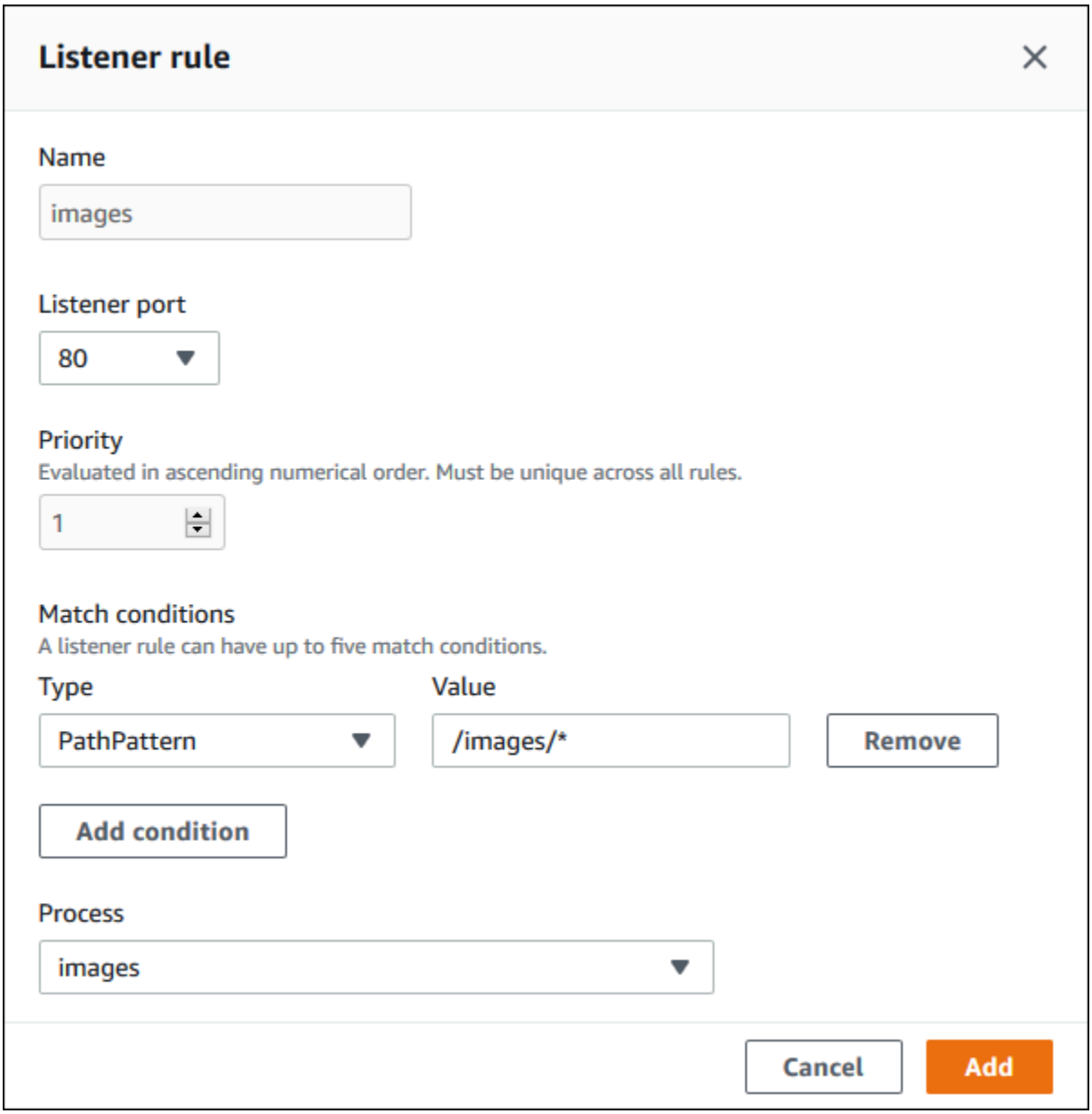
Você pode editar as configurações de uma regra existente ou adicionar uma nova regra. Para começar a editar uma regra na lista ou adicionar uma regra a ela, use as mesmas etapas listadas para lista de listener. A caixa de diálogo Listener rule é exibida com as seguintes configurações:

- Name (Nome): o nome da regra.
- Listener port (Porta do listener): porta do listener ao qual a regra se aplica.
- Priority (Prioridade): prioridade da regra. Um número de prioridade mais baixa tem maior precedência. As prioridades das regras de um listener devem ser únicas.
- Match conditions (Condições de correspondência): uma lista de condições do URL de solicitação às quais a regra se aplica. Há dois tipos de condições: HostHeader(a parte do domínio do URL) e

PathPattern(a parte do caminho do URL). É possível adicionar até cinco condições. Cada valor de condição tem até 128 caracteres de comprimento e pode incluir caracteres curinga.

- Process (Processo): o processo para o qual o balanceador de carga direciona solicitações que correspondem à regra.

Ao editar qualquer regra existente, você não pode alterar Name nem Listener port.



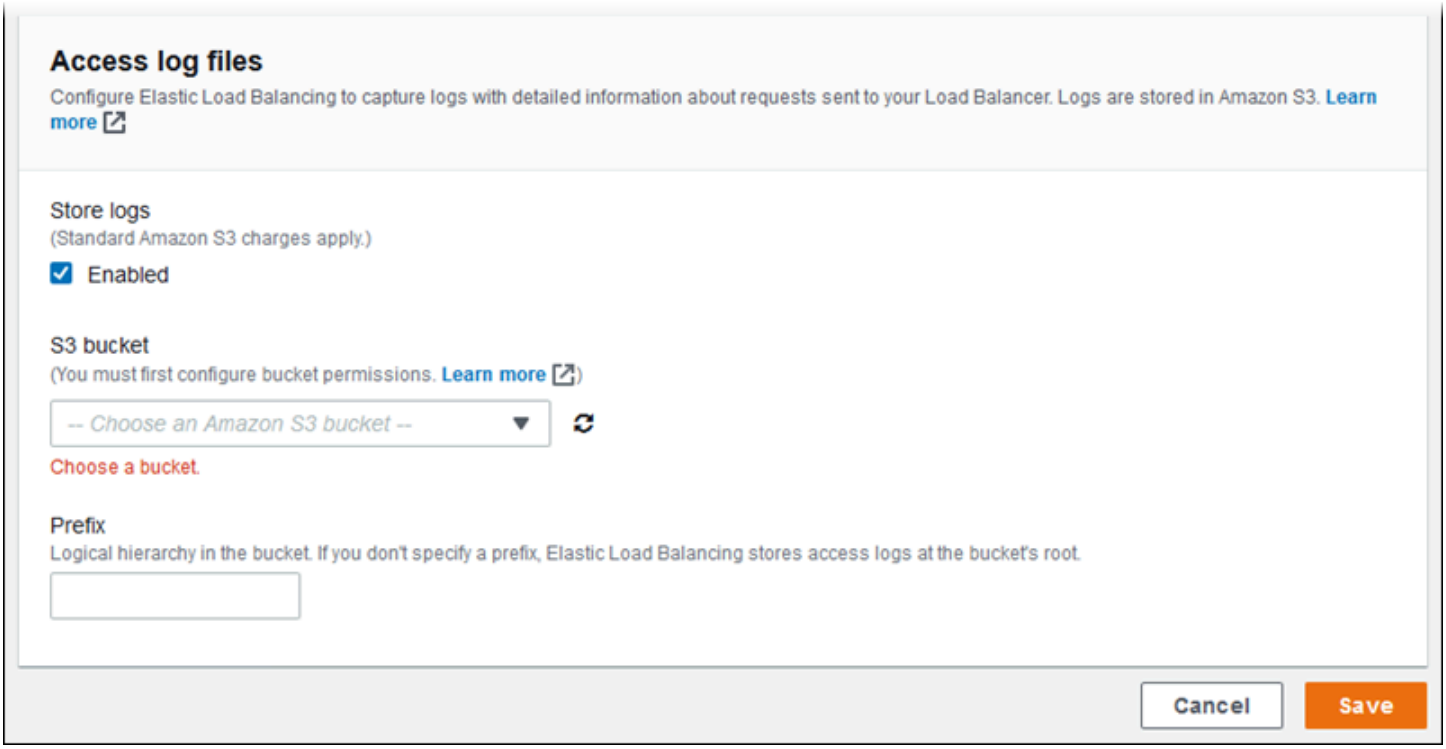
Captura de log de acesso

Use essas configurações para configurar o Elastic Load Balancing para capturar logs com informações detalhadas sobre solicitações enviadas ao Application Load Balancer. A captura de log

de acesso está desativada por padrão. Quando Store logs (Armazenar logs) está habilitado, o Elastic Load Balancing armazena os logs no S3 bucket (Bucket do S3) configurado. A configuração Prefix (Prefixo) especifica uma pasta de nível superior no bucket para os logs. O Elastic Load Balancing coloca os logs em uma pasta denominada `AWSLogs` sob seu prefixo. Se você não especificar um prefixo, o Elastic Load Balancing coloca sua pasta no nível raiz do bucket.

> **Note**
>
> Se o bucket do Amazon S3 que você configura para captura de log de acesso não for o bucket que o Elastic Beanstalk criou para sua conta, certifique-se de adicionar uma política de usuário com as permissões apropriadas para seus usuários (IAM). AWS Identity and Access Management As políticas de usuário gerenciadas que o Elastic Beanstalk fornece apenas permissões de cobertura para recursos gerenciados pelo Elastic Beanstalk.

Para obter detalhes sobre logs de acesso, incluindo permissões e outros requisitos, consulte Registros de acesso para o Application Load Balancer.



Exemplo: Application Load Balancer com um listener seguro e dois processos

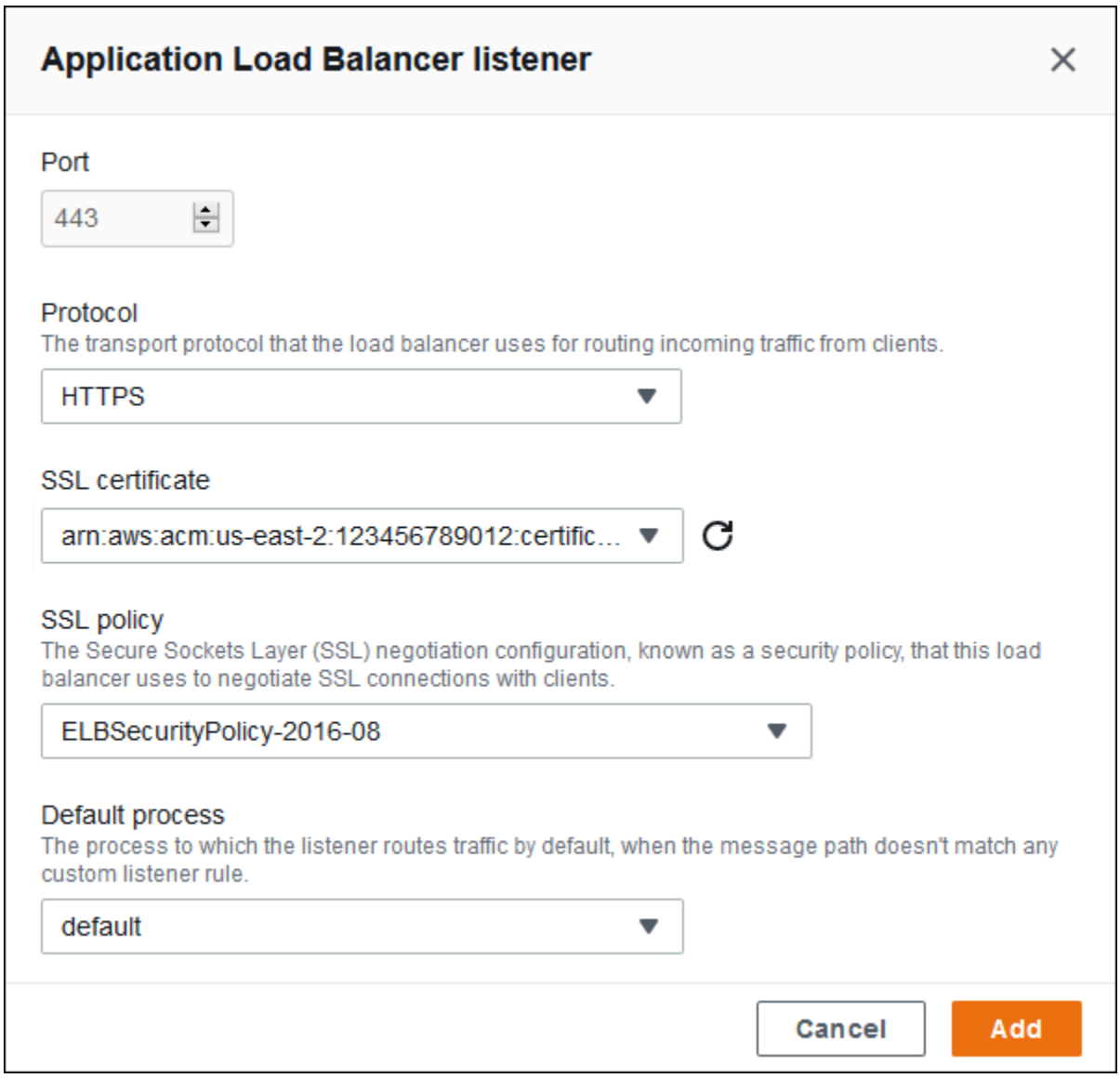
Neste exemplo, seu aplicativo exige criptografia de end-to-end tráfego e um processo separado para lidar com solicitações administrativas.

Para configurar o Application Load Balancer do ambiente de modo a atender esses requisitos, remova o listener padrão, adicione um listener HTTPS, indique que o processo padrão escuta a porta 443 em HTTPS e adicione um processo e uma regra de listener ao tráfego administrativo em um caminho diferente.

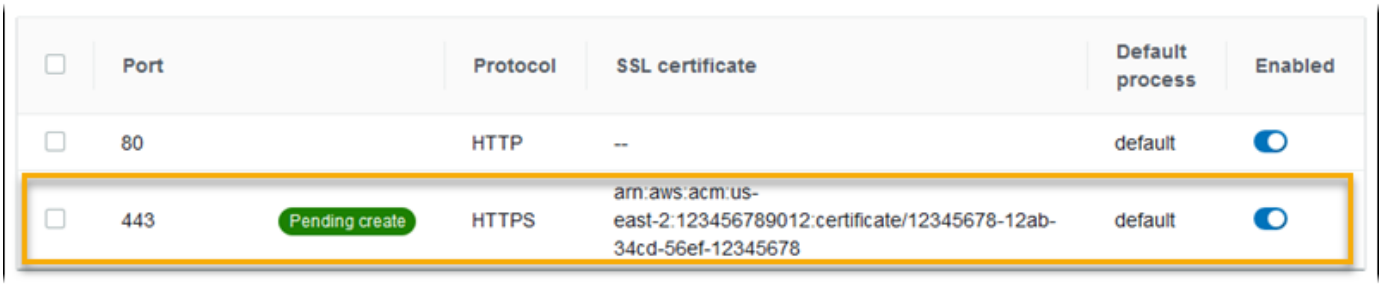
Para configurar o load balancer neste exemplo

1. Adicione um listener seguro. Em Porta, digite **443**. Em Protocolo, selecione **HTTPS**. Em Certificado SSL, escolha o ARN do seu certificado SSL. Por exemplo, **arn:aws:iam::123456789012:server-certificate/abc/certs/build** ou **arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678**.

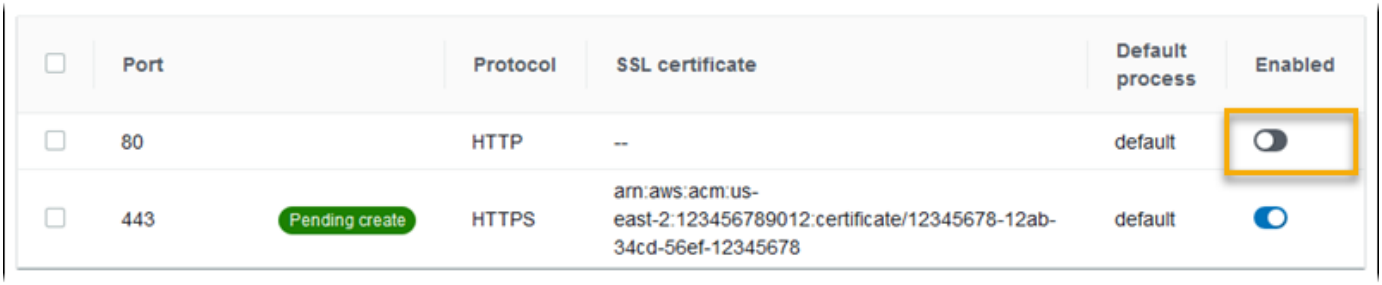
   Em Processo padrão, mantenha **default** selecionado.



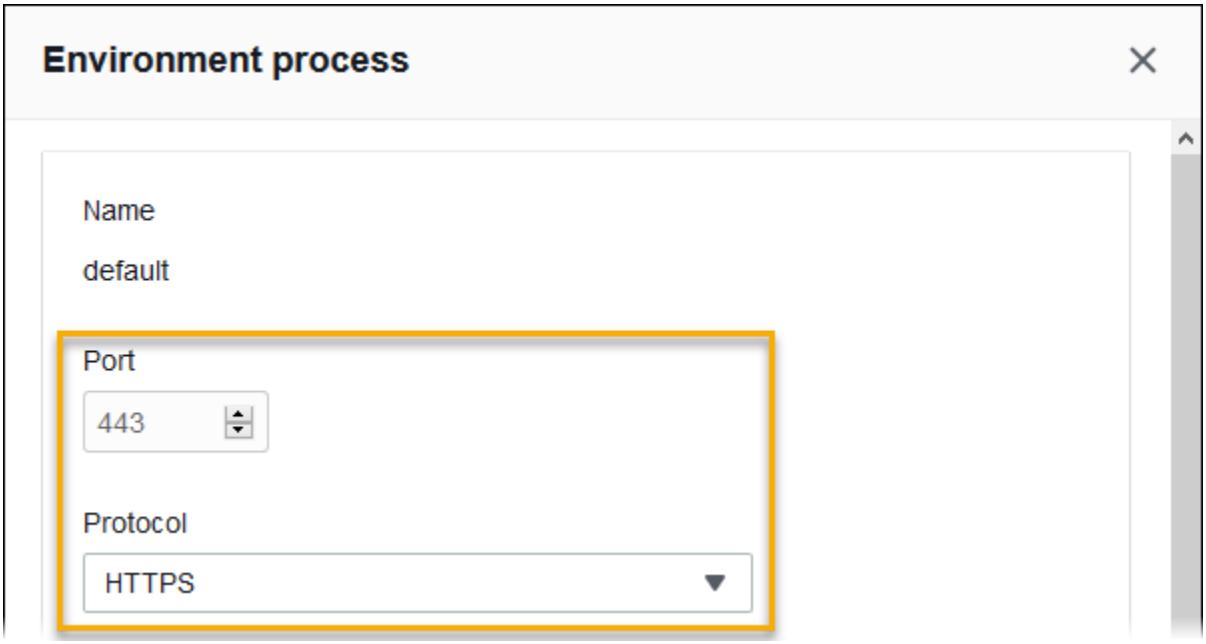
Agora você pode ver seu listener adicional na lista.



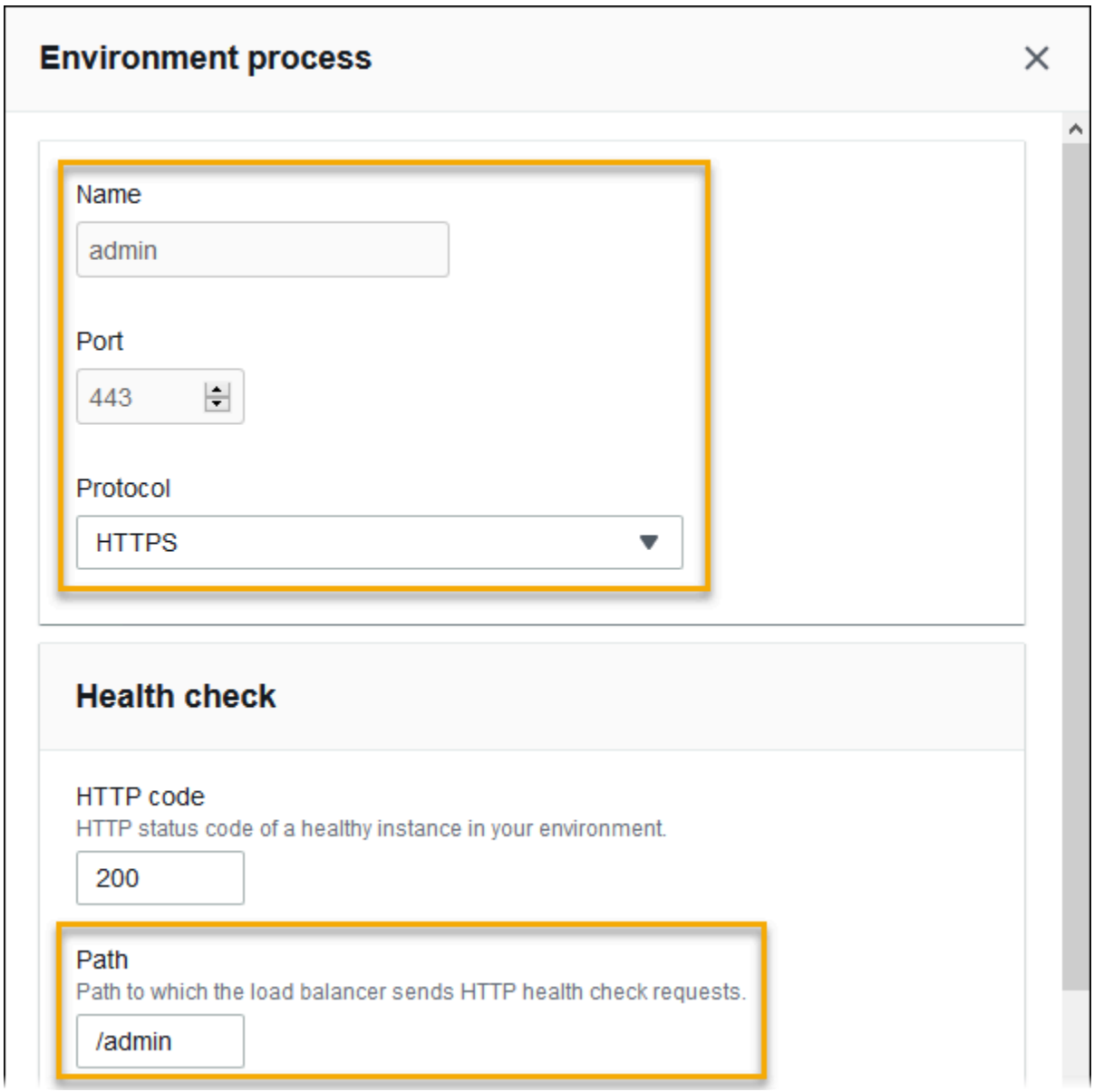
2. Desative o listener HTTP da porta padrão 80. Para o listener padrão, desative a opção Habilitada.



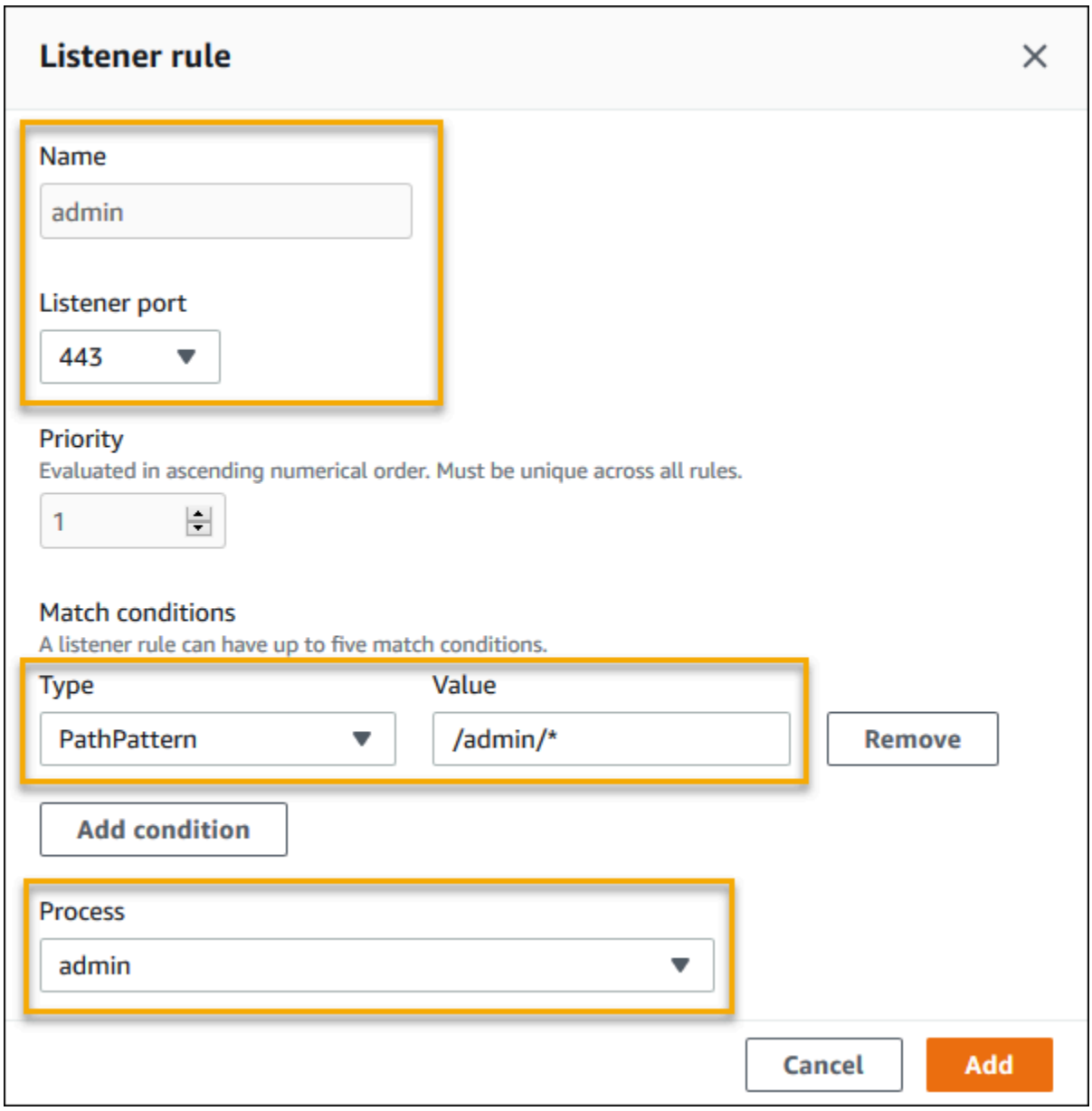
3. Configure o processo padrão como HTTPS. Selecione o processo padrão e, em seguida, para Actions, escolha Edit. Em Porta, digite **`443`**. Em Protocolo, selecione **`HTTPS`**.



4. Adicione um processo administrativo. Para Name (Nome), digite **`admin`**. Em Porta, digite **`443`**. Em Protocolo, selecione **`HTTPS`**. Em Verificação de integridade, para Caminho, digite **`/admin`**.



5. Adicione uma regra para o tráfego de administrador. Para Name (Nome), digite **`admin`**. Para Listener port, digite **`443`**. Para condições de partida, adicione um PathPatterncom o valor**`/admin/*`**. Em Processo, selecione **`admin`**.



Configurar um Application Load Balancer usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o eb create.

```
$ 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):
```

Você também pode especificar um tipo de load balancer com a opção `--elb-type`.

```
$ eb create test-env --elb-type application
```

Namespaces do Application Load Balancer

É possível encontrar configurações relacionadas aos Application Load Balancers nos seguintes namespaces:

- `aws:elasticbeanstalk:environment`: selecione o tipo de balanceador de carga para o ambiente. O valor de um Application Load Balancer é `application`.

  Não é possível definir essa opção nos arquivos de configuração (.Ebextensions).
- `aws:elbv2:loadbalancer`: defina logs de acesso e outras configurações que se aplicam ao Application Load Balancer como um todo.
- `aws:elbv2:listener`: configure listeners no Application Load Balancer. Essas configurações são mapeadas para as configurações em `aws:elb:listener` para Classic Load Balancers.
- `aws:elbv2:listenerrule`: configure as regras que roteiam o tráfego para os diferentes processos, dependendo do caminho da solicitação. As regras são exclusivas dos Application Load Balancers.
- `aws:elasticbeanstalk:environment:process`: configure as verificações de integridade e especifique a porta e o protocolo para os processos que são executados nas instâncias de seu ambiente. A porta e as configurações de protocolo mapeiam a porta e as configurações de protocolo da instância em `aws:elb:listener` para um listener no Classic Load Balancer. As configurações de verificação de integridade são mapeadas para as configurações nos namespaces `aws:elb:healthcheck` e `aws:elasticbeanstalk:application`.

Example .ebextensions/ .config alb-access-logs

O arquivo de configuração a seguir permite fazer upload do log de acesso para um ambiente com um Application Load Balancer.

```
option_settings:
```

```
  aws:elbv2:loadbalancer:
    AccessLogsS3Bucket: amzn-s3-demo-bucket
    AccessLogsS3Enabled: 'true'
    AccessLogsS3Prefix: beanstalk-alb
```

Example .ebextensions/ .config alb-default-process

O arquivo de configuração a seguir modifica as configurações de verificação de integridade e de perdurabilidade no processo padrão.

```
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/ .config alb-secure-listener

O arquivo de configuração a seguir adiciona um listener seguro e um processo de correspondência na 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/ .config alb-admin-rule

O arquivo de configuração a seguir adiciona um listener seguro com uma regra que roteia o tráfego com um caminho de solicitação de `/admin` para um processo chamado `admin` que escuta na porta 4.443.

```
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
```

Configurar um Application Load Balancer compartilhado

Quando você ativa o balanceamento de carga, seu AWS Elastic Beanstalk ambiente é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias em seu ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o Guia do usuário do Elastic Load Balancing. O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um Application Load Balancer compartilhado que você cria e associa ao ambiente. Consulte também the section called “Application Load Balancer”. Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte Balanceador de carga do ambiente do Elastic Beanstalk.

> **Note**
>
> Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo. Também não é possível alternar de um load balancer dedicado para um compartilhado ou vice-versa.

Introdução

Um balanceador de carga compartilhado é um balanceador de carga que você mesmo cria e gerencia usando o serviço Amazon Elastic Compute Cloud (Amazon EC2) e depois usa em vários ambientes do Elastic Beanstalk.

Quando um ambiente é criado com balanceamento de carga e escalabilidade automática e um Application Load Balancer é utilizado, o Elastic Beanstalk cria um balanceador de carga dedicado ao ambiente por padrão. Para saber o que é um Application Load Balancer e como ele funciona em um ambiente do Elastic Beanstalk, consulte a introdução à configuração de um Application Load Balancer para Elastic Beanstalk.

Em algumas situações, você pode economizar o custo de ter vários load balancers dedicados. Isso pode ser útil quando você tem vários ambientes, por exemplo, se a aplicação for um pacote de microsserviços em vez de um serviço monolítico. Nesses casos, é possível optar por usar um load balancer compartilhado.

Para usar um balanceador de carga compartilhado, primeiro crie-o na Amazon EC2 e adicione um ou mais ouvintes. Durante a criação de um ambiente do Elastic Beanstalk, é necessário fornecer o balanceador de carga e escolher uma porta do listener. O Elastic Beanstalk associa o listener ao processo padrão no ambiente. É possível adicionar regras de listener personalizadas para rotear o tráfego de cabeçalhos e caminhos de host específicos para outros processos do ambiente.

O Elastic Beanstalk adiciona uma tag ao balanceador de carga compartilhado. O nome da tag é `elasticbeanstalk:shared-elb-environment-count`, e seu valor é o número de ambientes que compartilham esse load balancer.

Usar um load balancer compartilhado é diferente de usar um dedicado de diversas maneiras.

Referente	Application Load Balancer dedicado	Application Load Balancer compartilhado
Gerenciamento	O Elastic Beanstalk cria e gerencia o balanceador de carga, os listeners, as regras de listener e os processos (grupos de destino). O Elastic Beanstalk também os remove quando o ambiente é encerrado. O Elastic Beanstalk pode definir a captura de logs de acesso do balanceador de carga, caso escolha essa opção.	O balanceador de carga e os listeners são criados e gerenciados fora do Elastic Beanstalk . O Elastic Beanstalk cria e gerencia uma regra padrão e um processo padrão, e regras e processos podem ser adicionados. O Elastic Beanstalk remove as regras e os processos de listener que foram adicionados durante a criação do ambiente.
Regras do listener	O Elastic Beanstalk cria uma regra padrão para cada listener a fim de rotear todo o tráfego para o processo padrão do listener.	O Elastic Beanstalk associa uma regra padrão apenas a um listener da porta 80, se existir. Se você escolher uma porta de listener padrão diferente, precisará associar a regra padrão a ela (o console do Elastic Beanstalk e a CLI do EB fazem isso por você). Para resolver conflitos de condição de regra de listener em ambientes que compartilham o balanceador de carga, o Elastic Balancead or adiciona o CNAME do ambiente à regra de listener como uma condição de cabeçalho de host. O Elastic Beanstalk trata as configurações de prioridade de regra como relativas entre ambientes que compartilham o balanceador de carga e as mapeia para prioridades absolutas durante a criação.
Grupos de segurança	O Elastic Beanstalk cria um grupo de segurança padrão e o anexa ao balanceador de carga.	É possível configurar um ou mais grupos de segurança a serem usados para o load balancer. Caso contrário, o Elastic Beanstalk

Referente	Application Load Balancer dedicado	Application Load Balancer compartilhado
		verifica se um grupo de segurança existente gerenciado pelo Elastic Beanstalk já está conectado ao balanceador de carga. Caso contrário, o Elastic Beanstalk cria um grupo de segurança e o anexa ao balanceador de carga. O Elastic Beanstalk exclui esse grupo de segurança quando o último ambiente que compartilha o balanceador de carga é encerrado.
Atualizações	É possível atualizar o Application Load Balancer após a criação do ambiente. É possível editar listeners, regras de listener e processos. É possível configurar a captura de log de acesso do load balancer.	Não é possível usar o Elastic Beanstalk para configurar a captura de logs de acesso no Application Load Balancer nem atualizar listeners e regras de listener após a criação do ambiente. Só é possível atualizar processos (grupos de destino). Para configurar a captura do registro de acesso e atualizar os ouvintes e as regras do ouvinte, use a Amazon. EC2

Configurar um Application Load Balancer compartilhado usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar um Application Load Balancer compartilhado durante a criação do ambiente. É possível selecionar um dos load balancers compartilháveis da conta para uso no ambiente, selecionar a porta de listener padrão e configurar processos adicionais e regras de listener.

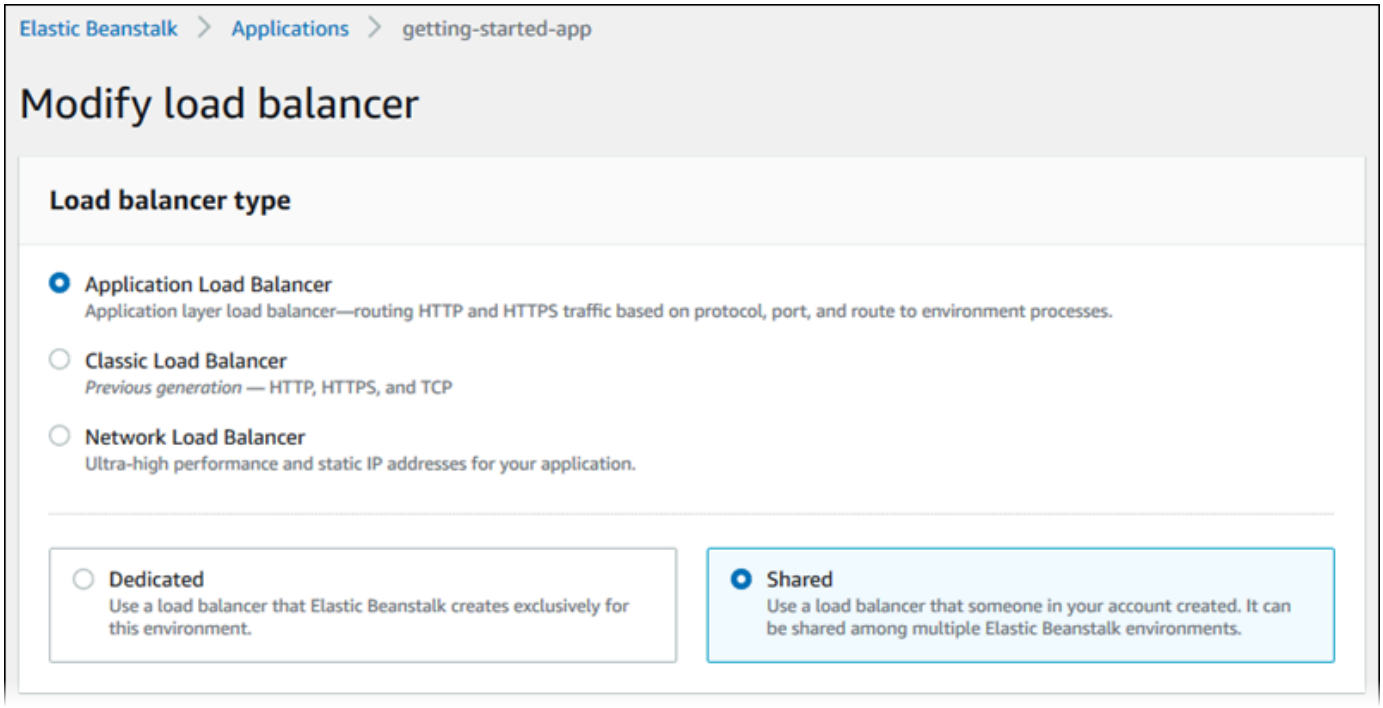
Não é possível editar a configuração do Application Load Balancer compartilhado no console do Application Load Balancer após a criação do ambiente. Para configurar ouvintes, regras de ouvintes, processos (grupos-alvo) e captura de registros de acesso, use a Amazon. EC2

Como configurar um Application Load Balancer no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS

2. No painel de navegação, escolha Ambientes.
3. Escolha Create a new environment (Criar um ambiente) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Escolha a predefinição de configuração High availability (Alta disponibilidade).

   Se preferir, na categoria de configuração Capacity (Capacidade), configure o tipo de ambiente Load balanced (Carga equilibrada). Para obter detalhes, consulte Capacidade.
6. Na categoria de configuração Load balancer, escolha Edit (Editar).
7. Selecione a opção Application Load Balancer se ainda não estiver selecionada e, depois, selecione a opção Shared (Compartilhado).



8. Faça todas as alterações de configuração do Application Load Balancer compartilhado necessárias ao seu ambiente.
9. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
10. Selecione Criar ambiente.

Configurações do Application Load Balancer compartilhado

- Application Load Balancer compartilhado

- Processos
- Regras

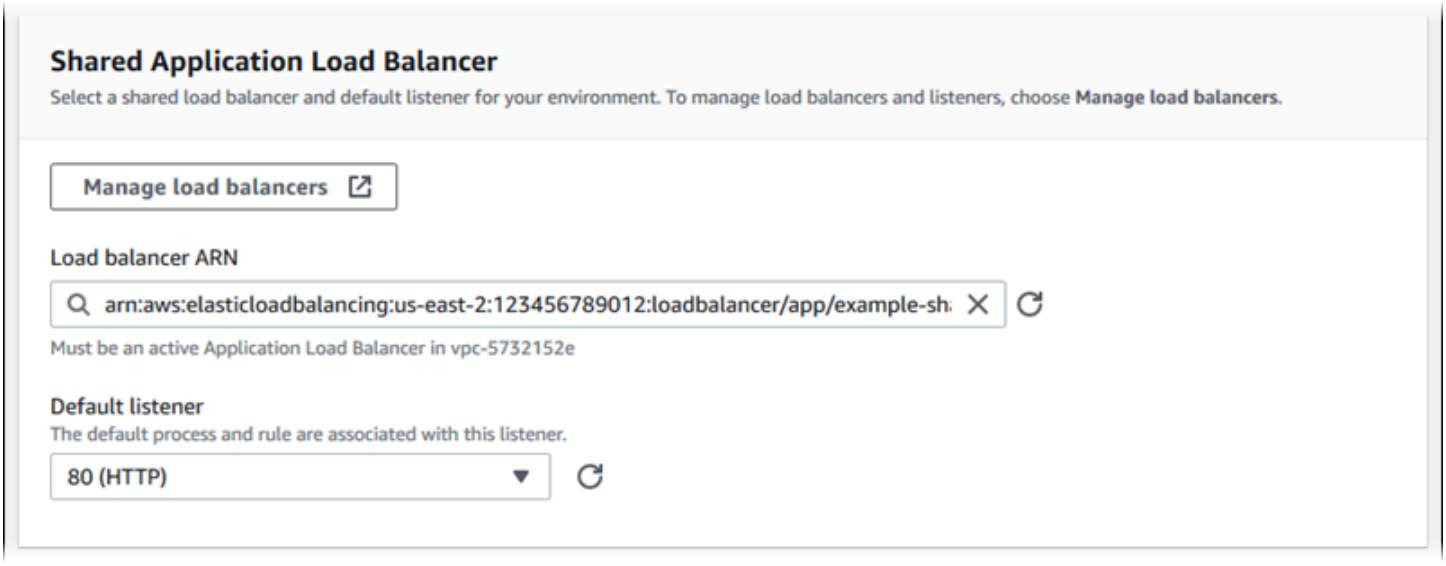
Application Load Balancer compartilhado

Use esta seção para escolher um Application Load Balancer compartilhado para seu ambiente e configurar o roteamento de tráfego padrão.

Antes de configurar um Application Load Balancer compartilhado aqui, use EC2 a Amazon para definir pelo menos um Application Load Balancer para compartilhamento, com pelo menos um ouvinte, em sua conta. Se você ainda não fez isso, poderá escolher Manage load balancers (Gerenciar balanceadores de carga). O Elastic Beanstalk abre EC2 o console da Amazon em uma nova guia do navegador.

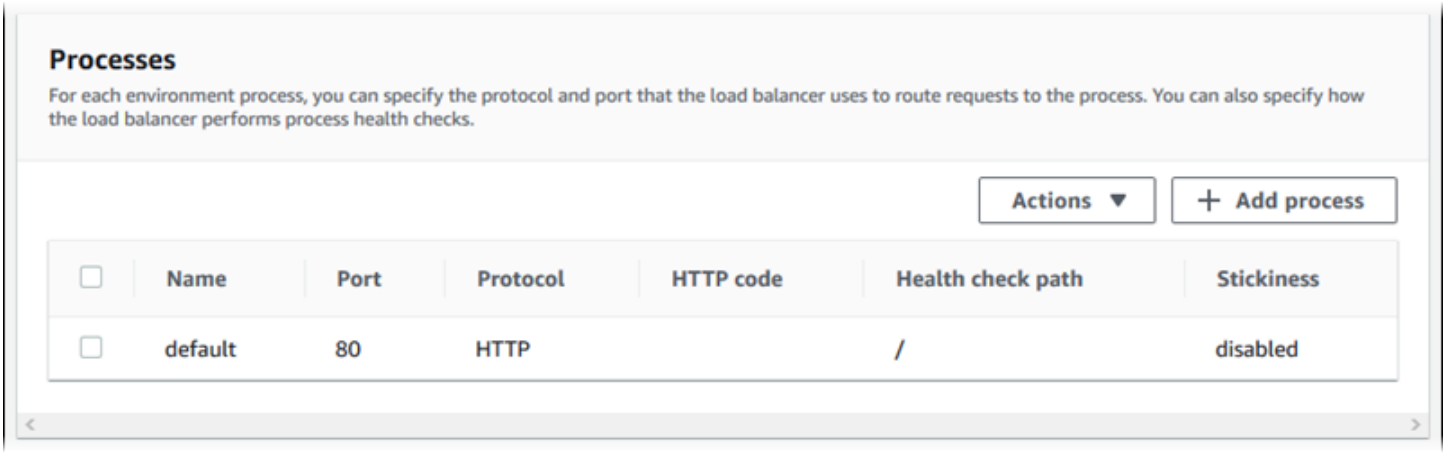
Quando terminar de configurar balanceadores de carga compartilhados fora do Elastic Beanstalk, defina as configurações a seguir nesta seção do console:

- Load balancer ARN (ARN do balanceador de carga): o balanceador de carga compartilhado a ser usado nesse ambiente. Selecione o load balancer na lista ou insira o nome de recurso da Amazon (ARN) de um load balancer.
- Default listener port (Porta do listener padrão): uma porta do listener que o balanceador de carga compartilhado escuta. Faça sua escolha em uma lista de portas de listener existentes. O tráfego desse listener com o CNAME do ambiente no cabeçalho do host é roteado para um processo padrão nesse ambiente.



Processos

Use essa lista para especificar processos para o load balancer compartilhado. Um processo é um destino para os listeners direcionarem o tráfego. Inicialmente, a lista mostra o processo padrão, que recebe tráfego do listener padrão.



Como configurar um processo existente

1. Marque a caixa de seleção ao lado da entrada da tabela e, em seguida, escolha Actions, Edit.
2. Use a caixa de diálogo Environment process (Processo do ambiente) para editar as configurações e selecione Save (Salvar).

Como adicionar um processo

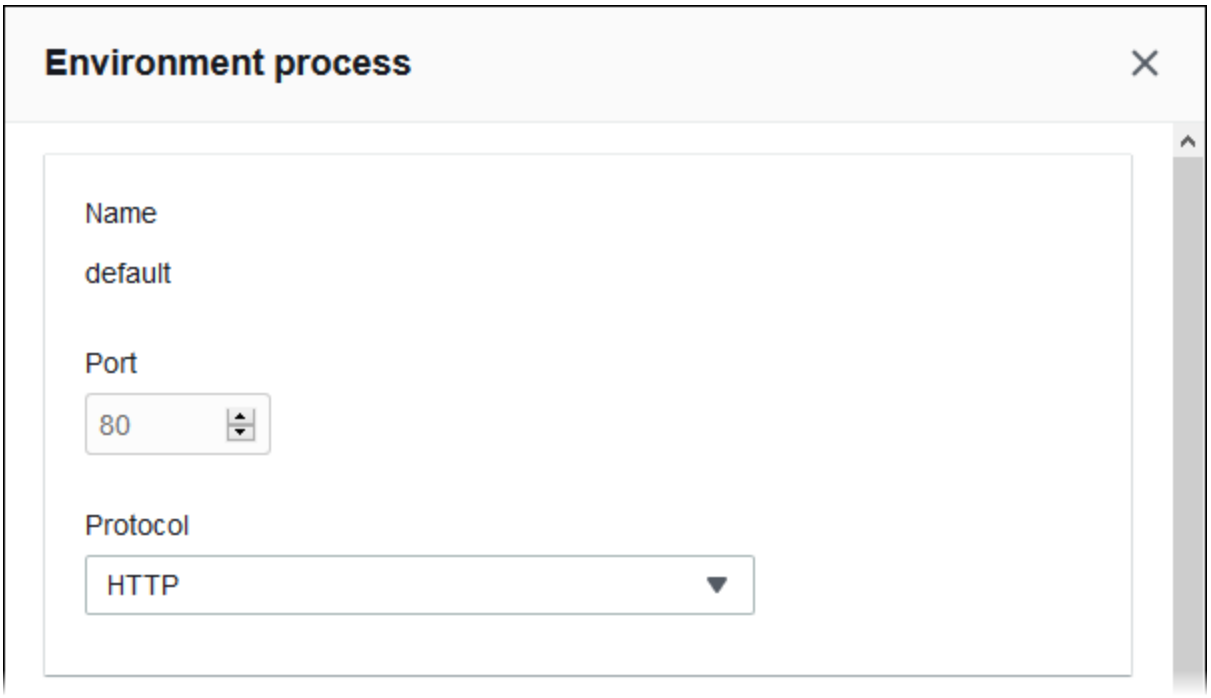
1. Selecione Add process (Adicionar processo).
2. Na caixa de diálogo Environment Process (Processo do ambiente), defina as configurações desejadas e selecione Add (Adicionar).

Configurações da caixa de diálogo Environment process (Processo do ambiente) do Application Load Balancer

- Definição
- Verificação de integridade
- Sessões

Definição

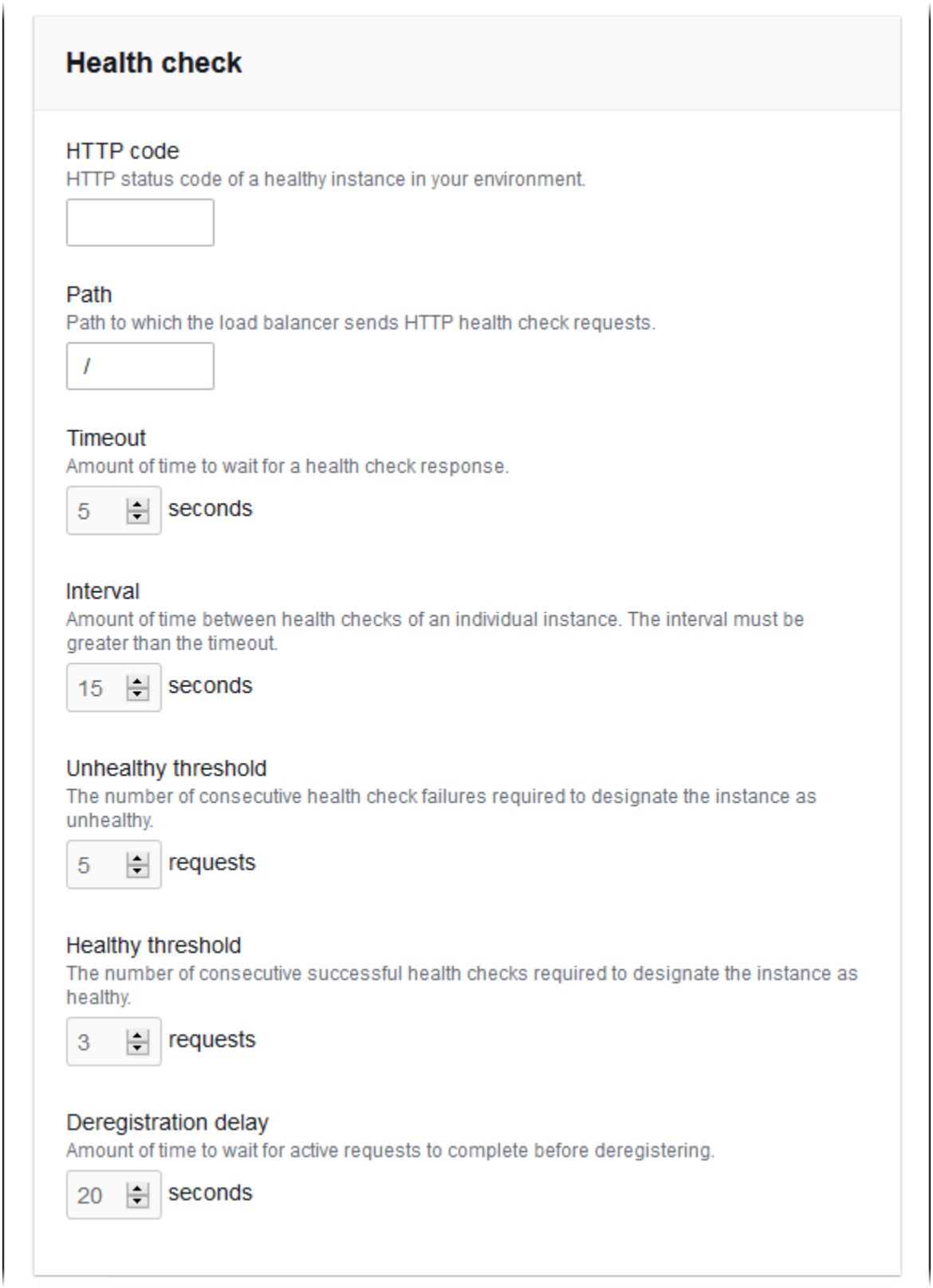
Use estas configurações para definir o processo Name, Port e Protocol em que as solicitações são observadas.



Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do processo:

- HTTP code (Código de HTTP): o código de status HTTP que designa um processo íntegro.
- Path (Caminho): o caminho da solicitação de verificação de integridade para o processo.
- Timeout (Tempo limite): a quantidade de tempo, em segundos, para aguardar uma resposta de verificação de integridade.
- Interval (Intervalo): a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual. O intervalo deve ser maior que o tempo limite.
- Unhealthy threshold (Limite não íntegro), Healthy threshold (Limite íntegro): o número de verificações de integridade que devem falhar ou passar, respectivamente, antes que o Elastic Load Balancing altere o estado de integridade de uma instância.
- Deregistration delay (Atraso do cancelamento de registro): a quantidade de tempo, em segundos, para aguardar que as solicitações ativas sejam concluídas antes de cancelar o registro de uma instância.

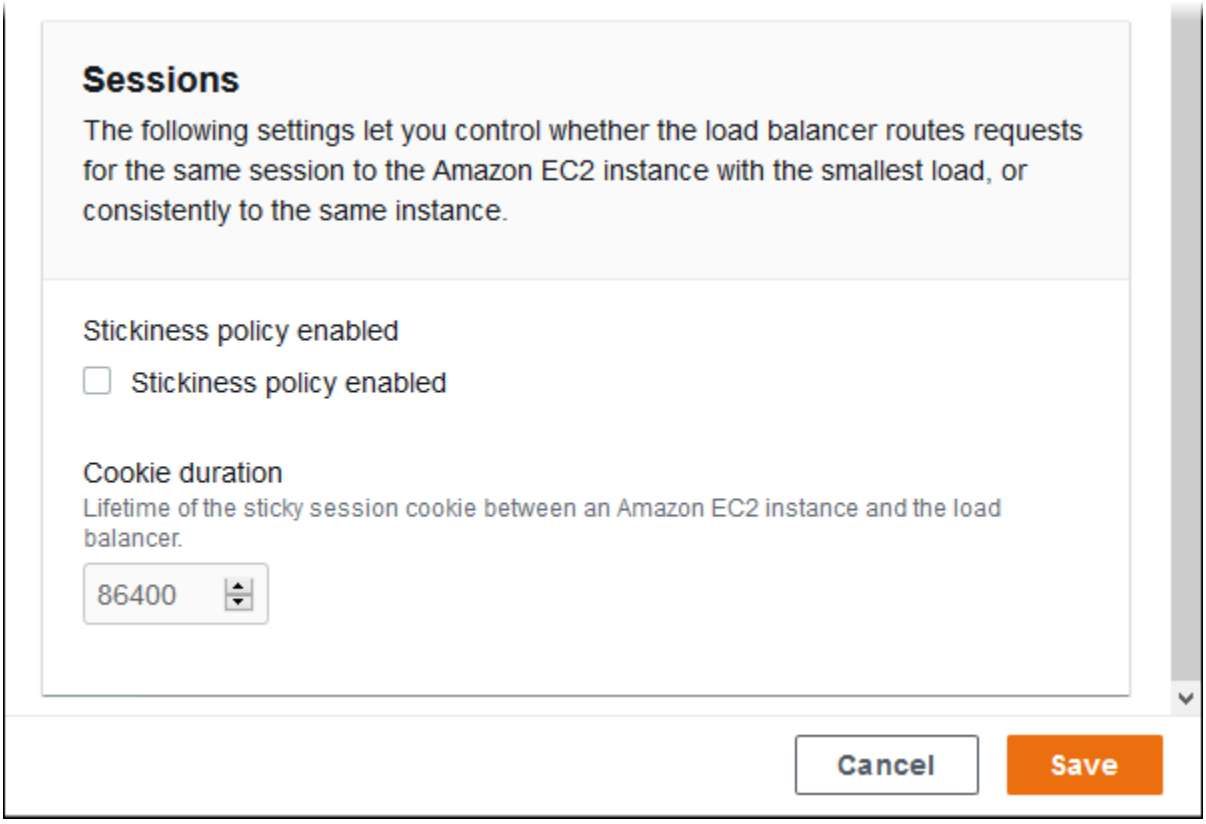

Health check
HTTP code
HTTP status code of a healthy instance in your environment.
Path
Path to which the load balancer sends HTTP health check requests.
/
Timeout
Amount of time to wait for a health check response.
5 seconds
Interval
Amount of time between health checks of an individual instance. The 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**
>
> A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falham em uma verificação de saúde do Elastic Load Balancing não são substituídas automaticamente pelo Amazon Auto EC2 Scaling, a menos que você configure manualmente o Amazon Auto EC2 Scaling para fazer isso. Para mais detalhes, consulte Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk.

Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte Relatórios de integridade básica.

Sessões

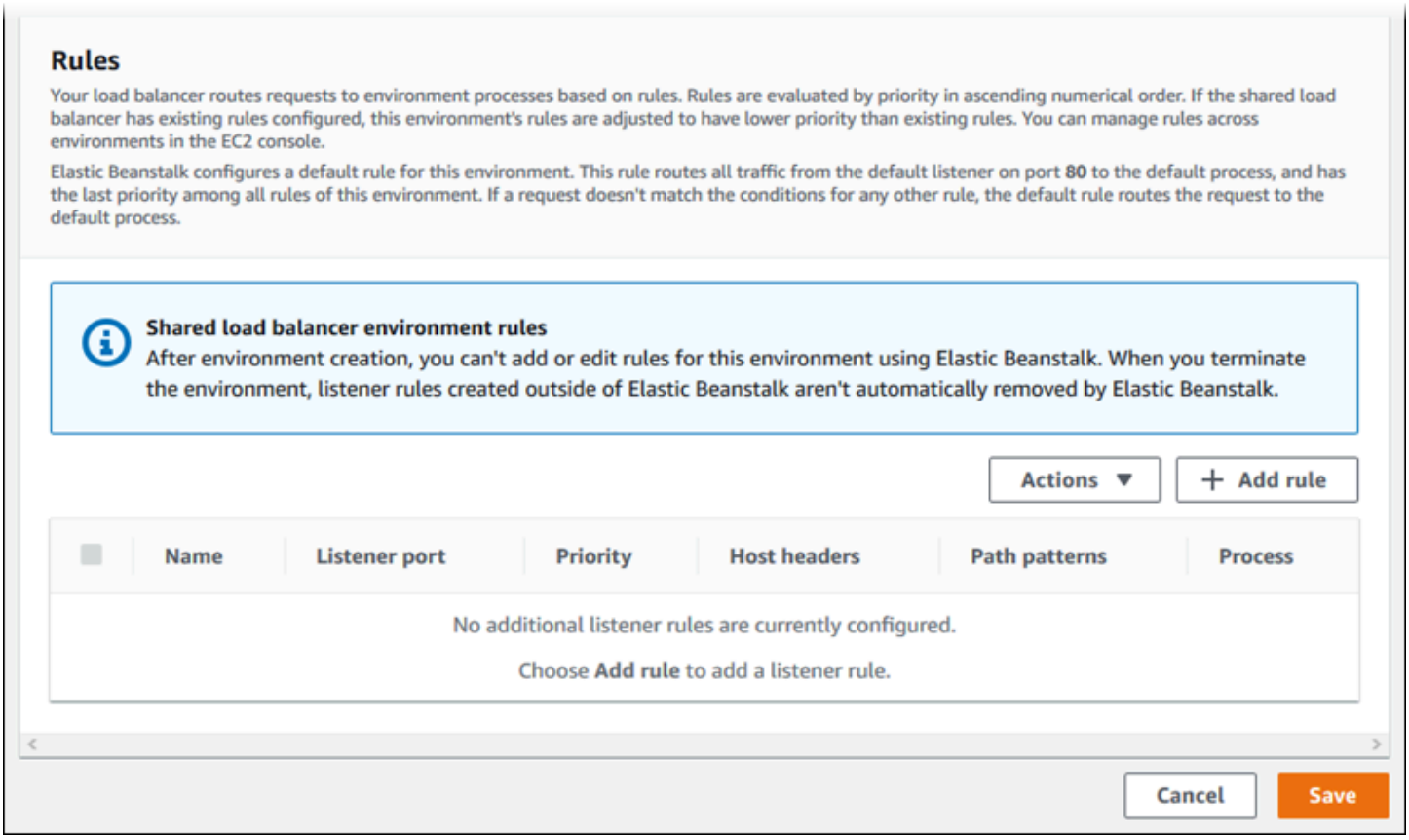
Selecione ou desmarque a caixa Política de perdurabilidade habilitada para habilitar ou desabilitar as sticky sessions. Use Duração do cookie para configurar a duração de uma sticky session até **`604800`** segundos.



Regras

Use essa lista para especificar regras de listener para o load balancer compartilhado. Uma regra mapeia solicitações que o listener recebe em um padrão de caminho específico para um processo de destino. Cada listener pode ter várias regras, roteando solicitações em caminhos diferentes para processos diferentes em instâncias dos diferentes ambientes que compartilham o listener.

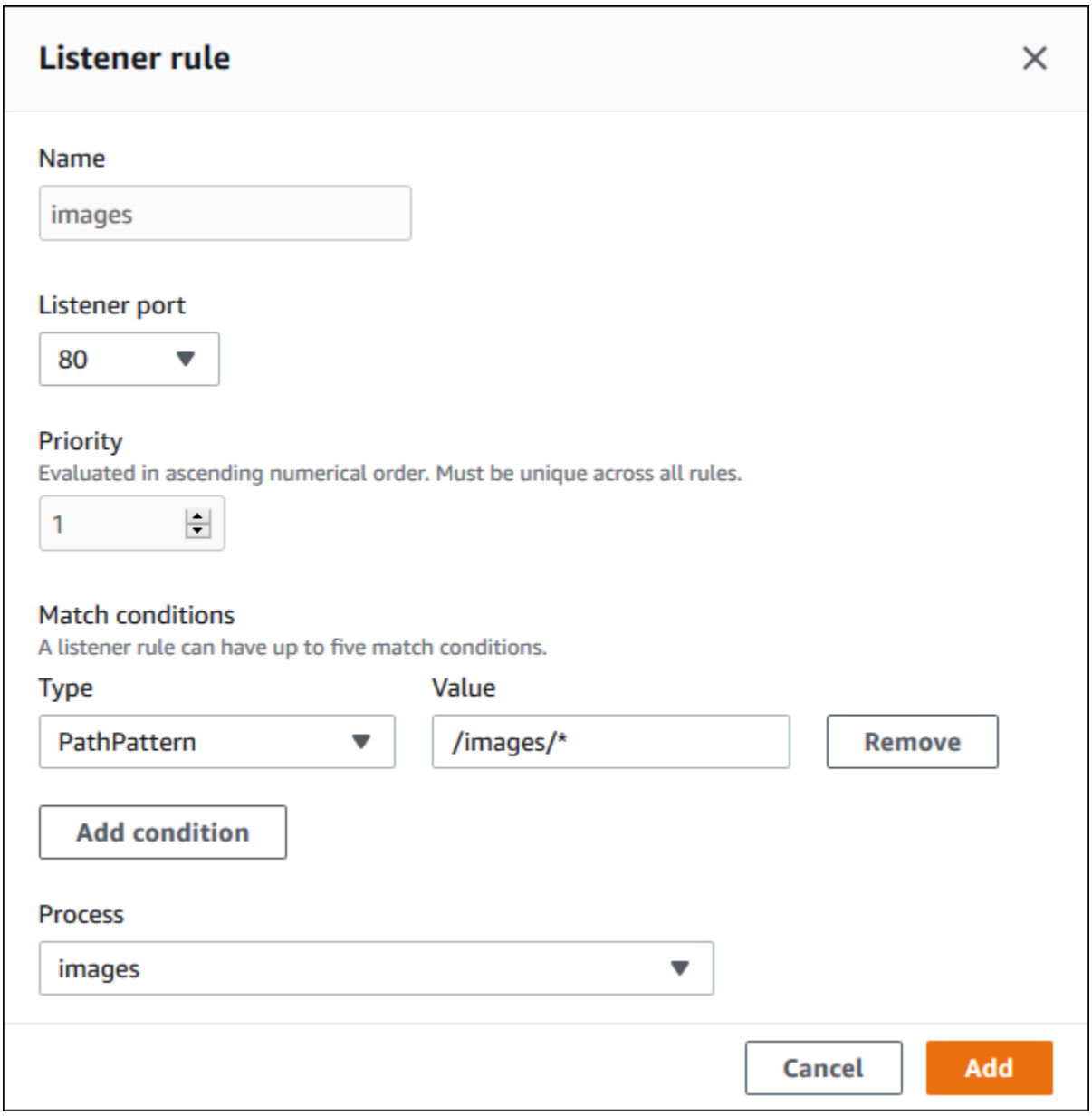
As regras têm prioridades numéricas que determinam a precedência em que são aplicadas às solicitações recebidas. O Elastic Beanstalk adiciona uma regra padrão que roteia todo o tráfego do listener padrão para o processo padrão do seu novo ambiente. A precedência da regra padrão é a mais baixa. Ela é aplicada se nenhuma outra regra para o mesmo listener corresponder à solicitação recebida. Inicialmente, se você não tiver adicionado regras personalizadas, a lista estará vazia. A regra padrão não é exibida.



Você pode editar as configurações de uma regra existente ou adicionar uma nova regra. Para começar a editar uma regra na lista ou adicionar uma regra a ela, use as mesmas etapas listadas para a lista de processos. A caixa de diálogo Listener rule é exibida com as seguintes configurações:

- Name (Nome): o nome da regra.
- Listener port (Porta do listener): porta do listener ao qual a regra se aplica.

- Priority (Prioridade): prioridade da regra. Um número de prioridade mais baixa tem maior precedência. As prioridades das regras de um listener devem ser únicas. O Elastic Beanstalk trata as prioridades de regras como relativas em ambientes de compartilhamento e as mapeia para prioridades absolutas durante a criação.
- Match conditions (Condições de correspondência): uma lista de condições do URL de solicitação às quais a regra se aplica. Há dois tipos de condições: HostHeader(a parte do domínio do URL) e PathPattern(a parte do caminho do URL). Uma condição é reservada para o subdomínio do ambiente e é possível adicionar até quatro condições. Cada valor de condição tem até 128 caracteres de comprimento e pode incluir caracteres curinga.
- Process (Processo): o processo para o qual o balanceador de carga direciona solicitações que correspondem à regra.



Exemplo: use um Application Load Balancer compartilhado para um aplicativo seguro micro-service-based

Neste exemplo, a aplicação consiste em vários microsserviços, cada um implementado como um ambiente do Elastic Beanstalk. Além disso, você precisa de criptografia de end-to-end tráfego. Vamos demonstrar um dos ambientes de microsserviço, que tem um processo principal para solicitações de usuários e um processo separado para lidar com solicitações administrativas.

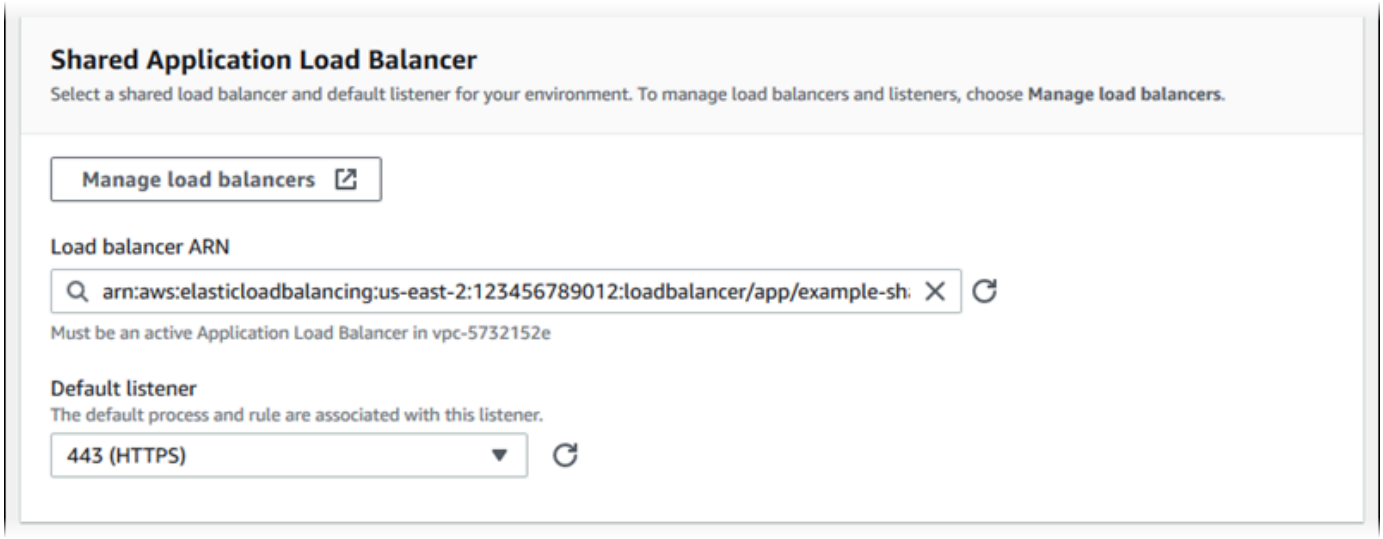
Para atender a esses requisitos, use EC2 a Amazon para criar um Application Load Balancer que você compartilhará entre seus microsserviços. Adicione um listener seguro na porta 443 e no protocolo HTTPS. Depois, adicione vários certificados SSL ao listener: um por domínio de

microsserviço. Para obter detalhes sobre como criar o Application Load Balancer e o listener seguro, consulte Criar um Application Load Balancer e Criar um listener HTTPS para o Application Load Balancer no Guia do usuário para Application Load Balancers.

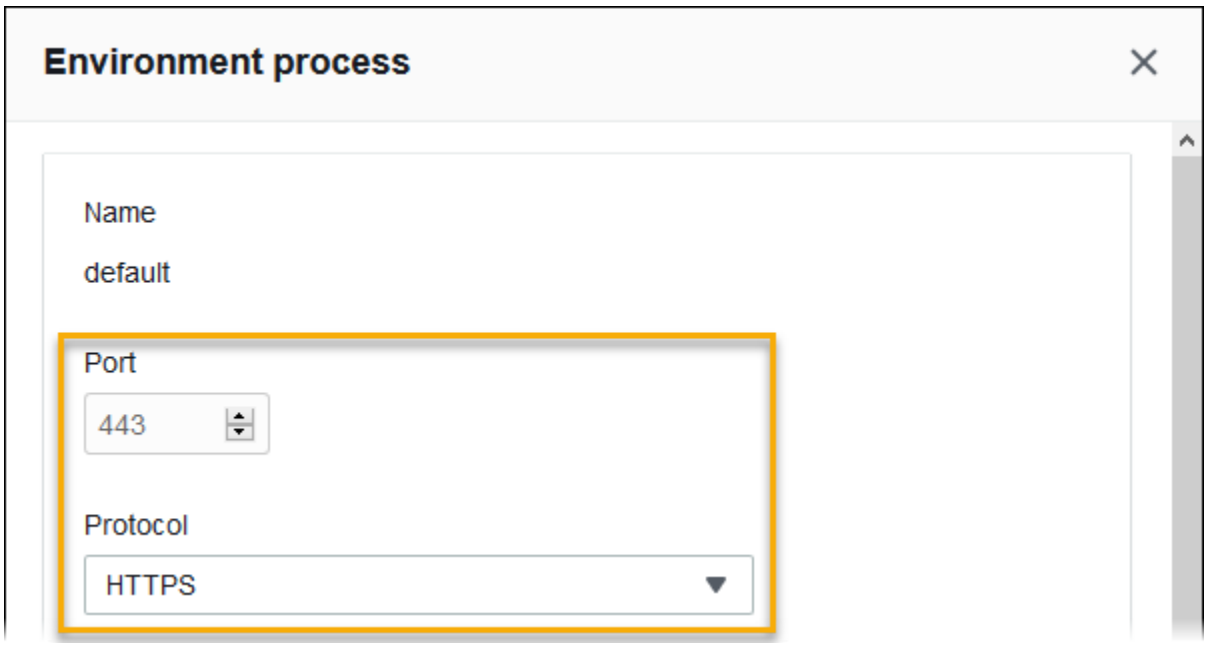
No Elastic Beanstalk, configure cada ambiente de microsserviço para usar o Application Load Balancer compartilhado e defina a porta do listener padrão como 443. No caso do ambiente específico que estamos demonstrando aqui, indique que o processo padrão escuta a porta 443 em HTTPS e adicione um processo e uma regra de listener ao tráfego administrativo em um caminho diferente.

Como configurar o load balancer compartilhado neste exemplo

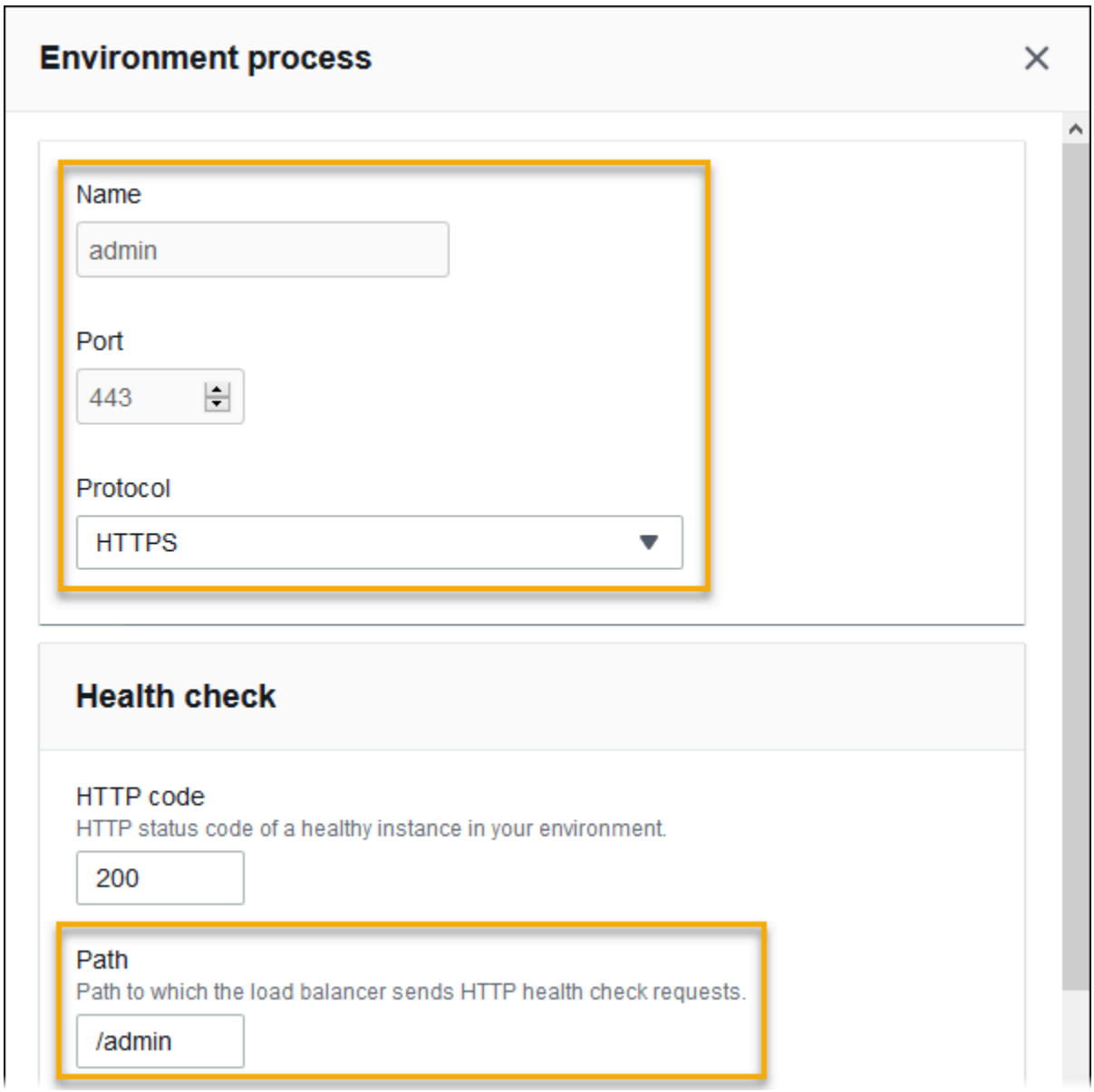
1. Na seção Shared Application Load Balancer (Application Load Balancer compartilhado), selecione o balanceador de carga e, depois, para Default listener port (Porta do listener padrão), selecione **443**. A porta do listener já deverá estar selecionada se for o único listener que o load balancer possui.



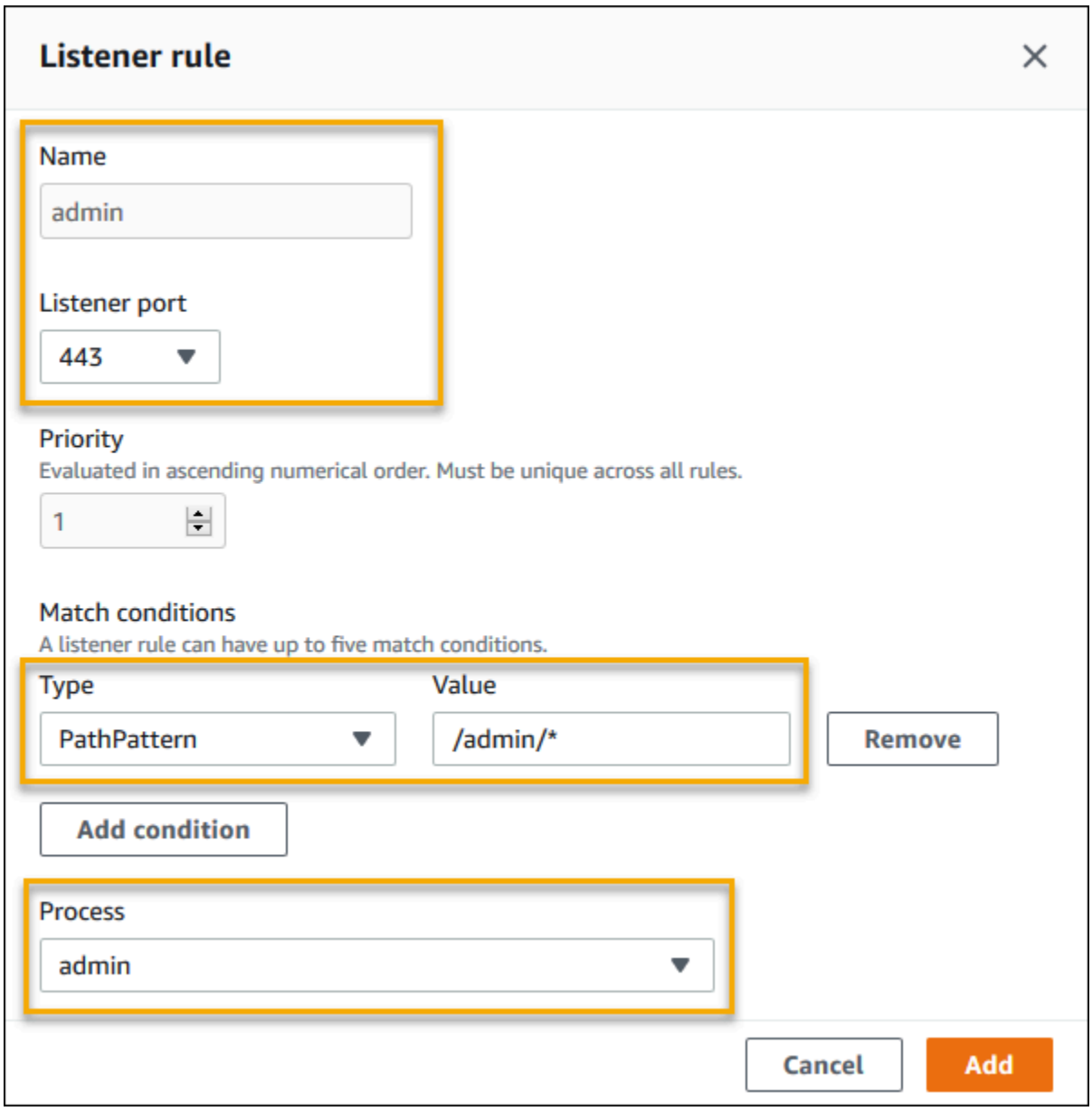
2. Configure o processo padrão como HTTPS. Selecione o processo padrão e, em seguida, para Actions, escolha Edit. Em Porta, insira **443**. Em Protocolo, selecione **HTTPS**.



3. Adicione um processo administrativo. Em Nome, digite **`admin`**. Em Porta, insira **`443`**. Em Protocolo, selecione **`HTTPS`**. Em Health check (Verificação de integridade), para Path (Caminho), insira **`/admin`**.



4. Adicione uma regra para o tráfego de administrador. Em Nome, digite **`admin`**. Em Listener port (Porta do listener), insira **`443`**. Para condições de partida, adicione um PathPatterncom o valor**`/admin/*`**. Em Processo, selecione **`admin`**.



Configurar um Application Load Balancer compartilhado usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o eb create. Se você escolher `application` (o padrão) e sua conta tiver pelo menos um Application Load Balancer compartilhável, a CLI do EB também perguntará se você deseja usar um Application Load Balancer compartilhado. Se responder **y**, você também será solicitado a selecionar o load balancer e a porta padrão.

```
$ 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
```

Também é possível especificar um load balancer compartilhado usando opções de comando.

```
$ eb create test-env --elb-type application --shared-lb MySharedALB2 --shared-lb-
port 443
```

Namespaces do Application Load Balancer compartilhado

Você pode encontrar configurações relacionadas aos Application Load Balancers compartilhados nos seguintes namespaces:

- `aws:elasticbeanstalk:environment`: escolha o tipo de balanceador de carga para o ambiente e informe o Elastic Beanstalk que você usará um balanceador de carga compartilhado.

  Não é possível definir essas duas opções em arquivos de configuração (.Ebextensions).
- `aws:elbv2:loadbalancer`: configure o ARN e os grupos de segurança do Application Load Balancer compartilhado.

- `aws:elbv2:listener`: associe os listeners do Application Load Balancer compartilhado aos processos de ambiente listando regras do listener.
- `aws:elbv2:listenerrule`: configure as regras do listener que roteiam o tráfego para os diferentes processos, dependendo do caminho da solicitação. As regras são exclusivas dos Application Load Balancers, sejam eles dedicados ou compartilhados.
- `aws:elasticbeanstalk:environment:process`: configure as verificações de integridade e especifique a porta e o protocolo para os processos que são executados nas instâncias de seu ambiente.

Example .ebextensions/ .config application-load-balancer-shared

Para começar a usar um Application Load Balancer compartilhado, use o console do Elastic Beanstalk, a CLI do EB ou a API para definir o tipo de balanceador de carga como `application` e optar por usar um balanceador de carga compartilhado. Use um arquivo de configuração para configurar o load balancer compartilhado.

```
option_settings:
  aws:elbv2:loadbalancer:
    SharedLoadBalancer: arn:aws:elasticloadbalancing:us-
east-2:123456789012:loadbalancer/app/MySharedALB2/e574ea4c37ad2ec8
```

> **Note**
>
> Só é possível configurar essa opção durante a criação do ambiente.

Example .ebextensions/ .config alb-shared-secure-listener

O arquivo de configuração a seguir seleciona um listener seguro padrão na porta 443 para o load balancer compartilhado e define o processo padrão para escutar a 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/ .config alb-shared-admin-rule

O arquivo de configuração a seguir se baseia no exemplo anterior e adiciona uma regra que roteia o tráfego com um caminho de solicitação de `/admin` para um processo chamado `admin` que escuta na 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
```

Configurar um Network Load Balancer

Quando você ativa o balanceamento de carga, seu AWS Elastic Beanstalk ambiente é equipado com um balanceador de carga do Elastic Load Balancing para distribuir o tráfego entre as instâncias em seu ambiente. O Elastic Load Balancing é compatível com vários tipos de balanceador de carga. Para saber mais sobre eles, consulte o Guia do usuário do Elastic Load Balancing. O Elastic Beanstalk pode criar um balanceador de carga para você ou permitir a especificação de um balanceador de carga compartilhado que você criou.

Este tópico descreve a configuração de um Network Load Balancer que o Elastic Beanstalk cria e dedica ao seu ambiente. Para obter informações sobre como configurar todos os tipos de balanceador de carga compatíveis com o Elastic Beanstalk, consulte Balanceador de carga do ambiente do Elastic Beanstalk.

> **Note**
>
> Você só pode escolher o tipo de load balancer que seu ambiente usará durante a criação do ambiente. Você pode alterar as configurações para gerenciar o comportamento do load balancer do ambiente em execução, mas não pode alterar seu tipo.

Introdução

Com o Network Load Balancer, o listener padrão aceita solicitações de TCP na porta 80 e as distribui para as instâncias no seu ambiente. Você pode configurar o comportamento de verificação de integridade, configurar uma porta do listener ou adicionar um listener a outra porta.

> **Note**
>
> Ao contrário de um Classic Load Balancer ou um Application Load Balancer, um Network Load Balancer não pode ter listeners HTTP ou HTTPS da camada de aplicação (camada 7). Ele só oferece suporte aos listeners de TCP da camada de transporte (camada 4). O tráfego HTTP e HTTPS podem ser roteados para o ambiente por TCP. Para estabelecer conexões HTTPS seguras entre os clientes da Web e seu ambiente, instale um certificado autoassinado nas instâncias do ambiente e configure as instâncias para escutar na porta apropriada (normalmente 443) e encerrar as conexões HTTPS. A configuração varia de acordo com a plataforma. Para obter instruções, consulte Configuração da terminação de HTTPS na instância. Depois, configure o Network Load Balancer para adicionar um listener que é mapeado para um processo que escuta na porta apropriada.

Um Network Load Balancer é compatível com verificações de integridade ativas. Essas verificações são baseadas em mensagens para o caminho da raiz (`/`). Além disso, um Network Load Balancer é compatível com verificações de integridade passivas. Ele detecta automaticamente as instâncias de back-end defeituosas e roteia o tráfego somente para instâncias íntegras.

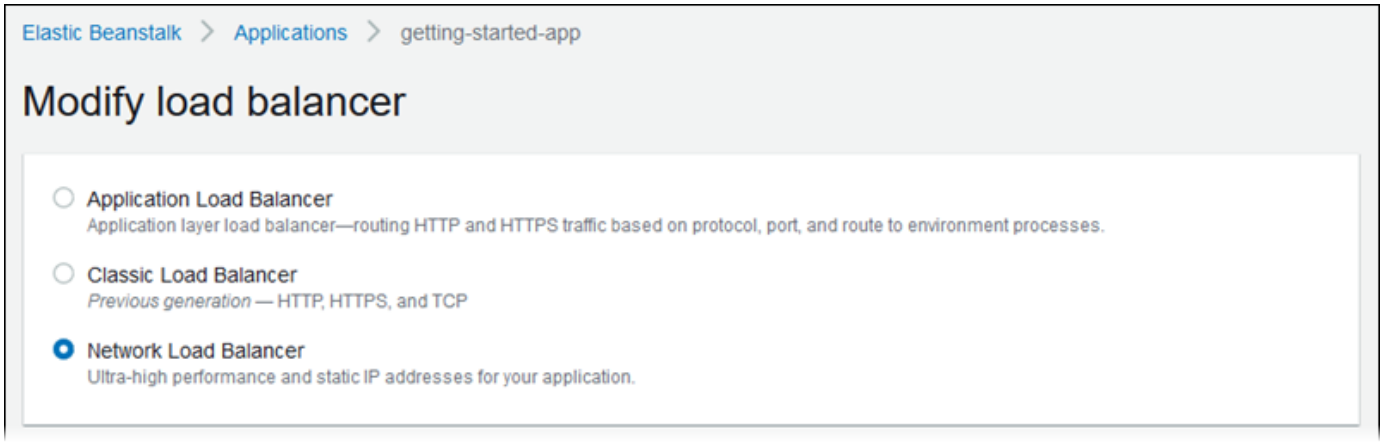
Configurar um Network Load Balancer usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar os listeners e os processos de um Network Load Balancer durante a criação do ambiente ou mais tarde quando seu ambiente estiver em execução.

Como configurar um Network Load Balancer no console do Elastic Beanstalk durante a criação do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Ambientes.
3. Escolha Create a new environment (Criar um ambiente) para começar a criar seu ambiente.
4. Na página principal do assistente, antes de escolher Create environment, escolha Configure more options.
5. Escolha a predefinição de configuração High availability (Alta disponibilidade).

   Se preferir, na categoria de configuração Capacity (Capacidade), configure o tipo de ambiente Load balanced (Carga equilibrada). Para obter detalhes, consulte Capacidade.
6. Na categoria de configuração Load balancer, escolha Edit (Editar).
7. Selecione a opção Network Load Balancer se ainda não estiver selecionada.



8. Faça todas as alterações de configuração do Network Load Balancer necessárias ao seu ambiente.
9. Escolha Save e, em seguida, faça as outras alterações de configuração de acordo com seu ambiente.
10. Selecione Criar ambiente.

Como configurar o Network Load Balancer de um ambiente em execução no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

> **Note**
>
> Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer. Para aprender como configurar um, consulte Alterar tipo de ambiente.

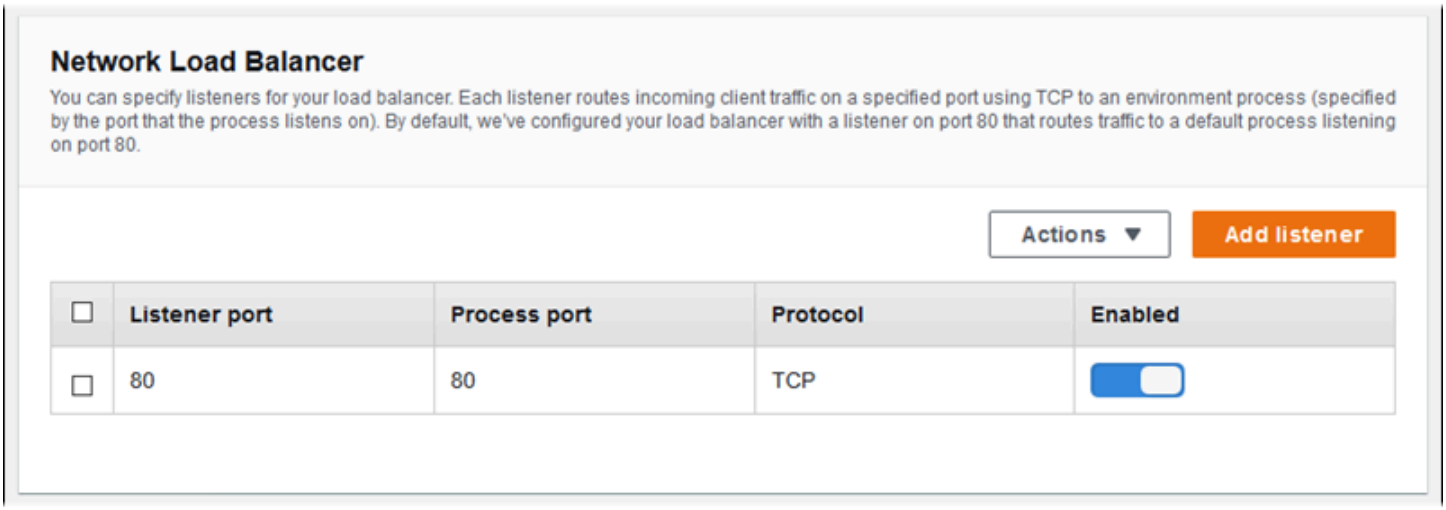
5. Faça as alterações de configuração do Network Load Balancer necessárias ao seu ambiente.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurações do Network Load Balancer

- Listeners
- Processos

Listeners

Use esta lista para especificar listeners para seu load balancer. Cada listener direciona o tráfego de entrada do cliente em uma porta especificada para um processo em suas instâncias. Inicialmente, a lista mostra o listener padrão, que encaminha o tráfego de entrada na porta 80 para um processo chamado default, que observa a porta 80.



Para configurar um listener existente

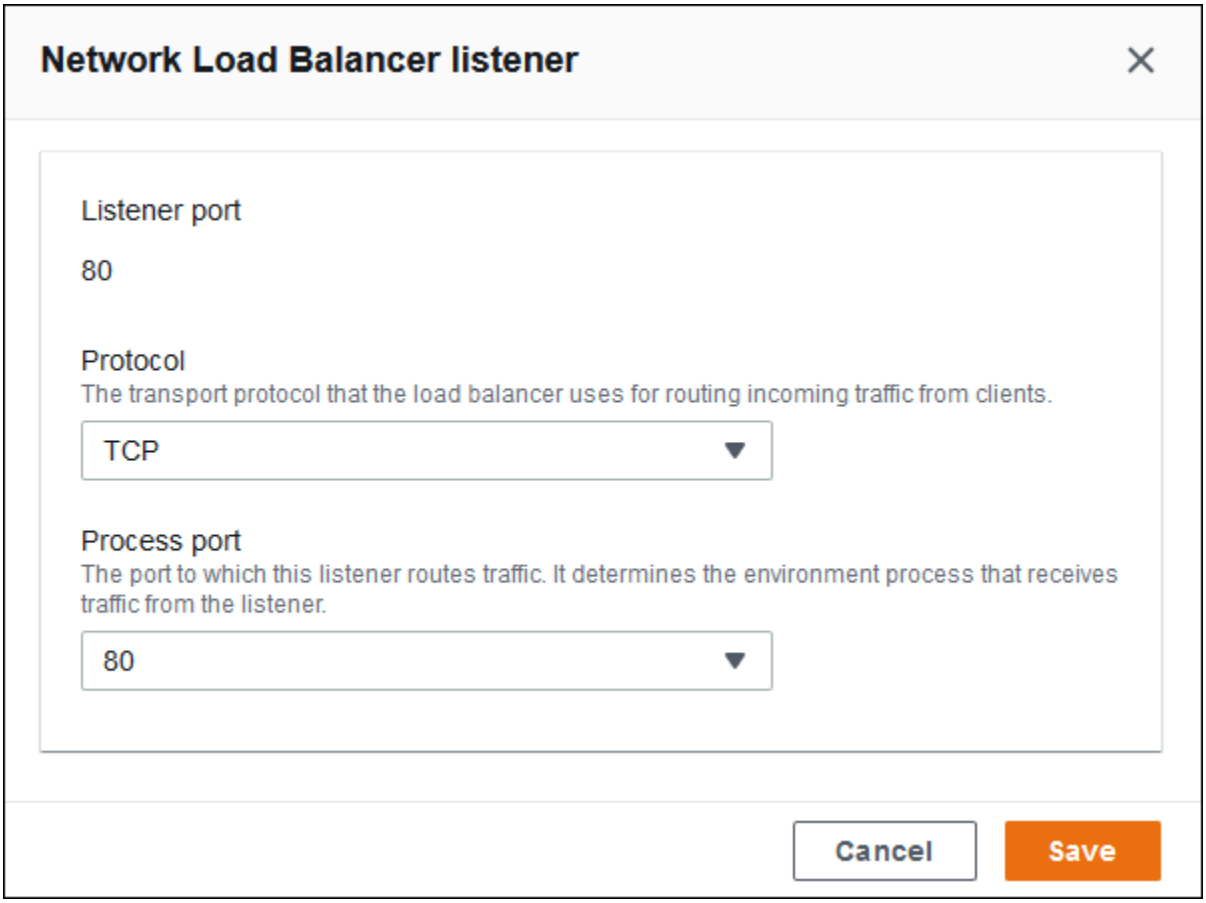
1. Marque a caixa de seleção ao lado da entrada da tabela e, em seguida, escolha Actions, Edit.

2. Use a caixa de diálogo Network Load Balancer listener para editar as configurações e, em seguida, escolha Save.

Para adicionar um listener

1. Escolha Add listener.
2. Na caixa de diálogo Network Load Balancer listener, defina as configurações obrigatórias e, em seguida, escolha Add.

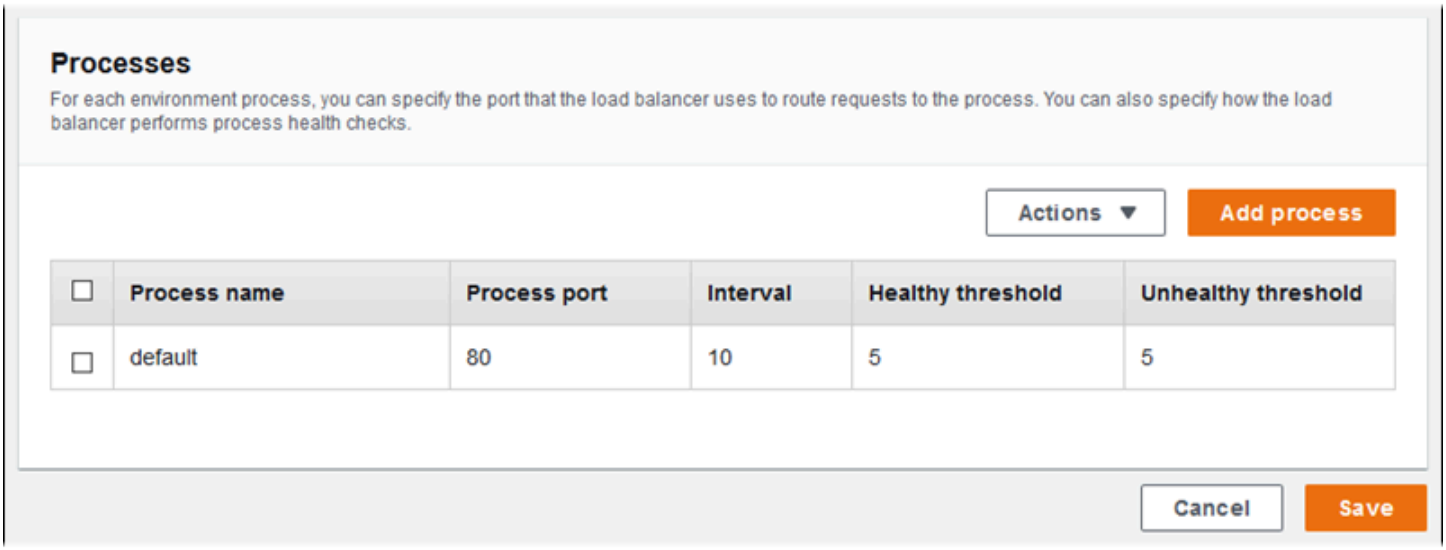
Use a caixa de diálogo Network Load Balancer listener para configurar a porta na qual o ouvinte atende ao tráfego e para escolher o processo para o qual você deseja rotear o tráfego (especificado pela porta na qual o processo escuta).



Processos

Use esta lista para especificar processos para seu load balancer. Um processo é um destino para os listeners direcionarem o tráfego. Cada listener direciona o tráfego de entrada do cliente em uma

porta especificada para um processo em suas instâncias. Inicialmente, a lista mostra o processo padrão, que observa o tráfego de entrada na porta 80.



Você pode editar as configurações de um processo existente ou adicionar um novo processo. Para começar a editar um processo na lista ou adicionar um processo a ele, use as mesmas etapas listadas para lista de listener. A caixa de diálogo Environment process é exibida.

Configurações da caixa de diálogo Environment process (Processo do ambiente) do Network Load Balancer

- Definição
- Verificação de integridade

Definição

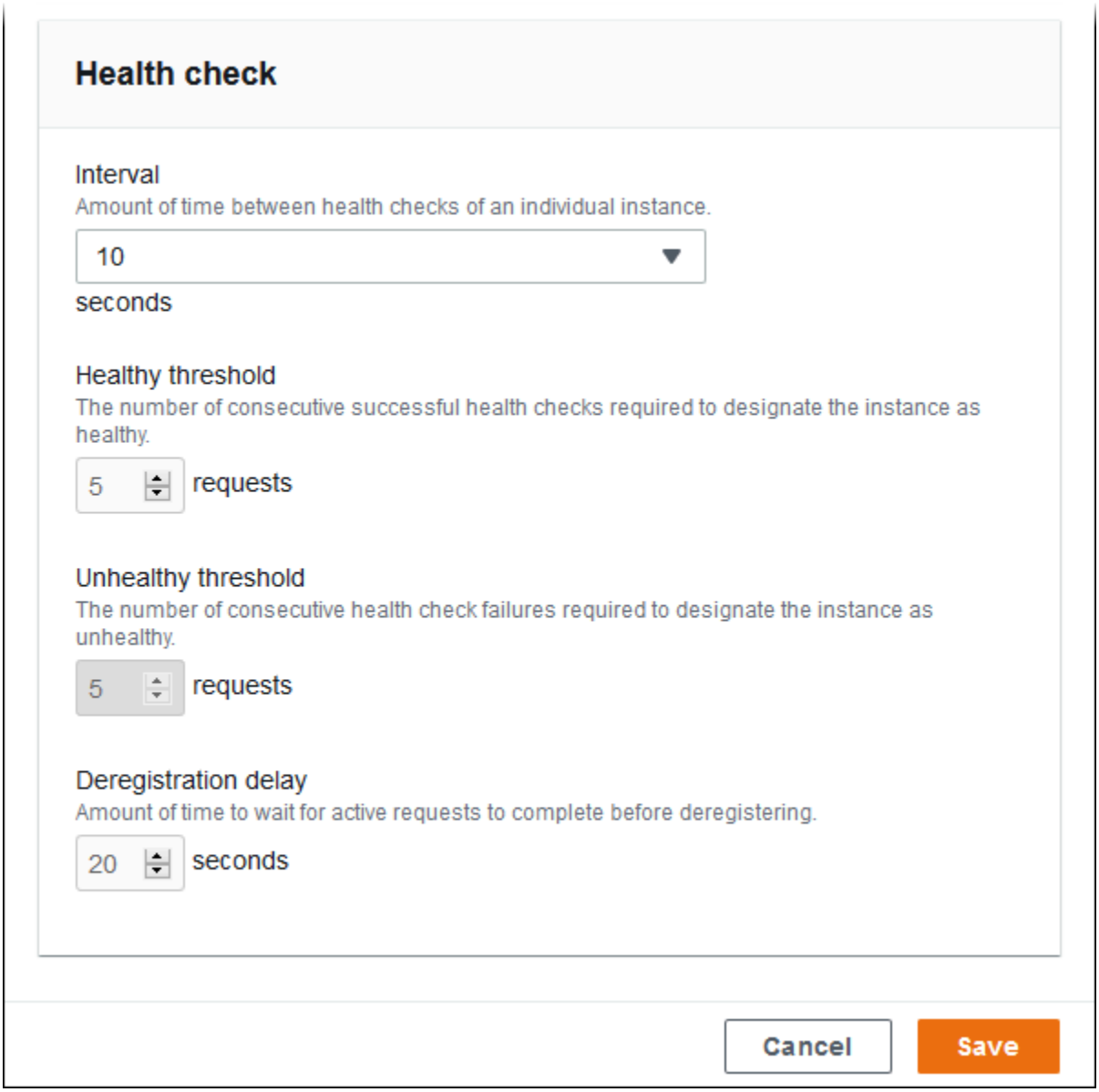
Use estas configurações para definir o processo Name e Process port no qual as solicitações são observadas.



Verificação de integridade

Use as configurações a seguir para configurar as verificações de integridade do processo:

- Interval (Intervalo): a quantidade de tempo, em segundos, entre as verificações de integridade de uma instância individual.
- Healthy threshold (Limite íntegro): o número de verificações de integridade que devem ser aprovadas antes de o Elastic Load Balancing alterar o estado de integridade de uma instância. (Para Network Load Balancer, Unhealthy threshold (Limite não íntegro) é uma configuração somente leitura que é sempre igual ao valor limite de integridade.)
- Deregistration delay (Atraso do cancelamento de registro): a quantidade de tempo, em segundos, para aguardar que as solicitações ativas sejam concluídas antes de cancelar o registro de uma instância.



> **Note**
>
> A verificação de integridade do Elastic Load Balancing não afeta o comportamento da verificação de integridade do grupo de Auto Scaling de um ambiente. As instâncias que falharem em uma verificação de saúde do Elastic Load Balancing não serão substituídas automaticamente pelo Amazon Auto EC2 Scaling, a menos que você configure manualmente o Amazon Auto EC2 Scaling para fazer isso. Para mais detalhes, consulte Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk.

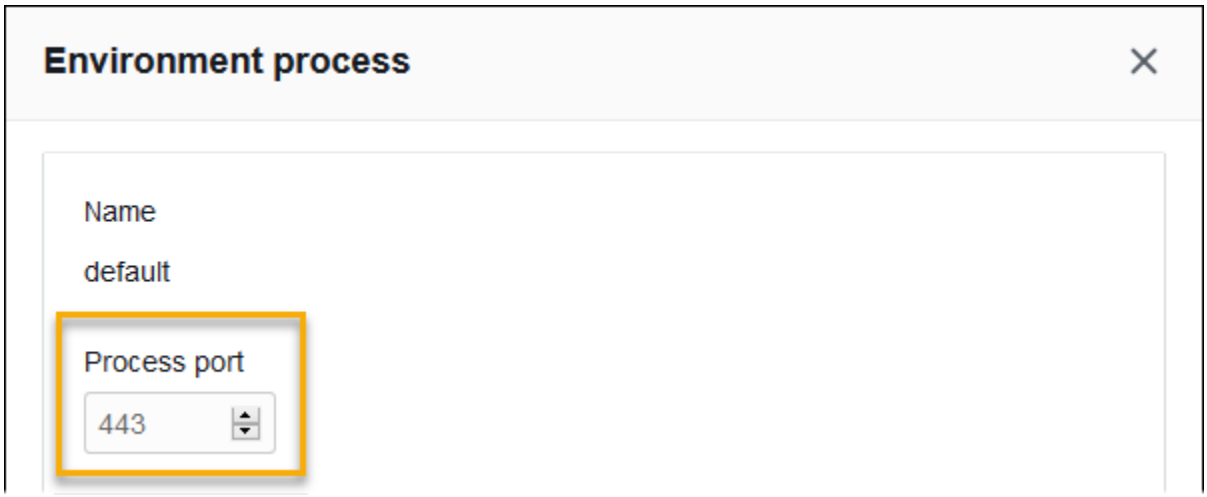
Para obter mais informações sobre verificações de integridade e como elas influenciam a integridade geral do ambiente, consulte Relatórios de integridade básica.

Exemplo: Network Load Balancer para um ambiente com criptografia end-to-end

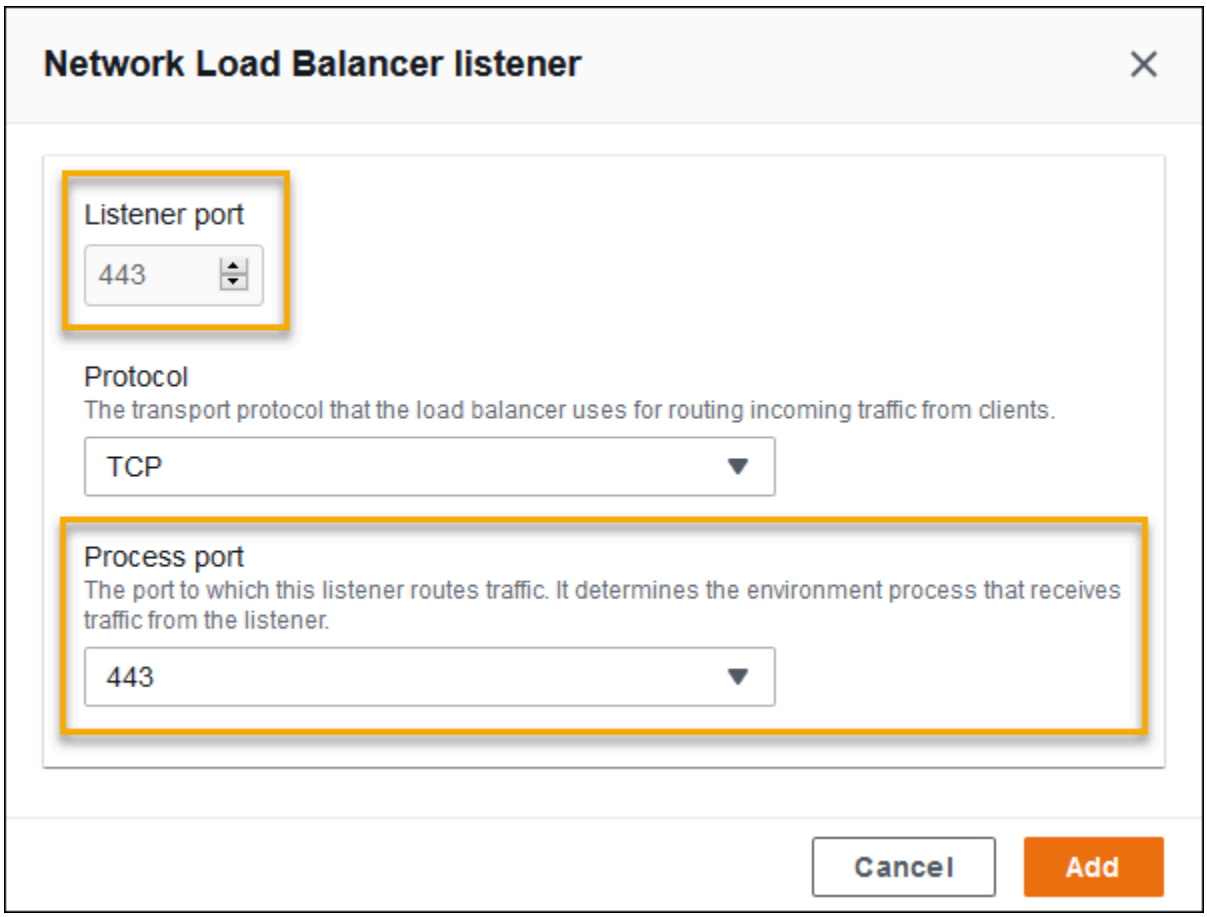
Neste exemplo, seu aplicativo exige criptografia end-to-end de tráfego. Para configurar o Network Load Balancer do ambiente para atender a esses requisitos, configure o processo padrão para escutar na porta 443, adicione um listener à porta 443 que roteia o tráfego para o processo padrão e desabilite o listener padrão.

Para configurar o load balancer neste exemplo

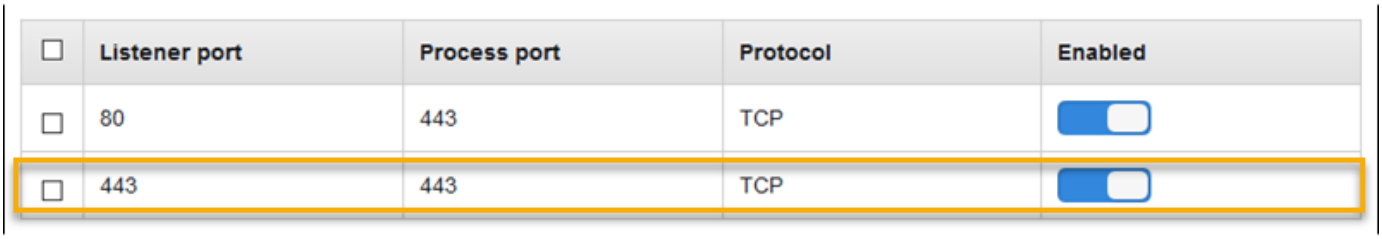
1. Configure o processo padrão. Selecione o processo padrão e, em seguida, em Actions, escolha Edit. Para Process port, digite `443`.



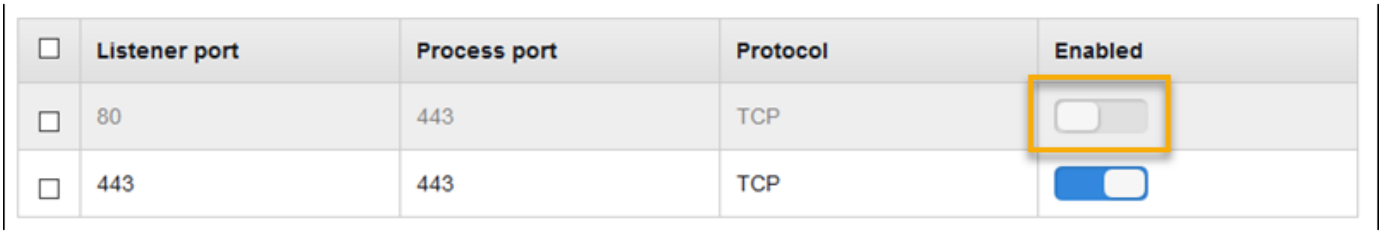
2. Adicione um listener da porta 443. Adicione um novo listener. Para Listener port, digite `443`. Para Process port, certifique-se de selecionar `443`.



Agora você pode ver seu listener adicional na lista.



3. Desative o listener da porta padrão 80. Para o listener padrão, desative a opção Habilitada.



Configurar um Network Load Balancer usando a CLI do EB

A EB CLI solicita que você escolha um tipo de load balancer ao executar o eb create.

```
$ 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
```

Você também pode especificar um tipo de load balancer com a opção `--elb-type`.

```
$ eb create test-env --elb-type network
```

Namespaces do Network Load Balancer

É possível encontrar as configurações relacionadas aos Network Load Balancers nos seguintes namespaces:

- `aws:elasticbeanstalk:environment`: selecione o tipo de balanceador de carga para o ambiente. O valor de um Network Load Balancer é `network`.
- `aws:elbv2:listener`: configure os listeners no Network Load Balancer. Essas configurações são mapeadas para as configurações em `aws:elb:listener` para Classic Load Balancers.
- `aws:elasticbeanstalk:environment:process`: configure as verificações de integridade e especifique a porta e o protocolo para os processos que são executados nas instâncias de seu ambiente. A porta e as configurações de protocolo mapeiam a porta e as configurações de protocolo da instância em `aws:elb:listener` para um listener no Classic Load Balancer. As configurações de verificação de integridade são mapeadas para as configurações nos namespaces `aws:elb:healthcheck` e `aws:elasticbeanstalk:application`.

Example .ebextensions/ .config network-load-balancer

Para começar a usar um Network Load Balancer, use um arquivo de configuração para definir o tipo de balanceador de carga como `network`.

```
option_settings:
  aws:elasticbeanstalk:environment:
```

```
    LoadBalancerType: network
```

> **Note**
>
> Você pode definir o tipo de load balancer apenas durante a criação do ambiente.

Example .ebextensions/ .config nlb-default-process

O arquivo de configuração a seguir modifica as configurações de verificação de integridade no processo padrão.

```
option_settings:
  aws:elasticbeanstalk:environment:process:default:
    DeregistrationDelay: '20'
    HealthCheckInterval: '10'
    HealthyThresholdCount: '5'
    UnhealthyThresholdCount: '5'
    Port: '80'
    Protocol: TCP
```

Example .ebextensions/ .config nlb-secure-listener

O arquivo de configuração a seguir adiciona um listener para o tráfego seguro na porta 443 e um processo de destino correspondente com listener na porta 443.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

A opção `DefaultProcess` é assim denominada em virtude dos Application Load Balancers, que podem ter listeners não padrão na mesma porta para tráfego para caminhos específicos (consulte Application Load Balancer para obter detalhes). Para um Network Load Balancer, a opção especifica o único processo de destino para esse listener.

Neste exemplo, chamamos o processo de `https` porque ele faz o listening do tráfego seguro (HTTPS). O listener envia o tráfego para o processo na porta designada usando o protocolo TCP,

pois o Network Load Balancer funciona apenas com TCP. Isso está correto, porque o tráfego de rede para HTTP e HTTPS é implementado na parte superior do TCP.

Configurar logs de acesso

Use os arquivos de configuração para configurar o balanceador de carga do ambiente a fazer upload de logs de acesso em um bucket do Amazon S3. Consulte os seguintes exemplos de arquivos de configuração em GitHub para obter instruções:

- `loadbalancer-accesslogs-existingbucket.config`: configure o balanceador de carga para fazer upload de logs de acesso em um bucket existente do Amazon S3.
- `loadbalancer-accesslogs-newbucket.config`: configure o balanceador de carga para fazer upload de logs de acesso em um novo bucket.

Adição de um banco de dados ao seu ambiente do Elastic Beanstalk

O Elastic Beanstalk fornece integração acoplada de banco de dados com o Amazon Relational Database Service (Amazon RDS). Você pode usar o Elastic Beanstalk para adicionar um banco de dados MySQL, PostgreSQL, Oracle ou SQL Server ao ambiente durante ou após a criação do ambiente. Quando você adiciona uma instância de banco de dados acoplada a um ambiente, o Elastic Beanstalk fornece as informações de conexão ao seu aplicativo. Ele faz isso definindo as propriedades do ambiente para o nome de host do banco de dados, a porta, o nome de usuário, a senha e o nome do banco de dados.

Vantagens de usar um banco de dados acoplado

Se nunca usou uma instância de banco de dados com sua aplicação, recomendamos que você primeiro use o processo descrito neste tópico para adicionar um banco de dados a um ambiente de teste com o serviço Elastic Beanstalk. Fazendo isso, você pode verificar se a aplicação pode ler as propriedades do ambiente, criar uma sequência de conexão e conectar-se a uma instância de banco de dados, sem o trabalho de configuração adicional necessário para um banco de dados externo ao Elastic Beanstalk.

Considerações ao iniciar a produção

Depois de verificar se a aplicação funciona corretamente com o banco de dados, você pode considerar a possibilidade de passar para um ambiente de produção. Neste ponto, você tem a

opção de desacoplar o banco de dados do ambiente do Elastic Beanstalk para passar para uma configuração que ofereça mais flexibilidade. O banco de dados desacoplado pode permanecer operacional como uma instância de banco de dados externa do Amazon RDS. A integridade do ambiente não é afetada pelo desacoplamento do banco de dados. Se precisar terminar o ambiente, você poderá fazê-lo e também escolher a opção para manter o banco de dados disponível e operacional fora do Elastic Beanstalk.

Vantagens de migrar para um banco de dados desacoplado

Usar um banco de dados externo tem várias vantagens. Você pode se conectar ao banco de dados externo a partir de vários ambientes, usar tipos de banco de dados que não são compatíveis com bancos de dados integrados e realizar blue/green implantações. Como alternativa ao uso de um banco de dados desacoplado criado pelo Elastic Beanstalk, você também pode criar uma instância de banco de dados fora do seu ambiente do Elastic Beanstalk. Ambas as opções resultam em uma instância de banco de dados externa ao seu ambiente do Elastic Beanstalk e exigem configurações adicionais do grupo de segurança e da sequência de conexão. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon RDS.

Seções

- Ciclo de vida do banco de dados
- Adição de uma instância de banco de dados do Amazon RDS ao seu ambiente usando o console
- Conectar ao banco de dados
- Configuração de uma instância de banco de dados do RDS integrada, usando o console
- Configuração de uma instância de banco de dados do RDS integrada usando arquivos de configuração
- Desacoplamento de uma instância de banco de dados do RDS usando o console
- Desacoplamento de uma instância de banco de dados do RDS usando arquivos de configuração

Ciclo de vida do banco de dados

Você pode escolher o que deseja que aconteça com o banco de dados depois de desacoplá-lo do seu ambiente do Elastic Beanstalk. As opções que você pode escolher são coletivamente chamadas de políticas de exclusão. As políticas de exclusão a seguir se aplicarão a um banco de dados após você desacoplá-lo de um ambiente do Elastic Beanstalk ou terminar o ambiente do Elastic Beanstalk.

- Snapshot: o Elastic Beanstalk faz um snapshot do banco de dados antes de terminá-lo. Você pode restaurar um banco de dados de um snapshot quando adicionar uma instância de banco de dados a um ambiente do Elastic Beanstalk ou criar um banco de dados autônomo. Para obter mais informações sobre como criar uma nova instância de banco de dados autônoma de um snapshot, consulte Restoring from a DB snapshot (Restaurar de um snapshot do banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS). Pode haver cobrança do armazenamento de snapshots de banco de dados. Para obter mais informações, consulte a seção Armazenamento de backup em Definição de preço do Amazon RDS.
- Excluir: o Elastic Beanstalk termina o banco de dados. Após ser terminada, a instância do banco de dados não está mais disponível para nenhuma operação.
- Reter: a instância do banco de dados não foi terminada. Ela permanece disponível e operacional, embora desacoplada do Elastic Beanstalk. Então, você pode configurar um ou vários ambientes para conectar ao banco de dados como uma instância de banco de dados externa do Amazon RDS. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon RDS.

Adição de uma instância de banco de dados do Amazon RDS ao seu ambiente usando o console

É possível adicionar uma instância de banco de dados ao seu ambiente usando o console do Elastic Beanstalk.

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

É possível configurar as seguintes opções:

- Snapshot: escolha um snapshot de banco de dados existente. O Elastic Beanstalk restaura o snapshot e o adiciona ao ambiente. O valor padrão é Nenhum. Quando o valor padrão é None

(Nenhum), você pode configurar um novo banco de dados usando as outras configurações dessa página.

- Engine (Mecanismo): escolha um mecanismo de banco de dados.
- Engine version (Versão do mecanismo): escolha uma versão específica do mecanismo de banco de dados.
- Instance class (Classe de instância): escolha uma classe de instância de banco de dados. Para obter informações sobre classes de instância de banco de dados, consulte https://aws.amazon.com/rds/.
- Storage (Armazenamento): escolha a quantidade de armazenamento que deve ser provisionada para seu banco de dados. Você pode aumentar o armazenamento alocado posteriormente, mas não pode diminuí-lo. Para obter mais informações sobre a alocação de armazenamento, consulte Features (Recursos).
- Username (Nome de usuário): insira um nome de usuário de sua escolha usando uma combinação de apenas números e letras.
- Password (Senha): insira uma senha de sua preferência que contenha de 8 a 16 caracteres ASCII imprimíveis (exceto `/`, `\` e `@`).
- Availability (Disponibilidade): escolha High (Multi-AZ) (Alta [Multi-AZ]) para executar um backup quente em uma segunda zona de disponibilidade para alta disponibilidade.
- Database deletion policy (política de exclusão do banco de dados): a política de exclusão determina o que acontece com o banco de dados após ele ser desacoplado do seu ambiente. Pode ser um dos seguintes valores: `Create Snapshot`, `Retain` ou `Delete`. Esses valores são descritos em Ciclo de vida do banco de dados neste mesmo tópico.

> **Note**
>
> O Elastic Beanstalk cria um usuário principal para o banco de dados usando o nome de usuário e a senha fornecidos. Para saber mais sobre o usuário principal e seus privilégios, consulte Master User Account Privileges (Privilégios da conta de usuário principal).

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização é concluída, o novo banco de dados é acoplado ao seu ambiente. O nome do host e outras informações de conexão da instância de banco de dados estão disponíveis para a sua aplicação por meio das seguintes propriedades do ambiente:

Nome da propriedade	Descrição	Valor da propriedade
RDS_HOSTNAME	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
RDS_PORT	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
RDS_DB_NAME	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Conectar ao banco de dados

Use as informações de conectividade para conectar ao banco de dados, de dentro da aplicação, por meio das variáveis do ambiente. Para obter mais informações sobre como usar o Amazon RDS com seus aplicativos, consulte os tópicos a seguir.

- Java SE: Conectar a um banco de dados (plataformas Java SE)
- Java com Tomcat: Conectar a um banco de dados (plataformas Tomcat)
- Node.js: Conectar-se a um banco de dados
- .NET: Conectar-se a um banco de dados

- PHP: Conectando-se a um banco de dados com um PDO ou My SQLi
- Python: Conectar-se a um banco de dados
- Ruby: Conectar-se a um banco de dados

Configuração de uma instância de banco de dados do RDS integrada, usando o console

É possível visualizar e modificar as configurações de sua instância de banco de dados na seção Database (Banco de dados) na página Configuration (Configuração) do ambiente no console do Elastic Beanstalk.

Como configurar a instância de banco de dados do seu ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).

Você pode modificar as configurações de Instance class (Classe de instância), Storage (Armazenamento), Password (Senha), Availability (Disponibilidade) e Database deletion policy (política de exclusão de banco de dados) após a criação do banco de dados. Se você alterar a classe de instância, o Elastic Beanstalk provisionará novamente a instância de banco de dados.

Se você não precisa mais do Elastic Beanstalk para associar o banco de dados ao ambiente, pode optar por desacoplá-lo selecionando Decouple database (Desacoplar banco de dados). É importante entender as opções e as considerações envolvidas nessa operação. Para obter mais informações, consulte the section called “Desacoplamento de uma instância de banco de dados do RDS usando o console”.

> **Aviso**
>
> Não modifique as configurações na instância do banco de dados desacoplado fora da funcionalidade que é fornecida pelo Elastic Beanstalk (por exemplo, no console do Amazon RDS). Se fizer isso, a configuração de banco de dados do Amazon RDS poderá ficar fora de sincronia com a definição do ambiente. Quando você atualiza ou reinicia o ambiente, as

configurações especificadas no ambiente substituem todas as configurações feitas fora do Elastic Beanstalk.
Se você precisar modificar as configurações não compatíveis diretamente com o Elastic Beanstalk, use arquivos de configuraçãodo Elastic Beanstalk.

Configuração de uma instância de banco de dados do RDS integrada usando arquivos de configuração

Você pode configurar a instância de banco de dados do seu ambiente usando arquivos de configuração. Use as opções no namespace `aws:rds:dbinstance`. O exemplo a seguir modifica o tamanho de armazenamento alocado do banco de dados para 100 GB.

Example .ebextensions/ .config db-instance-options

```
option_settings:
  aws:rds:dbinstance:
    DBAllocatedStorage: 100
```

Se quiser configurar propriedades de instância de banco de dados não compatíveis com o Elastic Beanstalk, você ainda pode usar um arquivo de configuração e especificar as configurações usando a chave `resources`. O exemplo a seguir define valores para as propriedades `StorageType` e `Iops` do Amazon RDS.

Example .ebextensions/ .config db-instance-properties

```
Resources:
  AWSEBRDSDatabase:
    Type: AWS::RDS::DBInstance
    Properties:
      StorageType:io1
      Iops: 1000
```

Desacoplamento de uma instância de banco de dados do RDS usando o console

Você pode desacoplar seu banco de dados de um ambiente do Elastic Beanstalk sem afetar a integridade do ambiente. Considere os seguintes requisitos antes de desacoplar o banco de dados:

- O que deverá acontecer com o banco de dados depois que ele for desacoplado?

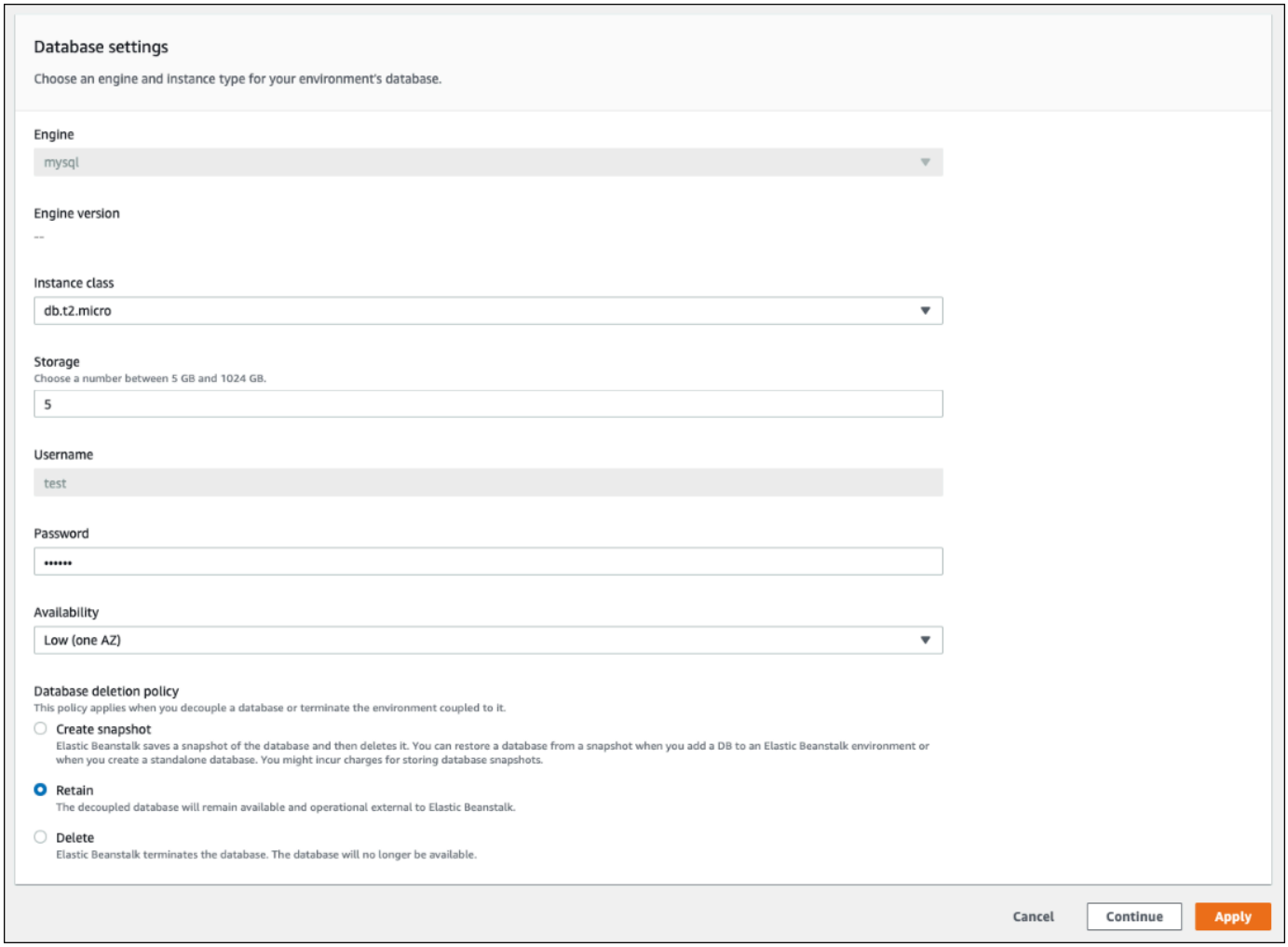
  Você pode optar por criar um snapshot do banco de dados e depois terminá-lo, manter o banco de dados operacional como um banco de dados autônomo externo ao Elastic Beanstalk ou excluir permanentemente o banco de dados. A configuração Database deletion policy (política de exclusão do banco de dados) determina esse resultado. Para obter uma descrição detalhada das políticas de exclusão, consulte Ciclo de vida do banco de dados neste mesmo tópico.

- Você precisa fazer algumas alterações nas configurações do banco de dados antes de desacoplá-lo?

  Se precisar fazer alterações de configuração no banco de dados, você deverá aplicá-las antes de desacoplar o banco de dados. Isso inclui alterações na Database deletion policy (política de exclusão do banco de dados). Alterações pendentes que sejam enviadas ao mesmo tempo que a configuração Decouple database (Desacoplar banco de dados) serão ignoradas, e somente a configuração de desacoplamento será aplicada.

Para desacoplar uma instância de banco de dados de um ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Revise todos os valores das configurações na seção Database settings (Configurações do banco de dados), especialmente Database deletion policy (política de exclusão do banco de dados), que determina o que acontece com o banco de dados após ele ser desacoplado.



Se todas as outras configurações estiverem corretas, pule para a Etapa 6 para desacoplar o banco de dados.

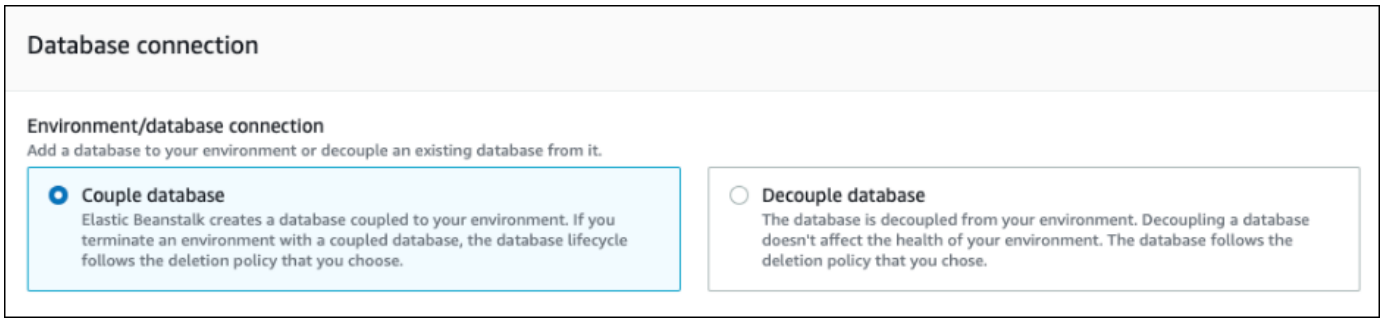
> **Warning**
>
> É importante aplicar a configuração Database deletion policy (política de exclusão do banco de dados) separada de Decouple database (Desacoplar banco de dados). Se você selecionar Apply (Aplicar) com a intenção de salvar Decouple database (Desacoplar banco de dados) e uma política de exclusão do banco de dados recém-selecionada, a nova política de exclusão escolhida será ignorada. O Elastic Beanstalk desacoplará o banco de dados seguindo a política de exclusão definida anteriormente. Se a política de exclusão definida anteriormente for `Delete` ou `Create Snapshot`,

você correrá o risco de perder o banco de dados em vez de seguir a política pendente pretendida.

Se alguma das configurações exigir atualizações, faça o seguinte:

1. Faça as modificações necessárias no painel Database settings (Configurações do banco de dados).
2. Escolha Aplicar. Salvar as alterações de configuração do banco de dados levará alguns minutos.
3. Volte para a Etapa 3 e escolha Configuration (Configuração) no painel de navegação.

6. Vá para a seção Database connection (Conexão do banco de dados) do painel.



7. Escolha Decouple database (Desacoplar banco de dados).
8. Escolha Apply (Aplicar) para iniciar a operação de desacoplamento do banco de dados.

A configuração de política de exclusão determina o resultado para o banco de dados e o tempo necessário para desacoplá-lo.

- Se a política de exclusão for definida como `Delete`, o banco de dados será excluído. A operação pode levar aproximadamente de 10 a 20 minutos, dependendo do tamanho do banco de dados.
- Se a política de exclusão for definida como `Snapshot`, um snapshot do banco de dados será criado. Em seguida, o banco de dados será excluído. O tempo necessário para esse processo varia de acordo com o tamanho do banco de dados.
- Se a política de exclusão for definida como `Retain`, o banco de dados permanecerá operacional, fora do ambiente do Elastic Beanstalk. Geralmente, desacoplar um banco de dados leva menos de cinco minutos.

Se decidiu reter o banco de dados externo ao seu ambiente do Elastic Beanstalk, você precisará realizar etapas adicionais para configurá-lo. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon RDS. Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte DB Instance Storage (Armazenamento de instâncias de banco de dados) e Modifying a DB instance (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

Desacoplamento de uma instância de banco de dados do RDS usando arquivos de configuração

Você pode desacoplar sua instância de banco de dados de um ambiente do Elastic Beanstalk sem afetar a integridade do ambiente. A instância de banco de dados segue a política de exclusão de banco de dados que foi aplicada quando o banco de dados foi desacoplado.

Ambas as opções necessárias para desacoplar o banco de dados estão no namespace the section called “aws:rds:dbinstance”. Eles são os seguintes:

- A opção `DBDeletionPolicy` define a política de exclusão. Pode ser um dos seguintes valores: `Snapshot`, `Delete` ou `Retain`. Esses valores são descritos em Ciclo de vida do banco de dados neste mesmo tópico.
- A opção `HasCoupledDatabase` determina se o ambiente tem um banco de dados acoplado.
  - Se definida como `true`, o Elastic Beanstalk cria uma nova instância de banco de dados acoplada ao seu ambiente.
  - Se definida como `false`, o Elastic Beanstalk começa a desacoplar a instância de banco de dados do seu ambiente.

Se quiser alterar a configuração do banco de dados antes de desacoplá-la, aplique todas as alterações de configuração primeiro, em uma operação separada. Isso inclui alterar a configuração de `DBDeletionPolicy`. Depois que as alterações forem aplicadas, execute um comando separado para definir a opção de desacoplamento. Se você enviar outras configurações ao mesmo tempo que a configuração de desacoplamento, as outras configurações da opção serão ignoradas enquanto a configuração de desacoplamento for aplicada.

> **Warning**
>
> É importante que você execute os comandos para aplicar as configurações `DBDeletionPolicy` e `HasCoupledDatabase` como duas operações separadas. Se a política de exclusão ativa já estiver definida como `Delete` ou `Snapshot`, você correrá o risco de perder o banco de dados. O banco de dados segue a política de exclusão que está ativa no momento, em vez da política de exclusão pendente que você pretendia.

Para desacoplar uma instância de banco de dados de um ambiente

Siga estas etapas para desacoplar o banco de dados do ambiente do Elastic Beanstalk. Você pode usar o EB CLI ou o AWS CLI para concluir as etapas. Para obter mais informações, consulte Personalização avançada do ambiente com arquivos de configuração.

1. Se quiser alterar a política de exclusão, configure um arquivo de configuração no formato a seguir. Neste exemplo, a política de exclusão está definida para reter.

   Example

   ```
   option_settings:
     aws:rds:dbinstance:
       DBDeletionPolicy: Retain
   ```

2. Execute o comando usando sua ferramenta preferida para atualizar a configuração.
3. Configure um arquivo de configuração para definir `HasCoupledDatabase` como `false`.

   Example

   ```
   option_settings:
     aws:rds:dbinstance:
       HasCoupledDatabase: false
   ```

4. Execute o comando usando sua ferramenta preferida para atualizar a configuração.

A configuração de política de exclusão determina o resultado para o banco de dados e o tempo necessário para desacoplá-lo.

- Se a política de exclusão for definida como `Delete`, o banco de dados será excluído. A operação pode levar aproximadamente de 10 a 20 minutos, dependendo do tamanho do banco de dados.
- Se a política de exclusão for definida como `Snapshot`, um snapshot do banco de dados será criado. Em seguida, o banco de dados será excluído. O tempo necessário para esse processo varia de acordo com o tamanho do banco de dados.
- Se a política de exclusão for definida como `Retain`, o banco de dados permanecerá operacional, fora do ambiente do Elastic Beanstalk. Geralmente, desacoplar um banco de dados leva menos de cinco minutos.

Se decidiu reter o banco de dados externo ao seu ambiente do Elastic Beanstalk, você precisará realizar etapas adicionais para configurá-lo. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon RDS. Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte DB Instance Storage (Armazenamento de instâncias de banco de dados) e Modifying a DB instance (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

A segurança AWS Elastic Beanstalk do seu ambiente

O Elastic Beanstalk fornece várias opções que controlam o acesso ao serviço (segurança) do seu ambiente e das instâncias da Amazon nele contidas. EC2 Este tópico discute a configuração dessas opções.

Seções

- Configurar a segurança do ambiente
- Namespaces de configuração de segurança do ambiente

Configurar a segurança do ambiente

É possível modificar a configuração de segurança do ambiente do Elastic Beanstalk no console do Elastic Beanstalk.

Para configurar o acesso ao serviço (segurança) do ambiente no console do Elastic Beanstalk

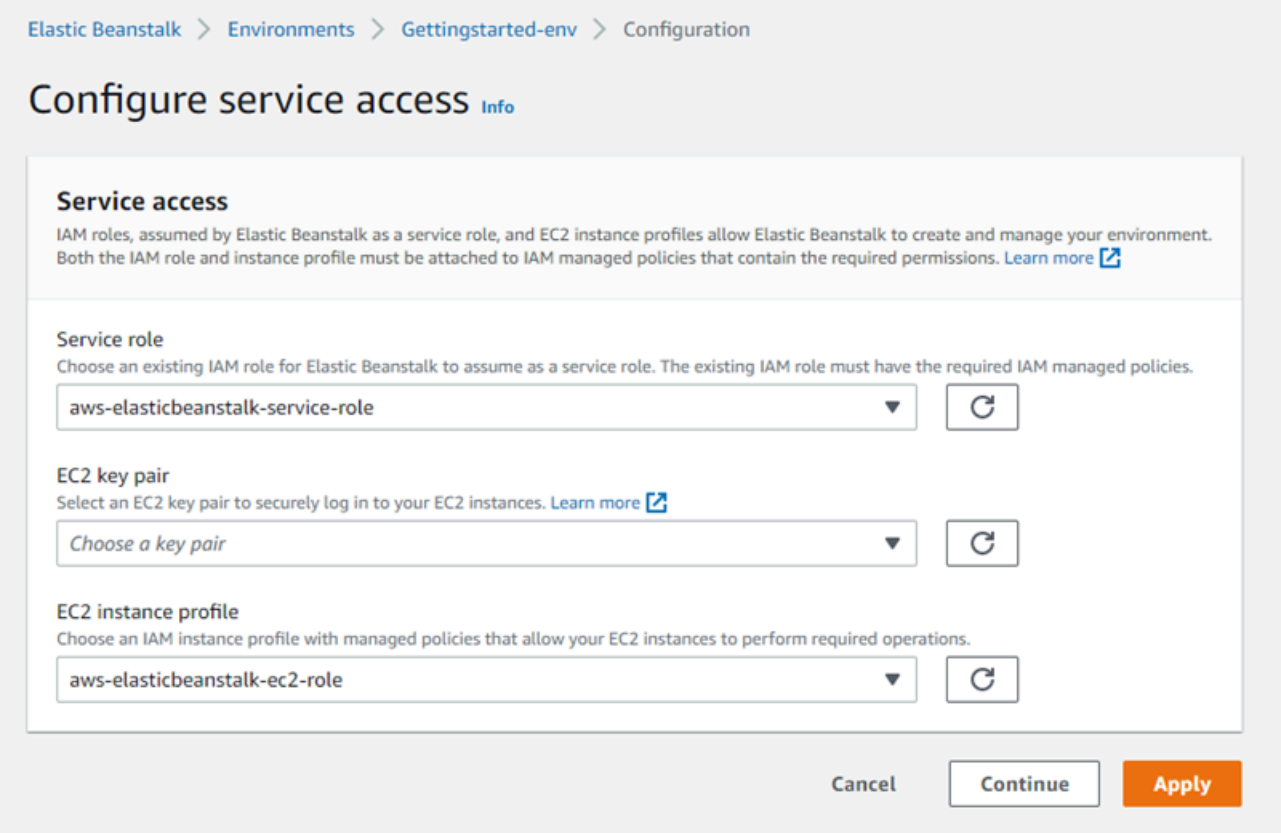
1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Service access (Acesso ao serviço), escolha Edit (Editar).

As seguintes configurações estão disponíveis.

Configurações

- Perfil de serviço
- EC2 key pair
- Perfil de instância do IAM



Perfil de serviço

Selecione uma função de serviço a ser associada ao ambiente do Elastic Beanstalk. O Elastic Beanstalk assume a função de serviço quando acessa AWS outros serviços em seu nome. Para obter detalhes, consulte Gerenciar funções de serviço do Elastic Beanstalk.

EC2 key pair

Você pode fazer login com segurança nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) provisionadas para seu aplicativo Elastic Beanstalk com um par de chaves da Amazon. EC2 Para obter instruções sobre como criar um par de chaves, consulte Como criar um par de chaves usando a Amazon EC2 no Guia EC2 do usuário da Amazon.

> **Note**
>
> Quando você cria um par de chaves, a Amazon EC2 armazena uma cópia da sua chave pública. Se você não precisar mais usá-lo para se conectar a nenhuma instância do ambiente, poderá excluí-lo da Amazon EC2. Para obter detalhes, consulte Excluindo seu par de chaves no Guia do EC2 usuário da Amazon.

Escolha um EC2 key pair no menu suspenso para atribuí-lo às instâncias do seu ambiente. Quando você atribui um par de chaves, a chave pública é armazenada na instância para autenticar a chave privada, que você armazena localmente. A chave privada nunca é armazenada AWS.

Para obter mais informações sobre como se conectar às EC2 instâncias da Amazon, consulte Connect to Your Instance e Connecting to Linux/UNIX Instances from Windows using PuTTY no Guia EC2 do usuário da Amazon.

Perfil de instância do IAM

Um perfil de EC2 instância é uma função do IAM aplicada às instâncias lançadas em seu ambiente do Elastic Beanstalk. EC2 As instâncias da Amazon assumem a função de perfil de instância para assinar solicitações AWS e acessar APIs, por exemplo, o upload de registros para o Amazon S3.

Na primeira vez que você cria um ambiente no console do Elastic Beanstalk, o Elastic Beanstalk solicita que você crie um perfil de instância com um conjunto padrão de permissões. Você pode adicionar permissões a esse perfil para fornecer às suas instâncias acesso a outros AWS serviços. Para obter detalhes, consulte Gerenciar perfis de instância do Elastic Beanstalk.

> **Note**
>
> Anteriormente, o Elastic Beanstalk criava EC2 um `aws-elasticbeanstalk-ec2-role` perfil de instância padrão chamado na primeira AWS vez que uma conta criava um ambiente.

Esse perfil de instância incluía as políticas gerenciadas padrão. Se sua conta já tiver esse perfil de instância, ele permanecerá disponível para você atribuí-lo aos seus ambientes. No entanto, as diretrizes de AWS segurança recentes não permitem que um AWS serviço crie automaticamente funções com políticas de confiança para outros AWS serviços, EC2 nesse caso. Por causa dessas diretrizes de segurança, o Elastic Beanstalk não cria mais um perfil de instância padrão `aws-elasticbeanstalk-ec2-role`.

> **Note**
>
> Há outro aspecto da segurança de EC2 instâncias que designa regras de firewall para EC2 instâncias. Isso é controlado por grupos EC2 de segurança. Para obter mais informações, consulte As EC2 instâncias da Amazon para seu ambiente Elastic Beanstalk.

Namespaces de configuração de segurança do ambiente

O Elastic Beanstalk fornece opções de configuração nos namespaces a seguir para permitir que você personalize a segurança do ambiente:

- `aws:elasticbeanstalk:environment`: configure a função de serviço do ambiente usando a opção `ServiceRole`.
- `aws:autoscaling:launchconfiguration`— Configure permissões para as EC2 instâncias Amazon do ambiente usando `SecurityGroups` as opções `EC2KeyName` `IamInstanceProfile`,`DisableDefaultEC2SecurityGroup`, e.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Marcar recursos em seus ambientes do Elastic Beanstalk

Você pode aplicar tags aos seus AWS Elastic Beanstalk ambientes. As tags são pares de valores-chave associados AWS aos recursos. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte Marcar recursos da aplicação do Elastic Beanstalk.

O Elastic Beanstalk aplica tags de ambiente ao próprio recurso do ambiente, bem como a outros AWS recursos que o Elastic Beanstalk cria para o ambiente. É possível usar tags para gerenciar permissões no nível de recurso específico em um ambiente. Para obter mais informações, consulte Como marcar seus EC2 recursos da Amazon no Guia do EC2 usuário da Amazon.

Por padrão, o Elastic Beanstalk aplica algumas tags ao seu ambiente:

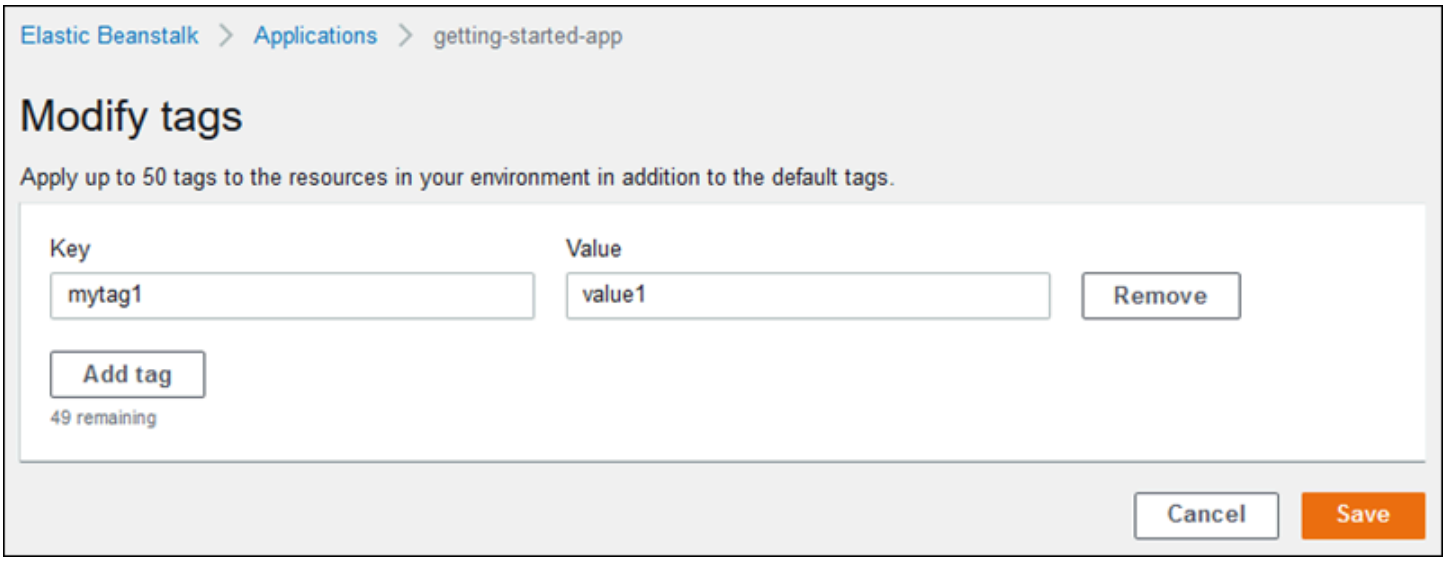
- `elasticbeanstalk:environment-name`: o nome do ambiente.
- `elasticbeanstalk:environment-id`: o ID do ambiente.
- `Name`— Também o nome do ambiente. `Name`é usado no EC2 painel da Amazon para identificar e classificar recursos.

Você não pode editar essas etiquetas padrão.

É possível especificar tags ao criar o ambiente do Elastic Beanstalk. Em um ambiente existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Um ambiente pode ter até 50 tags, incluindo as tags padrão.

Adicionar tags durante a criação do ambiente

Ao usar o console do Elastic Beanstalk para criar um ambiente, é possível especificar chaves e valores de tags na página Modify tags (Modificar tags) do assistente Create New Environment (Criar ambiente).



Se você usar a CLI do EB para criar um ambiente, use a opção `--tags` com eb create para adicionar tags.

```
~/workspace/my-app$ eb create --tags mytag1=value1,mytag2=value2
```

Com o AWS CLI ou outros clientes baseados em API, use o `--tags` parâmetro no 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
```

As configurações salvas incluem as tags definidas pelo usuário. Quando você aplica uma configuração salva que contém tags durante a criação do ambiente, essas tags são aplicadas ao novo ambiente, desde que você não especifique nenhuma tag nova. Se você adicionar tags a um ambiente usando um dos métodos anteriores, qualquer tag definida na configuração salva será descartada.

Gerenciar tags de um ambiente existente

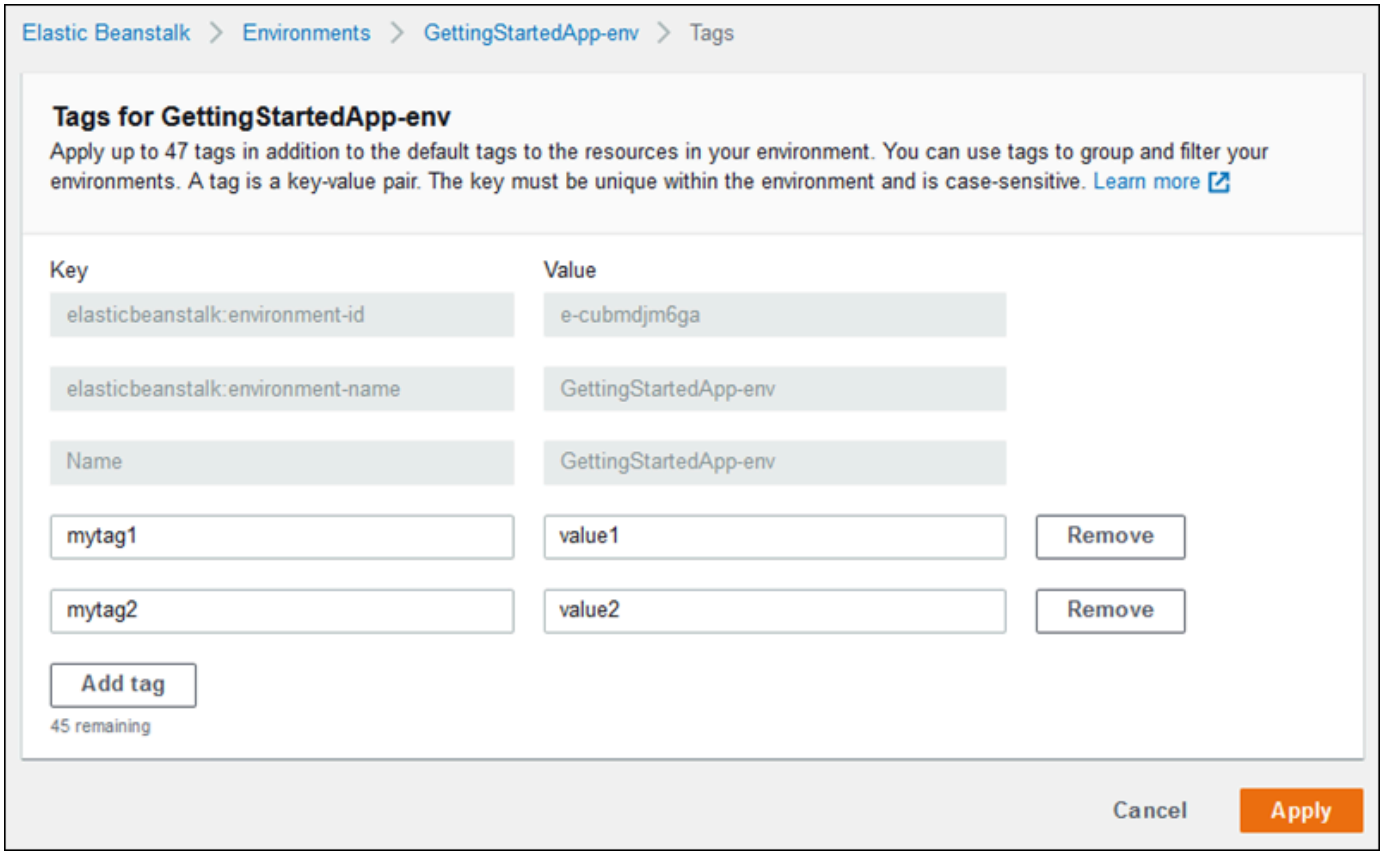
É possível adicionar, atualizar e excluir tags em um ambiente existente do Elastic Beanstalk. O Elastic Beanstalk aplica as alterações aos recursos do seu ambiente.

No entanto, não é possível editar as tags padrão que o Elastic Beanstalk aplica a seu ambiente.

Como gerenciar as tags de um ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, selecione Tags.

   A página de gerenciamento de tags mostra a lista de tags que existem atualmente no ambiente.



4. Adicione, atualize ou exclua tags:

   - Para adicionar uma tag, insira-a nas caixas vazias na parte inferior da lista. Para adicionar outra tag, escolha Add tag (Adicionar tag) e o Elastic Beanstalk adicionará outro par de caixas vazias.
   - Para atualizar uma chave ou valor da tag, edite a caixa respectiva na linha da tag.
   - Para excluir uma tag, escolha Remove (Remover) ao lado da caixa de valor da tag.

5. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar o ambiente, use eb tags para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em seu ambiente padrão.

```
~/workspace/my-app$ eb tags --list
```

O comando a seguir atualiza a tag `mytag1` e exclui a tag `mytag2`.

```
~/workspace/my-app$ eb tags --update mytag1=newvalue --delete mytag2
```

Para obter uma lista de opções e mais exemplos, consulte `eb tags`.

Com o AWS CLI ou outros clientes baseados em API, use o list-tags-for-resource comando para listar as tags de um ambiente.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:environment/my-app/my-env"
```

Use o comando update-tags-for-resource para adicionar, atualizar ou excluir tags em um 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"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do update-tags-for-resource. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

> **Note**
>
> Para usar esses dois AWS CLI comandos com um ambiente do Elastic Beanstalk, você precisa do ARN do ambiente. Você pode recuperar o ARN usando o comando a seguir.
>
> ```
> $ aws elasticbeanstalk describe-environments
> ```

Variáveis de ambiente e outras configurações de software

A página Configurar atualizações, monitoramento e configuração de registro permite que você configure o software nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que executam seu aplicativo. Você pode definir variáveis de ambiente, AWS X-Ray depuração, armazenamento e streaming de registros de instâncias e configurações específicas da plataforma.

Tópicos

- Definir configurações específicas da plataforma
- Configurar propriedades do ambiente (variáveis de ambiente)

- Namespaces de configurações de software
- Acessar propriedades de ambiente
- Configurando a depuração AWS X-Ray
- Visualizar seus logs de ambiente do Elastic Beanstalk

Definir configurações específicas da plataforma

Além do conjunto padrão de opções disponíveis para todos os ambientes, a maioria das plataformas do Elastic Beanstalk permite que você use configurações específicas de linguagens ou de estruturas de trabalho. Eles aparecem na seção Plataforma de software da página Configurar atualizações, monitoramento e logs e podem assumir as seguintes formas.

- Propriedades predefinidas do ambiente: a Plataforma Ruby utiliza propriedades do ambiente para configurações de estruturas de trabalho, como `RACK_ENV` e `BUNDLE_WITHOUT`.
- Propriedades de placeholder do ambiente: a plataforma Tomcat define uma propriedade de ambiente chamada `JDBC_CONNECTION_STRING` que não tem nenhum valor definido. Esse tipo de configuração era mais comuns em versões mais antigas da plataforma.
- Opções de configuração: a maioria das plataformas define as opções de configuração em namespaces específicos da plataforma ou compartilhados, como `aws:elasticbeanstalk:xray` ou `aws:elasticbeanstalk:container:python`.

Como definir configurações específicas da plataforma no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Em Software da plataforma), faça as alterações necessárias na configuração da opção.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para obter mais informações sobre as opções específicas da plataforma e sobre como obter valores de propriedade de ambiente no seu código, consulte o tópico da plataforma para sua linguagem ou estrutura:

- Docker: the section called “Configuração de ambientes”
- Go: Usar a plataforma Go do Elastic Beanstalk
- Java SE: Usar a plataforma Java SE do Elastic Beanstalk
- Tomcat: Usar a plataforma Tomcat do Elastic Beanstalk
- .NET Core no Linux: Usar o Elastic Beanstalk .NET Core na plataforma Linux
- .NET: Usar o Elastic Beanstalk .NET na plataforma Windows
- Node.js: Usar a plataforma Node.js do Elastic Beanstalk
- PHP: Usar a plataforma PHP do Elastic Beanstalk
- Python: Usar a plataforma Python do Elastic Beanstalk
- Ruby: Usar a plataforma Ruby do Elastic Beanstalk

Configurar propriedades do ambiente (variáveis de ambiente)

Você pode usar propriedades de ambiente (também conhecidas como variáveis de ambiente) para transmitir endpoints, configurações de depuração e outras informações para seu aplicativo. As variáveis de ambiente ajudam você a executar seu aplicativo em vários ambientes para diferentes propósitos, como desenvolvimento, teste, preparação e produção.

Além disso, quando você adiciona um banco de dados ao seu ambiente, o Elastic Beanstalk define variáveis de ambiente, `RDS_HOSTNAME` como as que você pode ler no código do seu aplicativo para construir um objeto de conexão ou uma string.

Para configurar variáveis de ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até Variáveis de ambiente de tempo de execução.
6. Selecione Adicionar variável de ambiente.
7. Em Fonte, selecione Texto sem formatação.

> **Note**
>
> Os valores do Secrets Manager e do SSM Parameter Store no menu suspenso servem para configurar variáveis de ambiente como segredos para armazenar dados confidenciais, como credenciais e chaves de API. Para obter mais informações, consulte Usando o Elastic AWS Secrets ManagerAWS Systems Manager Beanstalk com um Parameter Store.

8. Insira o nome da variável de ambiente e os pares de valores da variável de ambiente.
9. Se precisar adicionar mais variáveis, repita a Etapa 6 até a Etapa 8.
10. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Limites de propriedades de ambiente

- As chaves podem conter caracteres alfanuméricos e os seguintes símbolos: `_ . : / + \ - @`

  Os símbolos listados são válidos para chaves de propriedade do ambiente, mas podem não ser válidos para nomes de variáveis de ambiente na plataforma do seu ambiente. Para compatibilidade com todas as plataformas, limite as propriedades do ambiente ao padrão a seguir: `[A-Z_][A-Z0-9_]*`
- Os valores podem conter caracteres alfanuméricos, espaço em branco e os seguintes símbolos: `_ . : / = + \ - @ ' "`

  > **Note**
  >
  > Alguns caracteres em valores de propriedades de ambiente devem ser escapados. Use o caractere de barra invertida (\) para representar alguns caracteres especiais e de controle. A lista a seguir inclui exemplos para representar alguns caracteres que precisam ser escapados:
  >
  > - barra invertida (\): para representar, use `\\`
  > - aspas simples (`'`): para representar, use `\'`
  > - aspas duplas (`"`): para representar, use `\"`
- As chaves e os valores diferenciam letras maiúsculas de minúsculas.

- O tamanho combinado de todas as propriedades do ambiente não pode exceder 4.096 bytes quando armazenado como strings com o formato =. *key value*

Namespaces de configurações de software

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

É possível usar os arquivos de configuração do Elastic Beanstalk para definir propriedades do ambiente e opções de configuração em seu código-fonte. Use o `aws:elasticbeanstalk:application:environment` namespace para definir propriedades do ambiente.

Example .ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    API_ENDPOINT: www.example.com/api
```

Se você usar arquivos de configuração ou AWS CloudFormation modelos para criar recursos personalizados, poderá usar uma AWS CloudFormation função para obter informações sobre o recurso e atribuí-lo dinamicamente a uma propriedade do ambiente durante a implantação. O exemplo a seguir do elastic-beanstalk-samples GitHub repositório usa a função Ref para obter o ARN de um tópico do Amazon SNS que ele cria e o atribui a uma propriedade de ambiente chamada. `NOTIFICATION_TOPIC`

> **Observações**
>
> - Se você usar uma AWS CloudFormation função para definir uma propriedade de ambiente, o console do Elastic Beanstalk exibirá o valor da propriedade antes que a função seja avaliada. Você pode usar o script de plataforma `get-config` para confirmar os valores das propriedades do ambiente que estão disponíveis para seu aplicativo.
> - A plataforma Multicontainer Docker não é usada AWS CloudFormation para criar recursos de contêiner. Como resultado, essa plataforma não oferece suporte à definição de propriedades do ambiente usando AWS CloudFormation funções.

Example .Ebextensions/sns-topic.config

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

option_settings:
  aws:elasticbeanstalk:application:environment:
    NOTIFICATION_TOPIC: '`{"Ref" : "NotificationTopic"}`'
```

Você também pode usar esse recurso para propagar informações de pseudoparâmetros do AWS CloudFormation. Este exemplo obtém a região atual e a atribui a uma propriedade chamada `AWS_REGION`.

Example .Ebextensions/env-regionname.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    AWS_REGION: '`{"Ref" : "AWS::Region"}`'
```

A maioria das plataformas do Elastic Beanstalk define namespaces adicionais com opções para configurar o software que é executado na instância, como o proxy inverso que retransmite solicitações para sua aplicação. Para obter mais informações sobre os namespaces disponíveis para a sua plataforma, consulte a seguir:

- Go: Namespace de configuração do Go
- Java SE: Namespace de configuração do Java SE
- Tomcat: Namespaces de configuração do Tomcat
- .NET Core no Linux: Namespace de configuração do .NET Core no Linux
- .NET: O namespace aws:elasticbeanstalk:container:dotnet:apppool
- Node.js: Namespace de configuração do Node.js
- PHP: Namespaces para configuração
- Python: Namespaces de configuração do Python
- Ruby: Namespaces de configuração do Ruby

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console,

configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Acessar propriedades de ambiente

Na maioria dos casos, você acessa as propriedades do ambiente no código do seu aplicativo, como uma variável de ambiente. Porém, em geral, as propriedades do ambiente são transferidas apenas para o aplicativo e não podem ser visualizadas conectando uma instância em seu ambiente e executando-a `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`

  As versões da plataforma Tomcat lançadas em ou após 26 de março de 2025 também podem ser usadas `System.getenv` para acessar variáveis de ambiente de texto simples. Você pode continuar usando `System.getProperty` para acessar variáveis de ambiente de texto simples. No entanto, as variáveis de ambiente armazenadas como segredos só estão disponíveis usando`System.getenv`.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

```
String endpoint = System.getenv("API_ENDPOINT");
```

> **Important**
>
> A adição do `System.getenv` acesso às variáveis de ambiente nas versões da plataforma Tomcat lançadas em ou após 26 de março de 2025 pode causar um comportamento inesperado em aplicativos que dão precedência às variáveis de ambiente sobre as propriedades do sistema Java ou ao mudar explicitamente de para. `System.getProperty` `System.getenv` Para obter mais informações e ações recomendadas, consulte Usando a plataforma Elastic Beanstalk Tomcat.

- .NET Core no 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`

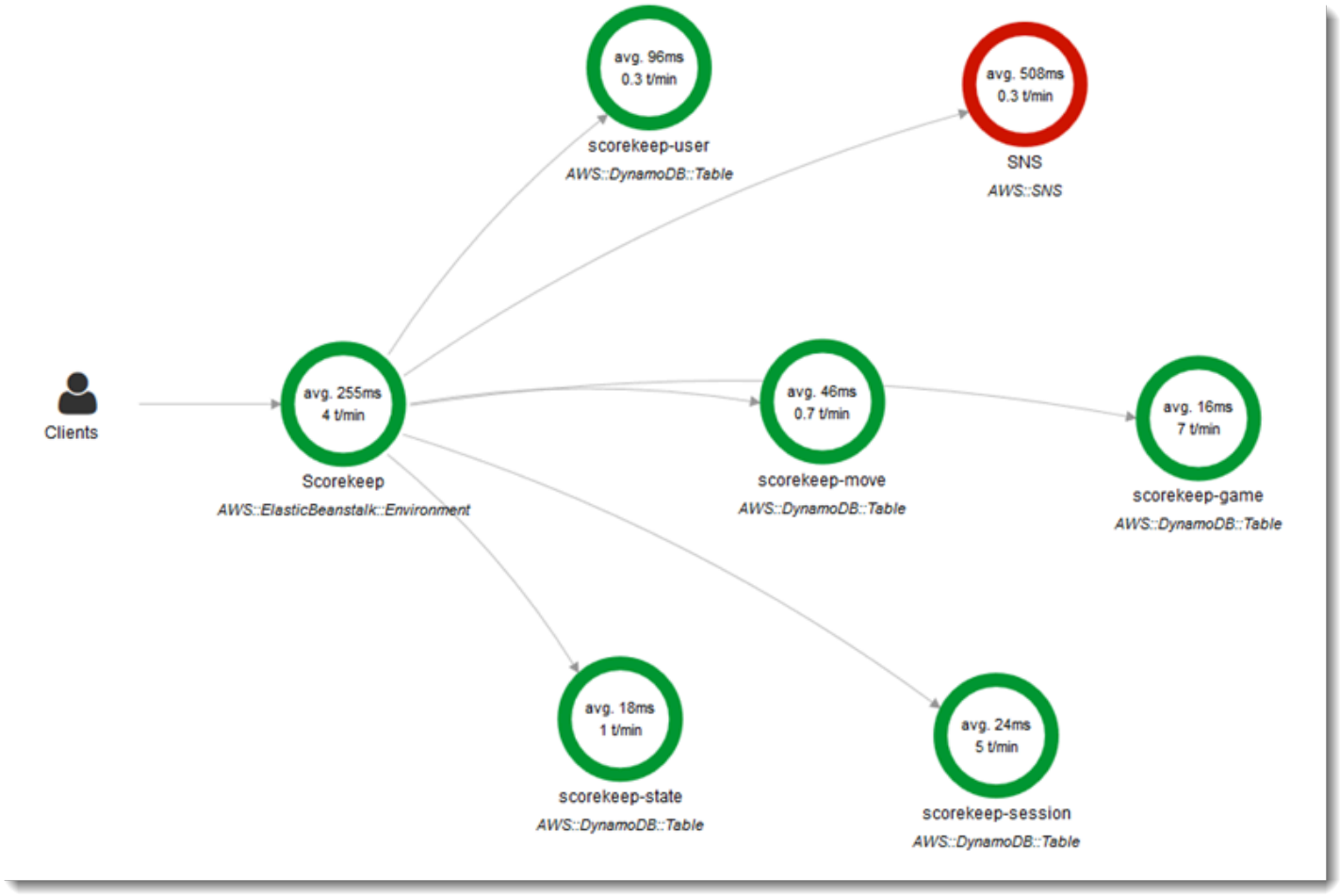
```
endpoint = ENV['API_ENDPOINT']
```

Fora do código do aplicativo, como em um script que é executado durante a implantação, você pode acessar as propriedades do ambiente com a `get-config` Plataforma de script. Consulte o elastic-beanstalk-samples GitHub repositório para ver exemplos de configurações que usam. `get-config`

Configurando a depuração AWS X-Ray

Você pode usar o AWS Elastic Beanstalk console ou um arquivo de configuração para executar o AWS X-Ray daemon nas instâncias do seu ambiente. X-Ray é um AWS serviço que reúne dados sobre as solicitações que seu aplicativo atende e os usa para criar um mapa de serviços que você pode usar para identificar problemas com seu aplicativo e oportunidades de otimização.

> **Note**
>
> Algumas regiões não oferecem o X-Ray. Se você criar um ambiente em uma dessas regiões, não poderá executar o daemon do X-Ray nas instâncias no ambiente.
> Para obter informações sobre os AWS serviços oferecidos em cada região, consulte a tabela de regiões.



O X-Ray fornece um SDK que você pode usar para instrumentar o código da aplicação, e uma aplicação daemon que transmite as informações de depuração do SDK para a API do X-Ray.

Plataformas compatíveis

É possível usar o SDK do X-Ray com as seguintes plataformas do Elastic Beanstalk:

- Go - versão 2.9.1 e posterior
- Java 8 – versão 2.3.0 e posterior

- Java 8 com Tomcat 8 – versão 2.4.0 e posterior
- Node.js - versão 3.2.0 e posterior
- Windows Server – todas as versões de plataforma lançadas em 18 de dezembro de 2016 ou posteriormente
- Python – versão 2.5.0 e posterior

Nas plataformas compatíveis, é possível usar uma opção de configuração para executar o daemon do X-Ray nas instâncias do ambiente. Você pode habilitar o daemon no console do Elastic Beanstalk ou usando um arquivo de configuração.

Para fazer upload de dados para o X-Ray, o daemon X-Ray requer permissões do IAM na política AWSXrayWriteOnlyAccessgerenciada. Essas permissões são incluídas no perfil da instância do Elastic Beanstalk. Se você não usa o perfil de instância padrão, consulte Conceder Permissão do Daemon para Enviar Dados ao X-Ray no Guia do desenvolvedor do AWS X-Ray .

A depuração com o X-Ray requer o uso do SDK do X-Ray. Consulte os Conceitos básicos do AWS X-Ray no Guia do Desenvolvedor do AWS X-Ray para obter instruções e aplicações de amostra.

Se você usa uma versão de plataforma que não inclui o daemon, ainda é possível executá-lo com um script em um arquivo de configuração. Para obter mais informações, consulte Fazer Download e Executar o Daemon do X-Ray Manualmente (Avançado) no Guia do desenvolvedor do AWS X-Ray .

Seções

- Configurar depuração
- O namespace aws:elasticbeanstalk:xray

Configurar depuração

É possível habilitar daemon do X-Ray em um ambiente em execução no console do Elastic Beanstalk.

Como habilitar a depuração no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Amazon X-Ray, selecione Ativado.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Essa opção pode ser ativada também durante a criação do ambiente. Para obter mais informações, consulte O assistente de criação de novo ambiente.

O namespace aws:elasticbeanstalk:xray

Use a opção `XRayEnabled` no namespace `aws:elasticbeanstalk:xray` para habilitar a depuração.

Para habilitar a depuração automaticamente quando você implanta o aplicativo, defina a opção em um arquivo de configuração no código-fonte, como a seguir.

Example .ebextensions/debugging.config

```
option_settings:
  aws:elasticbeanstalk:xray:
    XRayEnabled: true
```

Visualizar seus logs de ambiente do Elastic Beanstalk

AWS Elastic Beanstalk fornece duas maneiras de visualizar regularmente os registros das EC2 instâncias da Amazon que executam seu aplicativo:

- Configure seu ambiente do Elastic Beanstalk para fazer upload de logs de instância alternados para o bucket do Amazon S3 do ambiente.
- Configure o ambiente para transmitir os registros da instância para o Amazon CloudWatch Logs.

Quando você configura o streaming de CloudWatch registros de instâncias para o Logs, o Elastic CloudWatch Beanstalk cria grupos de log de Logs para proxy e logs de implantação nas instâncias da EC2 Amazon e transfere esses CloudWatch arquivos de log para o Logs em tempo real. Para obter mais informações sobre logs de instância, visite Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

Além dos registros de instância, se você ativar a integridade aprimorada do seu ambiente, poderá configurar o ambiente para transmitir informações de integridade para o CloudWatch Logs. Quando o status de integridade do ambiente for alterado, o Elastic Beanstalk adicionará um registro a um grupo de logs de integridade com o novo status e uma descrição da causa da alteração. Para obter informações sobre o streaming de integridade do ambiente, visite Streaming de informações sobre a saúde do ambiente do Elastic Beanstalk para o Amazon Logs CloudWatch .

Configurar visualização de logs de instância

Para visualizar logs de instância, você pode habilitar a alternância de logs de instância e o streaming de logs no console do Elastic Beanstalk.

Como configurar a alternância e o streaming de logs de instância no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Armazenamento de logs do S3, selecione Ativado embaixo de Alternar logs para habilitar o upload de logs alternados para o Amazon S3.
6. na seção Transmissão de registros de instâncias para CloudWatch registros, defina as seguintes configurações:
   - Streaming de logs: selecione Ativado para habilitar o streaming de logs.
   - Retenção — especifique o número de dias para reter CloudWatch registros em Registros.
   - Ciclo de vida — Defina como Excluir registros após o encerramento para excluir os CloudWatch registros dos Registros imediatamente se o ambiente for encerrado, em vez de esperar que eles expirem.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Depois de ativar o streaming de logs, você poderá voltar para a categoria ou a página de configuração Software para encontrar o link Log Groups (Grupos de logs). Clique neste link para ver os registros da sua instância no CloudWatch console.

Configurar visualização do log de integridade do ambiente

Para visualizar logs de integridade do ambiente, é possível habilitar o streaming de logs de integridade do ambiente no console do Elastic Beanstalk.

Como configurar o streaming de logs de integridade do ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Vá para a seção Monitoramento.
6. Em Streaming de eventos de saúde para CloudWatch registros, defina as seguintes configurações:
   - Streaming de logs: escolha Ativado para habilitar o streaming de logs.
   - Retenção — especifique o número de dias para reter CloudWatch registros em Registros.
   - Ciclo de vida — Defina como Excluir registros após o encerramento para excluir os CloudWatch registros dos Registros imediatamente se o ambiente for encerrado, em vez de esperar que eles expirem.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Namespaces de visualização de logs

Os seguintes namespaces contêm configurações para visualização de logs:

- `aws:elasticbeanstalk:hostmanager`: configure o upload de logs alternados para o Amazon S3.
- `aws:elasticbeanstalk:cloudwatch:logs`— Configure o streaming do log da instância para CloudWatch.
- `aws:elasticbeanstalk:cloudwatch:logs:health`— Configure o streaming de integridade do ambiente para CloudWatch.

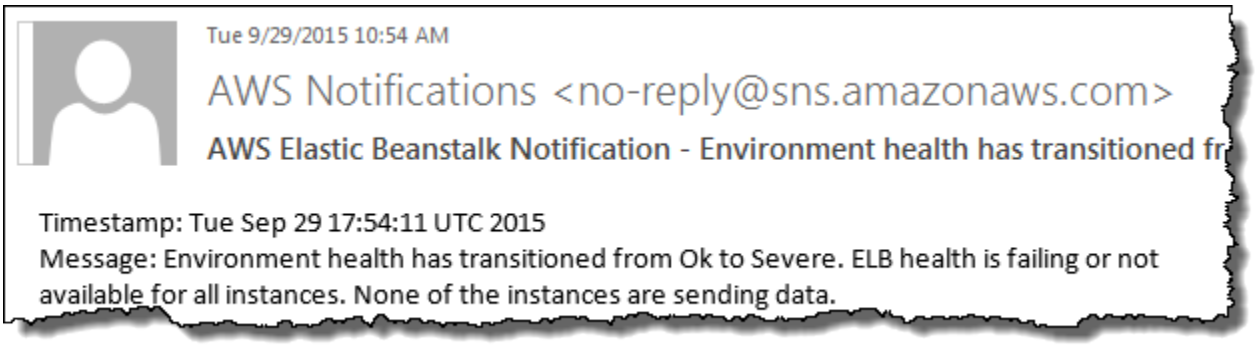
Notificações do ambiente do Elastic Beanstalk com o Amazon SNS

Você pode configurar seu AWS Elastic Beanstalk ambiente para usar o Amazon Simple Notification Service (Amazon SNS) para notificá-lo sobre eventos importantes que afetam seu aplicativo. Para receber e-mails AWS sempre que ocorrer um erro ou a integridade do seu ambiente mudar, especifique um endereço de e-mail ao criar um ambiente ou posteriormente.

> **Note**
>
> O Elastic Beanstalk usa o Amazon SNS para notificações. Para obter informações sobre os preços do Amazon SNS, consulte https://aws.amazon.com/sns/ preços/.

Quando você configura notificações para o seu ambiente, o Elastic Beanstalk cria um tópico do Amazon SNS para ele em seu nome. Para enviar mensagens para um tópico do Amazon SNS, o Elastic Beanstalk precisa ter a permissão necessária. Para obter mais informações, consulte Configurar permissões para enviar notificações.

Quando ocorre um evento notável, o Elastic Beanstalk envia uma mensagem para o tópico. Em seguida, o Amazon SNS retransmite as mensagens recebidas para os assinantes do tópico. Os eventos notáveis incluem erros de criação do ambiente e qualquer alteração na integridade do ambiente e da instância. Eventos para operações do Amazon EC2 Auto Scaling (como adicionar e remover instâncias do ambiente) e outros eventos informativos não acionam notificações.



Você pode inserir um endereço de e-mail no console do Elastic Beanstalk quando criar um ambiente ou em outro momento posterior. Isso criará um tópico do Amazon SNS e você estará inscrito nele. O Elastic Beanstalk gerencia o ciclo de vida do tópico e o exclui quando o ambiente é encerrado ou quando você remove seu endereço de e-mail no console de gerenciamento do ambiente.

O namespace do `aws:elasticbeanstalk:sns:topics` oferece opções para configurar um tópico do Amazon SNS usando arquivos de configuração, uma CLI ou um SDK. Usando um desses métodos, você pode configurar o tipo de assinante e o endpoint. Para o tipo de assinante, você pode escolher uma fila do Amazon SQS ou uma URL HTTP.

Só é possível ativar e desativar notificações do Amazon SNS. A frequência de notificações enviadas ao tópico pode ser alta, dependendo do tamanho e da composição do seu ambiente. Há outras opções para configurar notificações a serem enviadas em circunstâncias específicas. Você pode configurar regras orientadas por eventos com a Amazon EventBridge que notificam você quando o Elastic Beanstalk emite eventos que atendem a critérios específicos. Ou, como alternativa, você pode configurar seu ambiente para publicar métricas personalizadas e definir CloudWatch alarmes da Amazon para notificá-lo quando essas métricas atingirem um limite insalubre.

Configurar notificações usando o console do Elastic Beanstalk

Você pode inserir um endereço de e-mail no console do Elastic Beanstalk para criar um tópico do Amazon SNS para o seu ambiente.

Como configurar notificações no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até a seção Notificações por e-mail.
6. Insira um endereço de e-mail.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Quando você insere um endereço de e-mail para notificações, o Elastic Beanstalk cria um tópico do Amazon SNS para seu ambiente e adiciona uma inscrição. O Amazon SNS envia um e-mail para o endereço inscrito para confirmar a inscrição. Você deve clicar no link do e-mail de confirmação para ativar a inscrição e receber notificações.

Configurar notificações usando opções de configuração

Use as opções no namespace `aws:elasticbeanstalk:sns:topics` para configurar as notificações do Amazon SNS para o seu ambiente. Você pode definir estas opções usando arquivos de configuração, uma CLI ou um SDK.

- Endpoint da notificação: o endereço de e-mail, a fila do Amazon SQS ou a URL para enviar as notificações. Definir essa opção cria uma fila do SQS e uma inscrição para o endpoint especificado. Se o endpoint não for um endereço de e-mail, você deve definir também a opção `Notification Protocol`. O SNS valida o valor de `Notification Endpoint` de acordo com o valor de `Notification Protocol`. A configuração dessa opção várias vezes cria inscrições adicionais no tópico. Se você remover esta opção, o tópico será excluído.
- Protocolo de notificação: protocolo usado para enviar notificações ao `Notification Endpoint`. Essa opção usa como valor padrão `email`. Defina essa opção como `email-json` para enviar e-mails em formato JSON, `http` ou `https` para publicar notificações no formato JSON em um endpoint HTTP, ou `sqs` para enviar notificações para uma fila do SQS.

  > **Note**
  >
  > AWS Lambda as notificações não são suportadas.

- ARN do tópico de notificação ARN: depois de definir um endpoint de notificação para seu ambiente, leia esta configuração para obter o ARN do tópico do SNS. Você também pode definir esta opção para usar um tópico do SNS existente para notificações. Um tópico anexado ao seu ambiente por meio dessa opção não é excluído quando ela é alterada ou se o ambiente for encerrado.

  Para configurar as notificações do Amazon SNS, você precisa ter as permissões necessárias. Se seu usuário do IAM usa a política de usuário gerenciado do AdministratorAccessElastic AWSElastic Beanstalk - Beanstalk, então você já deve ter as permissões necessárias para configurar o tópico padrão do Amazon SNS que o Elastic Beanstalk cria para o seu ambiente. Contudo, se você configurar um tópico do Amazon SNS não gerenciado pelo Elastic Beanstalk, será necessário adicionar a política a seguir à sua função de usuário.

```
{
  "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 do tópico de identificação: defina esta opção para personalizar o nome do tópico do Amazon SNS usado para notificações de ambiente. Se já houver um tópico com o mesmo nome, o Elastic Beanstalk anexará o tópico ao ambiente.

  > **Warning**
  >
  > Se você anexar um tópico do SNS existente a um ambiente com `Notification Topic Name`, o Elastic Beanstalk excluirá o tópico quando você encerrar o ambiente ou alterar essa configuração futuramente.

  Se você alterar essa opção, o `Notification Topic ARN` também será alterado. Se um tópico já está conectado ao ambiente, o Elastic Beanstalk exclui o antigo e cria um novo tópico e assinatura.

  Ao usar um nome de tópico personalizado, você também deve fornecer um ARN de um tópico personalizado criado externamente. A política de usuário gerenciado não detecta automaticamente um tópico com um nome personalizado, portanto, você deve fornecer permissões personalizadas do Amazon SNS aos usuários do IAM. Use uma política semelhante à que foi utilizada para um ARN de tópico personalizado, mas faça as inclusões a seguir:

  - Inclua mais duas ações na lista de `Actions`, especificamente: `sns:CreateTopic`, `sns:DeleteTopic`
  - Se você estiver alterando o nome `Notification Topic Name` de um tópico personalizado para outro, você também deverá incluir os ARNs dois tópicos na `Resource` lista. Com

alternativa, inclua uma expressão regular que abranja ambos os tópicos. Dessa maneira, o Elastic Beanstalk tem permissões para excluir o tópico antigo e criar o novo.

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Configurar permissões para enviar notificações

Esta seção discute considerações de segurança relacionadas a notificações que utilizam o Amazon SNS. Há dois casos distintos:

- Use o tópico padrão do Amazon SNS criado pelo Elastic Beanstalk para o seu ambiente.
- Forneça um tópico externo do Amazon SNS por meio de opções de configuração.

A política de acesso padrão para tópicos do Amazon SNS permite que apenas o proprietário do tópico publique nele ou assine ele. Contudo, com a configuração de política adequada, o Elastic Beanstalk pode receber permissão para publicar em um tópico do Amazon SNS em um dos dois casos descritos nesta seção. As subseções a seguir fornecem mais informações.

Permissões para um tópico padrão

Quando você configura notificações para o ambiente, o Elastic Beanstalk cria um tópico do Amazon SNS para ele. Para enviar mensagens para um tópico do Amazon SNS, o Elastic Beanstalk precisa ter a permissão necessária. Se o seu ambiente usa a função de serviço que o console do Elastic Beanstalk ou a CLI do EB geraram para ele, ou a função vinculada ao serviço de monitoramento da sua conta, você não precisará fazer mais nada. Essas funções gerenciadas incluem a permissão necessária que permite que o Elastic Beanstalk envie mensagens para o tópico do Amazon SNS.

No entanto, se você tiver fornecido uma função de serviço personalizada ao criar o seu ambiente, certifique-se de que ela inclui a política a seguir.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:ElasticBeanstalkNotifications*"
      ]
    }
  ]
}
```

Permissões para um tópico externo

O Configurar notificações usando opções de configuração explica como substituir o tópico do Amazon SNS que o Elastic Beanstalk fornece por outro tópico do Amazon SNS. Se você substituiu o tópico, o Elastic Beanstalk deverá verificar se você tem permissão para publicar neste tópico do SNS para poder associar o tópico do SNS ao ambiente. Você deve ter `sns:Publish`. A função de serviço usa a mesma permissão. Para verificar se esse é o caso, o Elastic Beanstalk envia uma notificação de teste ao SNS como parte da sua ação para criar ou atualizar o ambiente. Se este teste falhar, sua tentativa de criar ou atualizar o ambiente também falhará. O Elastic Beanstalk exibe uma mensagem que explica o motivo dessa falha.

Se você fornecer uma função de serviço personalizada para o seu ambiente, certifique-se de que a sua função de serviço personalizada inclua a política a seguir para permitir que o Elastic Beanstalk envie mensagens para o tópico do Amazon SNS. No código a seguir, substitua *sns_topic_name* pelo nome do tópico do Amazon SNS fornecido nas opções de configuração.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:us-east-2:123456789012:sns_topic_name"
      ]
    }
  ]
}
```

Para obter mais informações sobre o controle de acesso do Amazon SNS, consulte Casos de exemplo para o controle de acesso do Amazon SNS no Guia do desenvolvedor do Amazon Simple Notification Service.

Configurar a Amazon Virtual Private Cloud (Amazon VPC) com o Elastic Beanstalk

A Amazon Virtual Private Cloud (Amazon VPC) é o serviço de rede que roteia o tráfego com segurança para as EC2 instâncias que executam seu aplicativo no Elastic Beanstalk. Se você não configurar uma VPC ao executar o ambiente, o Elastic Beanstalk usará a VPC padrão.

Você pode executar seu ambiente em uma VPC personalizada para customizar as configurações de rede e de segurança. Com o Elastic Beanstalk, é possível escolher quais sub-redes serão usadas nos seus recursos e como configurar endereços IP para as instâncias e o balanceador de carga no ambiente. Um ambiente é bloqueado para uma VPC quando você o cria. No entanto, é possível alterar as configurações de sub-rede e de endereço IP em um ambiente em execução.

Definir as configurações da VPC no console do Elastic Beanstalk

Se você escolheu uma VPC personalizada ao criar seu ambiente, poderá modificar as configurações de VPC no console do Elastic Beanstalk.

Para definir as configurações da VPC do seu ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Network (Rede), escolha Edit (Editar).

As seguintes configurações estão disponíveis.

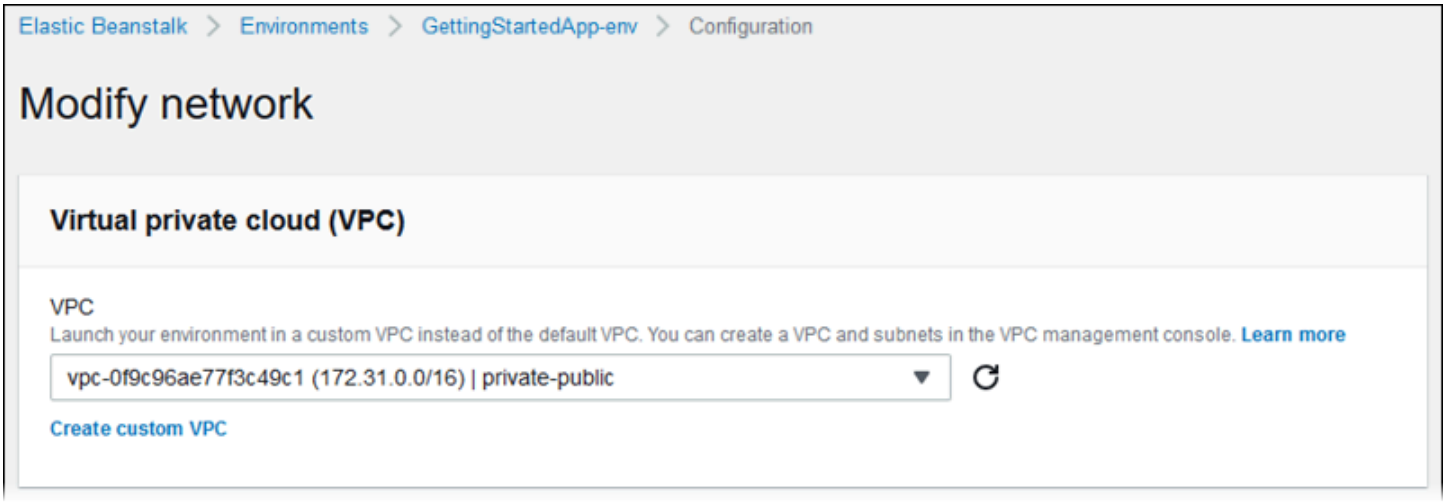
Opções

- VPC
- Visibilidade do load balancer
- Sub-redes do load balancer
- Endereço IP público da instância

- Sub-redes de instância
- Sub-redes do banco de dados

VPC

Escolha uma VPC em seu ambiente. Você só pode alterar essa configuração durante a criação do ambiente.

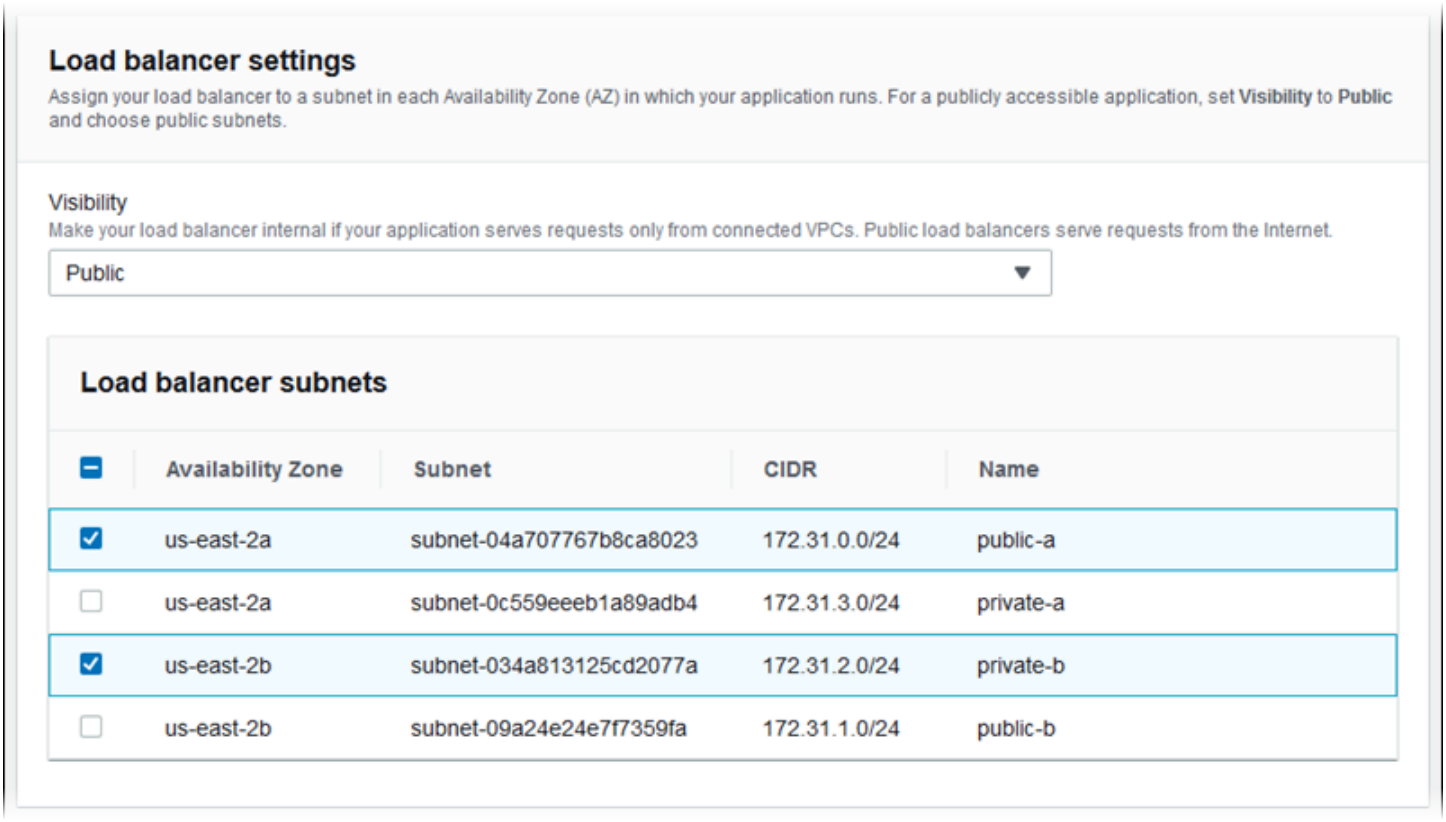


Visibilidade do load balancer

Para um ambiente com balanceamento de carga, escolha o esquema do load balancer. Por padrão, o load balancer é público e apresenta um endereço IP público e um nome de domínio. Se seu aplicativo somente veicular tráfego de dentro da sua VPC ou VPN conectada, desmarque essa opção e escolha as sub-redes privadas para que seu load balancer torná-lo interno e desative o acesso da Internet.

Sub-redes do load balancer

Para um ambiente com balanceamento de carga, escolha as sub-redes que o load balancer usa para veicular o tráfego. Para um aplicativo público, escolha sub-redes públicas. Use sub-redes em várias zonas de disponibilidade para alta disponibilidade. Para um aplicativo interno, escolha sub-redes privadas e desative a visibilidade do load balancer.

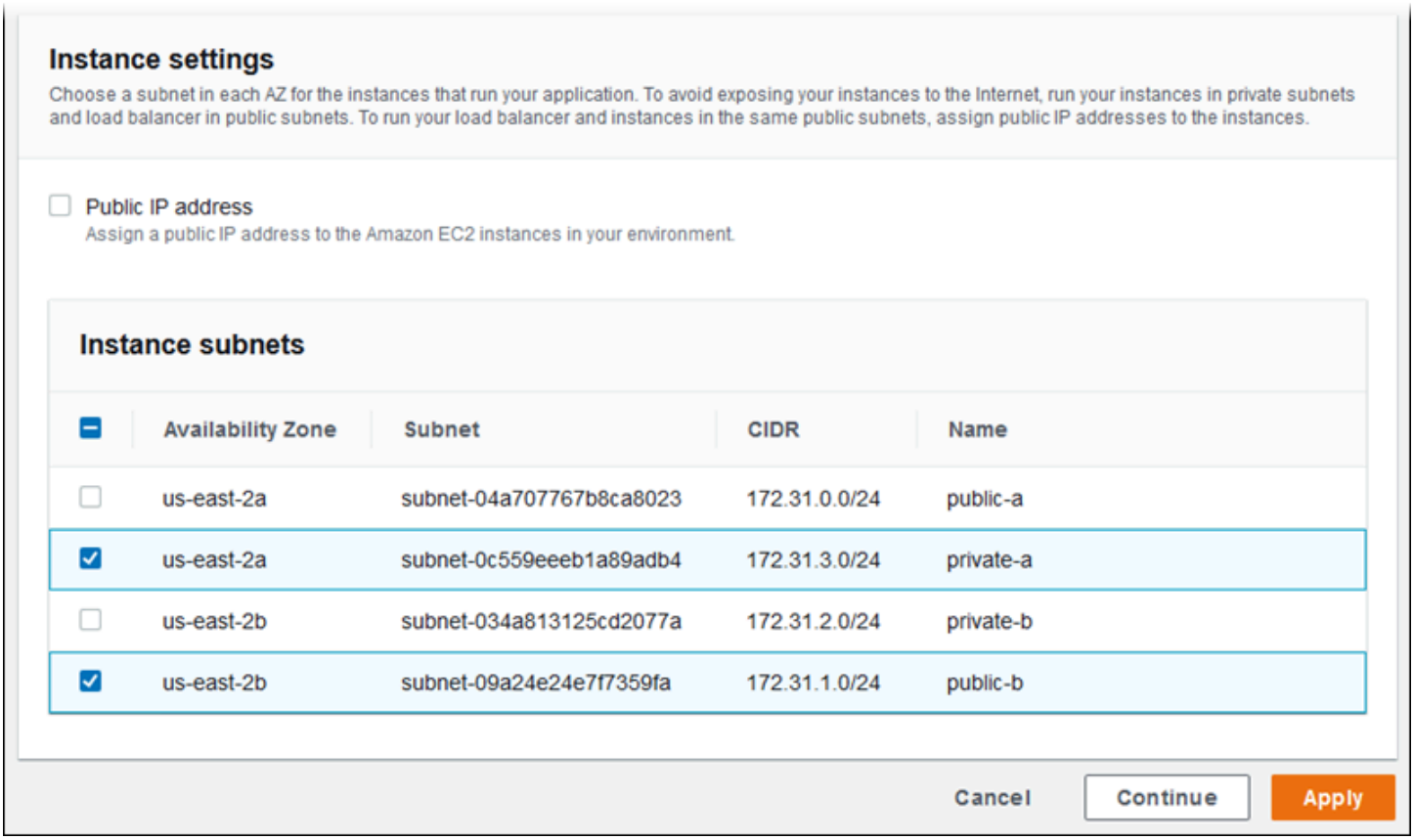


Endereço IP público da instância

Se você escolher sub-redes públicas para suas instâncias de aplicativos, ative endereços IP públicos para torná-los roteáveis a partir da Internet.

Sub-redes de instância

Escolha sub-redes para suas instâncias de aplicativo. Escolha pelo menos uma sub-rede para cada zona de disponibilidade que seu load balancer usa. Se você escolher sub-redes privadas para suas instâncias, sua VPC precisará ter um gateway NAT em uma sub-rede pública que as instâncias possam usar para acessar a Internet.



Sub-redes do banco de dados

Quando você executa um banco de dados do Amazon RDS anexado ao ambiente do Elastic Beanstalk, escolha as sub-redes para suas instâncias de banco de dados. Para alta disponibilidade, torne o banco de dados multi-AZ e escolha uma sub-rede para cada zona de disponibilidade. Para garantir que seu aplicativo possa se conectar ao banco de dados, execute ambos nas mesmas sub-redes.

O namespace aws:ec2:vpc

Você pode usar as opções de configuração no namespace `aws:ec2:vpc` para definir as configurações de rede do seu ambiente.

O seguinte arquivo de configuração usa opções neste namespace para definir a VPC e as sub-redes do ambiente em uma configuração público-privada. Para definir o ID da VPC em um arquivo de configuração, o arquivo precisará ser incluído no pacote de origem do aplicativo durante a criação do ambiente. Consulte Definir as opções de configuração durante a criação do ambiente para outros métodos de definição dessas configurações durante a criação do ambiente.

Example .ebextensions/vpc.config: público-privado

```
option_settings:
   aws:ec2:vpc:
      VPCId: vpc-087a68c03b9c50c84
      AssociatePublicIpAddress: 'false'
      ELBScheme: public
      ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
      Subnets: subnet-026c6117b178a9c45,subnet-0839e902f656e8bd1
```

Este exemplo mostra uma configuração pública-pública, em que o balanceador de carga e as EC2 instâncias são executados nas mesmas sub-redes públicas.

Example .ebextensions/vpc.config: público-privado

```
option_settings:
   aws:ec2:vpc:
      VPCId: vpc-087a68c03b9c50c84
      AssociatePublicIpAddress: 'true'
      ELBScheme: public
      ELBSubnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
      Subnets: subnet-0fe6b36bcb0ffc462,subnet-032fe3068297ac5b2
```

Migração de ambientes do Elastic EC2 Beanstalk do -Classic para um VPC

Este tópico descreve diferentes opções de como migrar seus ambientes do Elastic Beanstalk de uma plataforma de rede -Classic para uma EC2 rede Amazon Virtual Private Cloud (Amazon VPC).

Se você criou sua AWS conta antes de 4 de dezembro de 2013, talvez tenha ambientes que usam a configuração de rede EC2 -Classic em alguns Regiões da AWS. Todas as AWS contas criadas em ou após 4 de dezembro de 2013 já são exclusivas para VPC em todas as regiões. AWS As únicas isenções são se o Amazon EC2 -Classic foi ativado como resultado de uma solicitação de suporte.

> **Note**
>
> Você pode visualizar as definições da configuração de rede do ambiente na categoria Network configuration (Configuração de rede) na página Configuration overview (Visão geral da configuração) do console do Elastic Beanstalk.

Por que você deve migrar

O Amazon EC2 -Classic chegará ao fim do suporte padrão em 15 de agosto de 2022. Para evitar interrupções em suas cargas de trabalho, recomendamos que você migre do Amazon EC2 -Classic para uma VPC antes de 15 de agosto de 2022. Também solicitamos que você não lance nenhum AWS recurso no Amazon EC2 -Classic no futuro e use o Amazon VPC em vez disso.

Ao migrar seus ambientes do Elastic Beanstalk do EC2 Amazon -Classic para o Amazon VPC, você deve criar uma nova conta. AWS Você também deve recriar seus ambientes AWS EC2 -Classic em sua nova AWS conta. Não é necessário fazer nenhum trabalho adicional de configuração de seus ambientes para usar a VPC padrão. Se a VPC padrão não atender aos seus requisitos, você pode criar uma VPC personalizada manualmente e associá-la aos seus ambientes.

Como alternativa, se sua AWS conta existente tiver recursos que você não pode migrar para uma nova AWS conta, adicione uma VPC à sua conta atual. Depois, configure seus ambientes para usar a VPC.

Para obter mais informações, consulte a postagem do blog EC2-Classic Networking is Triing - Veja como preparar.

Migrar um ambiente do EC2 -Classic para uma nova AWS conta (recomendado)

Se você ainda não tiver uma AWS conta criada em ou após 4 de dezembro de 2013, crie uma nova conta. Você migrará seus ambientes para essa nova conta.

1. Sua nova AWS conta fornece uma VPC padrão para seus ambientes. Se você não precisar criar uma VPC personalizada, vá para a etapa 2.

   É possível criar uma VPC personalizada de uma das seguintes maneiras:

   - Crie uma VPC rapidamente usando o assistente do console da Amazon VPC, como uma das opções de configuração disponíveis. Para obter mais informações, consulte Configurações do assistente do console da Amazon VPC.
   - Crie uma VPC personalizada no console da Amazon VPC se você tem requisitos mais específicos para a sua VPC. Recomendamos que você faça isso, por exemplo, se seu caso de uso exigir um número específico de sub-redes. Para obter mais informações, consulte VPCs e sub-redes.
   - Crie uma VPC usando o elastic-beanstalk-samplesrepositório no GitHub site se você preferir usar AWS CloudFormation modelos com seus ambientes do Elastic Beanstalk. Esse

repositório inclui AWS CloudFormation modelos. Para obter mais informações, consulte Usar o Elastic Beanstalk com Amazon VPC.

> **Note**
>
> Você também pode criar uma VPC personalizada ao mesmo tempo em que recria o ambiente em sua nova AWS conta usando o assistente de criação de novo ambiente. Se você usar o assistente e optar por criar uma VPC personalizada, o assistente redireciona você para o console da Amazon VPC.

2. Em sua nova AWS conta, crie um novo ambiente. Recomendamos que o ambiente inclua a mesma configuração do ambiente existente na AWS conta da qual você está migrando. Isso pode ser feito usando uma das abordagens a seguir.

> **Note**
>
> Se seu novo ambiente precisar usar o mesmo CNAME após a migração, encerre o ambiente original na plataforma -Classic. EC2 Isso libera o CNAME para uso. No entanto, isso pode resultar em tempo de inatividade para esse ambiente e também pode arriscar que outro cliente selecione seu CNAME entre você encerrar seu ambiente EC2 - Classic e criar o novo. Para obter mais informações, consulte Encerrar um ambiente do Elastic Beanstalk.
>
> Para ambientes que têm seu próprio nome de domínio proprietário, o CNAME não tem esse problema. Você pode apenas atualizar o Domain Name System (DNS) para encaminhar solicitações ao novo CNAME.

- Use o assistente de criação de novo ambiente no console do Elastic Beanstalk. O assistente fornece uma opção para criar uma VPC personalizada. Se você não optar por criar uma VPC personalizada, uma VPC padrão é atribuída.
- Use a interface de linha de comando (EB CLI) do Elastic Beanstalk para recriar seu ambiente em sua nova conta. AWS Um dos exemplos na descrição do comando eb create demonstra a criação de um ambiente em uma VPC personalizada. Se você não fornecer o ID de uma VPC, o ambiente usa a VPC padrão.

  Usando essa abordagem, você pode usar um arquivo de configurações salvo nas duas AWS contas. Assim, você não precisa inserir manualmente todas as informações de

configuração. No entanto, você deve salvar as configurações do ambiente EC2 -Classic que você está migrando com o comando eb config save. Copie o arquivo de configuração salvo em um novo diretório para o ambiente da nova conta.

> **Note**
>
> Você deve editar alguns dos dados no arquivo de configuração salvo antes de usá-lo na nova conta. Você deve também atualizar as informações relativas à conta antiga com os dados corretos da nova conta. Por exemplo, deve substituir o nome do recurso da Amazon (ARN) da função do AWS Identity and Access Management (IAM) pelo ARN da função do AIM para a nova conta.

Se você usar o comando eb create com `cfg`, o novo ambiente é criado usando o arquivo de configuração salvo especificado. Para obter mais informações, consulte Usar configurações salvas do Elastic Beanstalk.

Migre um ambiente do EC2 -Classic na mesma conta AWS

Sua AWS conta existente pode ter recursos que você não pode migrar para uma nova AWS conta. Nesse caso, será necessário recriar os seus ambientes e configurar uma VPC manualmente para cada ambiente criado.

Migrar os ambientes para uma VPC personalizada

Pré-requisitos

Antes de começar, é necessário ter uma VPC. É possível criar uma VPC não padrão (personalizada) de uma das seguintes maneiras:

- Crie uma VPC rapidamente usando o assistente do console da Amazon VPC, com uma das opções de configuração disponíveis. Para obter mais informações, consulte Configurações do assistente do console da Amazon VPC.
- Crie uma VPC personalizada no console da Amazon VPC se você tem requisitos mais específicos para a sua VPC. Recomendamos que você faça isso, por exemplo, se seu caso de uso exigir um número específico de sub-redes. Para obter mais informações, consulte VPCs e sub-redes.
- Crie uma VPC usando o elastic-beanstalk-samplesrepositório no GitHub site se você preferir usar AWS CloudFormation modelos com seus ambientes do Elastic Beanstalk. Esse repositório inclui

AWS CloudFormation modelos. Para obter mais informações, consulte Usar o Elastic Beanstalk com Amazon VPC.

Nas etapas a seguir, você usa o ID e a sub-rede da VPC gerados IDs ao configurar a VPC no novo ambiente.

1. Crie um ambiente que inclua a mesma configuração do ambiente existente. Isso pode ser feito usando uma das abordagens a seguir.

   > **Note**
   >
   > O recurso Saved Configurations (Configurações salvas) pode ajudar a recriar os ambientes na nova conta. Esse recurso pode salvar a configuração de um ambiente, para que você possa aplicá-lo ao criar ou atualizar outros ambientes. Para obter mais informações, consulte Usar configurações salvas do Elastic Beanstalk.

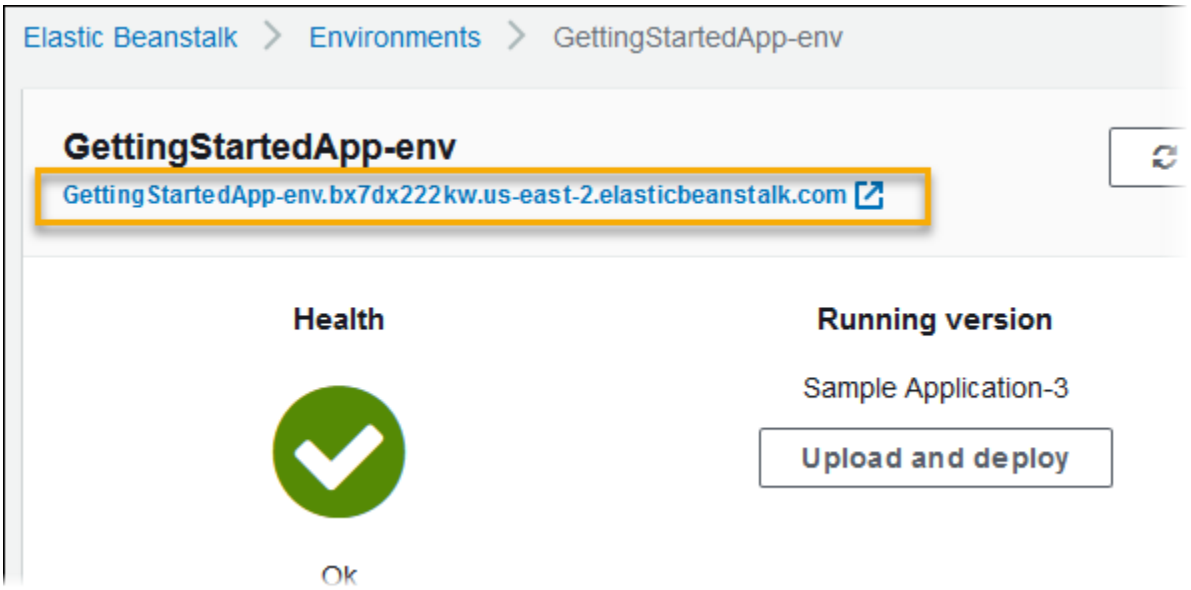
   - Usando o console do Elastic Beanstalk, aplique uma configuração salva EC2 do seu ambiente -Classic ao configurar o novo ambiente. Essa configuração usará a VPC. Para obter mais informações, consulte Usar configurações salvas do Elastic Beanstalk.
   - Com a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB), execute o comando eb create para recriar o ambiente. Forneça os parâmetros do ambiente original e do identificador da VPC. Um dos exemplos na descrição do comando eb create mostra como criar um ambiente em uma VPC personalizada.
   - Use o AWS Command Line Interface (AWS CLI) e recrie seu ambiente usando o elasticbeanstalk create-environment comando. Forneça os parâmetros do ambiente original com o identificador da VPC. Para obter instruções, consulte Criação de ambientes do Elastic Beanstalk com a CLI da AWS.
2. Troque CNAMEs o ambiente existente pelo novo ambiente. Dessa forma, o novo ambiente que você criou pode ser referenciado com o endereço já conhecido. Você pode usar a EB CLI ou a AWS CLI.
   - Usando a CLI do EB, troque o ambiente CNAMEs executando o comando. eb swap Para obter mais informações, consulte Configuração da interface de linha de comando do EB (EB CLI) para gerenciar o Elastic Beanstalk.

- Usando o AWS CLI, troque o ambiente CNAMEs com o comando swap-environment-cnameselasticbeanstalk. Para obter mais informações, consulte Referência de comandos da AWS CLI.

Nome de domínio do ambiente do Elastic Beanstalk

Por padrão, seu ambiente está disponível para os usuários em um subdomínio do `elasticbeanstalk.com`. Quando você cria um ambiente, pode escolher um nome de host para seu aplicativo. O subdomínio e o domínio são preenchidos automaticamente para *`region`*`.elasticbeanstalk.com`.

Para direcionar usuários para o ambiente do Elastic Beanstalk, registre um registro CNAME que aponte para o balanceador de carga do ambiente. É possível ver o URL da aplicação do ambiente com o valor atual do CNAME na página de visão geral do ambiente do console do Elastic Beanstalk.



Escolha o URL na página de visão geral ou escolha Go to environment (Acessar o ambiente) no painel de navegação para navegar até a página da web do aplicativo.

Você pode alterar o CNAME em seu ambiente substituindo-o pelo CNAME de outro ambiente. Para obter instruções, consulte Implantações azuis/verdes com o Elastic Beanstalk.

Se você tem um nome de domínio, pode usar o Amazon Route 53 para resolvê-lo para o ambiente. É possível comprar um nome de domínio com o Amazon Route 53 ou usar um adquirido de outro provedor.

Para comprar um nome de domínio com o Route 53, consulte Inscrever um novo domínio no Guia do desenvolvedor do Amazon Route 53.

Para saber mais sobre o uso de um domínio personalizado, consulte Encaminhamento de tráfego para um ambiente do AWS Elastic Beanstalk no Guia do desenvolvedor do Amazon Route 53.

> **Important**
>
> Se você encerrar um ambiente, também deverá excluir quaisquer mapeamentos CNAME criados, pois outros clientes poderão reutilizar um nome de host disponível. Certifique-se de excluir os registros DNS que apontam para seu ambiente encerrado para evitar uma entrada de DNS pendente. Uma entrada de DNS pendente pode expor o tráfego da Internet destinado ao seu domínio a vulnerabilidades de segurança. Também pode apresentar outros riscos.
>
> Para obter mais informações, consulte Proteção contra um registro de delegação pendente no Route 53 no Guia do desenvolvedor do Amazon Route 53. Você também pode aprender mais sobre entradas de DNS pendentes em Proteções de domínio aprimoradas para solicitações da CloudFront Amazon no AWS blog de segurança.

Configurar ambientes do Elastic Beanstalk (avançado)

Quando você cria um AWS Elastic Beanstalk ambiente, o Elastic Beanstalk provisiona e configura todos os recursos necessários para executar e dar AWS suporte ao seu aplicativo. Além de configurar os metadados e o comportamento de atualização do ambiente, você pode personalizar esses recursos fornecendo valores para opções de configuração. Por exemplo, você pode querer adicionar uma fila do Amazon SQS e um alarme sobre a profundidade da fila, ou você pode querer adicionar um cluster da Amazon. ElastiCache

A maioria das opções de configuração tem valores padrão que são aplicados automaticamente pelo Elastic Beanstalk. É possível substituir esses padrões com os arquivos de configuração, configurações salvas, opções da linha de comando ou chamando diretamente a API do Elastic Beanstalk. A CLI do EB e o console do Elastic Beanstalk também aplicam valores recomendados para algumas opções.

Você pode personalizar facilmente seu ambiente ao mesmo tempo que implanta a versão do aplicativo incluindo um arquivo de configuração com o pacote de origem. Ao personalizar o software em sua instância, é mais vantajoso usar um arquivo de configuração do que criar uma AMI personalizada, pois você não precisa manter um conjunto de. AMIs

Ao implantar seus aplicativos, você pode personalizar e configurar o software do qual o aplicativo depende. Esses arquivos podem ser dependências exigidas pela aplicação - por exemplo, pacotes adicionais do repositório yum - ou podem ser arquivos de configuração, como uma modificação para httpd.conf para anular configurações específicas que são usadas como padrão pelo AWS Elastic Beanstalk.

Tópicos

- Opções de configuração
- Personalização avançada de ambiente com arquivos de configuração (.ebextensions)
- Usar configurações salvas do Elastic Beanstalk
- Manifesto de ambiente (env.yaml)
- Usar uma imagem de máquina Amazon personalizada (AMI) no ambiente Elastic Beanstalk
- Fornecer arquivos estáticos
- Configurar HTTPS para o seu ambiente do Elastic Beanstalk

Opções de configuração

O Elastic Beanstalk define um grande número de opções de configuração que podem ser usadas para configurar o comportamento de seu ambiente e os recursos contidos neles. As opções de configuração são organizadas em namespaces, como `aws:autoscaling:asg`, que define opções para um grupo de Auto Scaling do ambiente.

O console do Elastic Beanstalk e a CLI do EB definem opções de configuração ao criar um ambiente, incluindo opções definidas explicitamente e valores recomendados definidos pelo cliente. Você também pode definir as opções de configuração em configurações salvas e em arquivos de configuração. Se a mesma opção for definida em várias localizações, o valor usado é determinado pela ordem de precedência.

As definições de opção de configuração podem ser elaboradas em formato de texto e salvas antes da criação do ambiente, aplicadas durante a criação do ambiente usando qualquer cliente compatível e adicionadas, modificadas ou removidas após a criação do ambiente. Para ver uma discriminação detalhada de todos os métodos disponíveis para trabalhar com as opções de configuração em cada um desses três estágios, leia os seguintes tópicos:

- Definir as opções de configuração antes da criação do ambiente
- Definir as opções de configuração durante a criação do ambiente
- Definir as opções de configuração depois da criação do ambiente

Para obter uma lista completa de namespaces e opções, incluindo os valores padrão e compatíveis de cada um deles, consulte Opções gerais para todos os ambientes e Opções específicas para a plataforma.

Precedência

Durante a criação do ambiente, as opções de configuração são aplicadas de várias fontes com a seguinte precedência, da mais alta para a mais baixa:

- Configurações aplicadas diretamente ao ambiente — Configurações especificadas durante uma operação de criação ou atualização de ambiente na API do Elastic Beanstalk por qualquer cliente, incluindo o console do Elastic Beanstalk, o EB CLI e. AWS CLI SDKs O console do Elastic Beanstalk e a CLI do EB também aplicam os valores recomendados de algumas opções aplicáveis nesse nível, a menos que sejam substituídos.

- Configurações salvas: as configurações para quaisquer opções que não são aplicadas diretamente ao ambiente são carregadas de uma configuração salva, se especificado.
- Arquivos de configuração (.ebextensions): as configurações para quaisquer opções que não são aplicadas diretamente ao ambiente, e também não são especificadas em uma configuração salva, são carregadas de arquivos de configuração na pasta `.ebextensions` na raiz do pacote de origem da aplicação.

  Os arquivos de configuração são executados em ordem alfabética. Por exemplo, `.ebextensions/01run.config` é executado antes de `.ebextensions/02do.config`.
- Valores padrão: se uma opção de configuração tem um valor padrão, ela é aplicada apenas quando a opção não é definida em nenhum dos níveis acima.

Se a mesma opção de configuração for definida em mais de um local, a configuração com a precedência mais alta será aplicada. Quando uma configuração é aplicada de uma configuração salva ou de configurações aplicadas diretamente ao ambiente, ela é armazenada como parte da configuração do ambiente. Essas configurações podem ser removidas com a AWS CLI ou com a EB CLI.

As configurações nos arquivos de configuração não são aplicadas diretamente ao ambiente e não podem ser removidas sem modificar os arquivos de configuração e implantar uma nova versão do aplicativo. Se uma configuração aplicada com um dos outros métodos for removida, a mesma configuração será carregada dos arquivos de configuração no pacote de origem.

Por exemplo, vamos supor que você defina o número mínimo de instâncias no ambiente como 5 durante a criação dele, usando o console do Elastic Beanstalk, uma opção de linha de comando ou uma configuração salva. O pacote de origem do seu aplicativo também inclui um arquivo de configuração que define o número mínimo de instâncias como 2.

Quando você cria o ambiente, o Elastic Beanstalk define a opção `MinSize` no namespace `aws:autoscaling:asg` como 5. Em seguida, se você remover a opção da configuração do ambiente, o valor no arquivo de configuração será carregado, e o número mínimo de instâncias será definido como 2. Na sequência, se você remover o arquivo de configuração do pacote de origem e reimplantá-lo, o Elastic Beanstalk usará a configuração padrão de 1.

Valores recomendados

A Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB) e o console do Elastic Beanstalk oferecem valores recomendados para algumas opções de configuração. Esses valores podem

ser diferentes dos valores padrão e são definidos no nível da API quando o ambiente é criado. Os valores recomendados permitem que o Elastic Beanstalk melhore a configuração do ambiente padrão sem fazer alterações na API incompatíveis com as versões anteriores.

Por exemplo, tanto o EB CLI quanto o console do Elastic Beanstalk definem a opção de configuração EC2 para o tipo de instância (no namespace). `InstanceType` `aws:autoscaling:launchconfiguration` Cada cliente dispõe de um método diferente para substituir a configuração padrão. No console, você pode escolher um tipo de instância diferente em um menu suspenso na página Configuration Details do assistente Create New Environment. Com a EB CLI, você pode usar o parâmetro `--instance_type` para eb create.

Como os valores recomendados são definidos no nível da API, eles substituirão os valores para as mesmas opções que você definir nos arquivos de configuração ou nas configurações salvas. As seguintes opções são definidas:

Console do Elastic Beanstalk

- Namespace: `aws:autoscaling:launchconfiguration`

  Nomes das opções: `IamInstanceProfile`, `EC2KeyName`, `InstanceType`
- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`

  Nomes das opções: `RollingUpdateType` e `RollingUpdateEnabled`
- Namespace: `aws:elasticbeanstalk:application`

  Nome da opção: `Application Healthcheck URL`
- Namespace: `aws:elasticbeanstalk:command`

  Nome da opção: `DeploymentPolicy`, `BatchSize` e `BatchSizeType`
- Namespace: `aws:elasticbeanstalk:environment`

  Nome da opção: `ServiceRole`
- Namespace: `aws:elasticbeanstalk:healthreporting:system`

  Nome da opção: `SystemType` e `HealthCheckSuccessThreshold`
- Namespace: `aws:elasticbeanstalk:sns:topics`

  Nome da opção: `Notification Endpoint`
- Namespace: `aws:elasticbeanstalk:sqsd`

Nome da opção: `HttpConnections`

- Namespace: `aws:elb:loadbalancer`

  Nome da opção: `CrossZone`

- Namespace: `aws:elb:policies`

  Nomes das opções: `ConnectionDrainingTimeout` e `ConnectionDrainingEnabled`

EB CLI

- Namespace: `aws:autoscaling:launchconfiguration`

  Nomes das opções: `IamInstanceProfile`, `InstanceType`

- Namespace: `aws:autoscaling:updatepolicy:rollingupdate`

  Nomes das opções: `RollingUpdateType` e `RollingUpdateEnabled`

- Namespace: `aws:elasticbeanstalk:command`

  Nome da opção: `BatchSize` e `BatchSizeType`

- Namespace: `aws:elasticbeanstalk:environment`

  Nome da opção: `ServiceRole`

- Namespace: `aws:elasticbeanstalk:healthreporting:system`

  Nome da opção: `SystemType`

- Namespace: `aws:elb:loadbalancer`

  Nome da opção: `CrossZone`

- Namespace: `aws:elb:policies`

  Nomes das opções: `ConnectionDrainingEnabled`

Definir as opções de configuração antes da criação do ambiente

AWS Elastic Beanstalk suporta um grande número de opções de configuração que permitem modificar as configurações aplicadas aos recursos em seu ambiente. Várias dessas opções têm

valores padrão que podem ser substituídos para personalizar o ambiente. Outras opções podem ser configuradas para ativar recursos adicionais.

O Elastic Beanstalk é compatível com dois métodos de salvamento das definições de opção de configuração. Os arquivos de configuração no formato JSON ou YAML podem ser incluídos no código-fonte do aplicativo em um diretório chamado `.ebextensions` e implantados como parte do seu pacote de origem do aplicativo. Crie e gerencie os arquivos de configuração localmente.

Configurações salvas são modelos que você cria em um ambiente em execução ou no arquivo de opções do JSON e armazena no Elastic Beanstalk. As configurações salvas existentes também podem ser estendidas para criar uma nova configuração.

> **Note**
>
> As configurações definidas nos arquivos de configuração e as configurações salvas têm menor precedência do que as configurações definidas durante ou após a criação do ambiente, inclusive os valores recomendados aplicados pelo console do Elastic Beanstalk e pela CLI do EB. Para mais detalhes, consulte Precedência.

As opções também podem ser especificadas em um documento JSON e fornecidas diretamente ao Elastic Beanstalk quando você criar ou atualizar um ambiente com a EB CLI ou a AWS CLI. As opções fornecidas diretamente ao Elastic Beanstalk dessa forma substituem todos os outros métodos.

Para uma lista completa de opções disponíveis, consulte Opções de configuração.

Métodos

- Arquivos de configuração (.ebextensions)
- Configurações salvas
- Documento JSON
- Configuração da EB CLI

Arquivos de configuração (`.ebextensions`)

Use `.ebextensions` para configurar as opções necessárias ao funcionamento do aplicativo e forneça valores padrão para outras opções que podem ser substituídas em um nível mais alto de

precedência. As opções especificadas em `.ebextensions` têm o nível mais baixo de precedência e são substituídas por configurações de qualquer outro nível.

Para usar arquivos de configuração, crie uma pasta chamada `.ebextensions` no nível superior do código-fonte do projeto. Adicione um arquivo com a extensão `.config` e especifique opções da seguinte maneira:

```
option_settings:
  - namespace:  namespace
    option_name:  option name
    value:  option value
  - namespace:  namespace
    option_name:  option name
    value:  option value
```

Por exemplo, o seguinte arquivo de configuração define o URL de verificação de integridade do aplicativo como `/health`:

`healthcheckurl.config`

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application
    option_name:  Application Healthcheck URL
    value:  /health
```

Em JSON:

```
{
 "option_settings" :
    [
      {
        "namespace" : "aws:elasticbeanstalk:application",
        "option_name" : "Application Healthcheck URL",
        "value" : "/health"
      }
    ]
}
```

Isso configura o load balancer do Elastic Load Balancing em seu ambiente do Elastic Beanstalk para fazer uma solicitação HTTP no `/health` caminho de EC2 cada instância para determinar se ela está íntegra ou não.

> **Note**
>
> YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Inclua o diretório `.ebextensions` em seu Pacote de origem da aplicação e implante-o no ambiente do Elastic Beanstalk novo ou já existente.

Os arquivos de configuração oferecem suporte a várias seções, além de `option_settings` para personalizar o software e arquivos que são executados nos servidores em seu ambiente. Para obter mais informações, consulte .Ebextensions.

Configurações salvas

Crie uma configuração salva para salvar as configurações aplicadas a um ambiente existente durante ou após a criação do ambiente, usando o console do Elastic Beanstalk, a EB CLI ou a AWS CLI. As configurações salvas pertencem a um aplicativo e podem ser aplicadas a ambientes novos ou existentes desse aplicativo.

Clientes

- Console do Elastic Beanstalk
- EB CLI
- AWS CLI

Console do Elastic Beanstalk

Como criar uma configuração salva (console do Elastic Beanstalk)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Save Configuration (Salvar configuração).
4. Use a caixa de diálogo na tela para concluir a ação.

As configurações salvas são armazenadas no bucket do S3 do Elastic Beanstalk em uma pasta com o mesmo nome da sua aplicação. Por exemplo, as configurações de uma aplicação chamada

`my-app` na região us-west-2 da conta número 123456789012 podem ser encontradas em `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app`.

EB CLI

A EB CLI também fornece subcomandos para interagir com configurações salvas em eb config:

Para criar uma configuração salva (EB CLI)

1. Salve a configuração atual associada ao ambiente:

   ```
   ~/project$ eb config save --cfg my-app-v1
   ```

   A EB CLI salva a configuração em `~/project/.elasticbeanstalk/saved_configs/my-app-v1.cfg.yml`
2. Modifique a configuração salva localmente, se necessário.
3. Faça upload da configuração salva no S3:

   ```
   ~/project$ eb config put my-app-v1
   ```

AWS CLI

Crie uma configuração salva de um ambiente em execução com `aws elasticbeanstalk create-configuration-template`

Para criar uma configuração salva (AWS CLI)

1. Identifique o ID do ambiente do Elastic Beanstalk com `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. Salve a configuração atual do ambiente com `create-configuration-template`:

```
$ aws elasticbeanstalk create-configuration-template --environment-id e-vcghmm2zwk
 --application-name my-app --template-name v1
```

O Elastic Beanstalk salva a configuração no bucket do Elastic Beanstalk no Amazon S3.

Documento JSON

Se você usar o AWS CLI para criar e atualizar ambientes, também poderá fornecer opções de configuração no formato JSON. Uma biblioteca de arquivos de configuração em JSON é útil se você usar o AWS CLI para criar e gerenciar ambientes.

Por exemplo, o seguinte documento JSON define o URL de verificação de integridade do aplicativo como `/health`:

~/ebconfigs/healthcheckurl.json

```
[
  {
    "Namespace": "aws:elasticbeanstalk:application",
    "OptionName": "Application Healthcheck URL",
```

```
    "Value": "/health"
  }
]
```

Configuração da EB CLI

Além de oferecer suporte a configurações salvas e à configuração direta do ambiente com eb config comandos, o EB CLI tem um arquivo de configuração com uma opção `default_ec2_keyname` chamada que você pode usar para especificar um EC2 par de chaves da Amazon para acesso SSH às instâncias em seu ambiente. A EB CLI usa essa opção para definir a opção de configuração `EC2KeyName` no 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
```

Definir as opções de configuração durante a criação do ambiente

Ao criar um AWS Elastic Beanstalk ambiente usando o console do Elastic Beanstalk, o EB AWS CLI CLI, um SDK ou a API do Elastic Beanstalk, você pode fornecer valores para as opções de configuração para personalizar seu ambiente e os recursos que são lançados nele. AWS

Para qualquer outro fim que não seja uma alteração de configuração isolada, é possível armazenar os arquivos de configuração localmente, em seu pacote de origem ou no Amazon S3.

Este tópico inclui procedimentos para todos os métodos de definição de opções de configuração durante a criação do ambiente.

Clientes

- No console do Elastic Beanstalk
- Uso da EB CLI
- Usando o AWS CLI

No console do Elastic Beanstalk

Ao criar um ambiente do Elastic Beanstalk no console do Elastic Beanstalk, é possível especificar as opções de configuração usando os arquivos de configuração, as configurações salvas e os formulários no assistente Create New Environment (Criar novo ambiente).

Métodos

- Usar arquivos de configuração (.ebextensions)
- Usar uma configuração salva
- Usar o assistente de novo ambiente

Usar arquivos de configuração (`.ebextensions`)

Inclua os arquivos `.config` em seu pacote de origem do aplicativo em uma pasta chamada `.ebextensions`.

Para obter detalhes sobre arquivos de configuração, consulte .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Faça upload do pacote de origem para o Elastic Beanstalk normalmente, durante a criação do ambiente.

O console do Elastic Beanstalk aplica valores recomendados a algumas opções de configuração e tem campos de formulário para outras. As opções configuradas pelo console do Elastic Beanstalk são aplicadas diretamente ao ambiente e substituem aquelas nos arquivos de configuração.

Usar uma configuração salva

Quando você cria um novo ambiente usando o console do Elastic Beanstalk, uma das primeiras etapas é escolher uma configuração. A configuração pode ser uma predefined configuration (configuração predefinida), normalmente a versão mais recente de uma Plataforma, como o PHP ou o Tomcat, ou pode ser uma saved configuration (configuração salva).

Como aplicar uma configuração salva durante a criação do ambiente (console do Elastic Beanstalk)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja aplicar e escolha Launch environment (Iniciar ambiente).
5. Continue no assistente para criar seu ambiente.

As configurações salvas são específicas ao aplicativo. Consulte Configurações salvas para obter detalhes sobre como criar configurações salvas.

Usar o assistente de novo ambiente

A maioria das opções de configuração padrão estão disponíveis na página Configure more options (Configurar mais opções) do Assistente de configuração de novo ambiente. Se você criar um banco de dados do Amazon RDS ou configurar uma VPC para o ambiente, as opções de configuração adicionais estarão disponíveis nas páginas desses recursos.

Como definir opções de configuração durante a criação do ambiente (console do Elastic Beanstalk)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, escolha Applications (Aplicativos).
3. Selecione ou crie um aplicativo.
4. Escolha Actions (Ações) e Create environment (Criar ambiente).
5. Continue no assistente e escolha Configure more options (Configurar mais opções).
6. Escolha qualquer uma das configuration presets (predefinições da configuração) e escolha Edit (Editar) em uma ou mais das categorias de configuração para alterar um grupo de opções de configuração relacionado.

7. Ao concluir as seleções das opções, escolha Create environment (Criar ambiente).

Todas as opções que você definir no assistente de novo ambiente serão definidas diretamente no ambiente e substituirão qualquer configuração de opção nas configurações salvas ou nos arquivos de configuração (`.ebextensions`) que você aplicar. Você poderá remover as configurações após a criação do ambiente usando a EB CLI ou a AWS CLI para permitir que as definições nas configurações salvas ou nos arquivos de configuração sejam exibidas.

Para obter detalhes sobre o novo assistente de ambiente, consulte O assistente de criação de novo ambiente.

Uso da EB CLI

Métodos

- Usar arquivos de configuração (.ebextensions)
- Usar as configurações salvas
- Usar as opções de linha de comando

Usar arquivos de configuração (**`.ebextensions`**)

Inclua os arquivos `.config` na pasta do seu projeto em `.ebextensions` para implantá-los com o código do seu aplicativo.

Para obter detalhes sobre arquivos de configuração, consulte .Ebextensions.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   `-- config.yml
|-- index.php
`-- styles.css
```

Crie seu ambiente e implante o código-fonte nele com eb create.

```
~/workspace/my-app$ eb create my-env
```

Usar as configurações salvas

Para aplicar uma configuração salva ao criar um ambiente com eb create, use a opção `--cfg`.

```
~/workspace/my-app$ eb create --cfg savedconfig
```

É possível armazenar a configuração salva na pasta do projeto ou no local de armazenamento do Elastic Beanstalk no Amazon S3. No exemplo anterior, a EB CLI primeiro procura um arquivo de configuração salvo chamado `savedconfig.cfg.yml` na pasta `.elasticbeanstalk/saved_configs/`. Não inclua as extensões de arquivo (`.cfg.yml`) ao aplicar uma configuração salva com `--cfg`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   |-- saved_configs
|   |   `-- savedconfig.cfg.yml
|   `-- config.yml
|-- index.php
`-- styles.css
```

Se a CLI do EB não encontrar a configuração localmente, ela verificará no local de armazenamento do Elastic Beanstalk no Amazon S3. Para obter detalhes sobre como criar, editar, e fazer upload das configurações salvas, consulte Configurações salvas.

Usar as opções de linha de comando

O comando eb create da EB CLI tem várias opções que você pode usar para definir as opções de configuração durante a criação do ambiente. Você pode usar essas opções para adicionar um banco de dados do RDS a seu ambiente, configurar uma VPC ou substituir os valores recomendados.

Por exemplo, a EB CLI usa o tipo de instância `t2.micro` por padrão. Para escolher um tipo de instância diferente, use a opção `--instance_type`.

```
$ eb create my-env --instance_type t2.medium
```

Para criar uma instância de banco de dados do Amazon RDS e anexá-la ao ambiente, use as opções `--database`.

```
$ eb create --database.engine postgres --database.username dbuser
```

Se você não inserir o nome do ambiente, a senha do banco de dados ou qualquer outro parâmetro necessário para criar o ambiente, a EB CLI solicitará que você os insira.

Consulte eb create para obter uma lista completa de opções disponíveis e exemplos de uso.

Usando o AWS CLI

Quando você usa o `create-environment` comando para criar um ambiente do Elastic Beanstalk AWS CLI com o AWS CLI , o não aplica nenhum valor recomendado. Todas as opções de configuração são definidas nos arquivos de configuração no pacote de origem especificado por você.

Métodos

- Usar arquivos de configuração (.ebextensions)
- Usar uma configuração salva
- Usar as opções de linha de comando

Usar arquivos de configuração (**`.ebextensions`**)

Para aplicar arquivos de configuração a um ambiente criado com o AWS CLI, inclua-os no pacote de origem do aplicativo que você carrega no Amazon S3.

Para obter detalhes sobre arquivos de configuração, consulte .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Para fazer o upload de um pacote de origem de aplicativos e criar um ambiente com o AWS CLI

1. Se você ainda não tiver um bucket do Elastic Beanstalk no Amazon S3, crie um com `create-storage-location`.

   ```
   $ aws elasticbeanstalk create-storage-location
   {
   ```

```
    "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
}
```

2. Faça upload do pacote de origem da aplicação para o Amazon S3.

```
$ aws s3 cp sourcebundle.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
sourcebundle.zip
```

3. Crie a versão do aplicativo.

```
$ 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. Criar o 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"
```

Usar uma configuração salva

Para aplicar uma configuração salva a um ambiente durante a criação, use o parâmetro `--template-name`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-name
 my-env --template-name savedconfig --version-label v1
```

Quando você especificar uma configuração salva, não especifique também o nome da pilha da solução. As configurações salvas já especificam uma pilha da soluções, e o Elastic Beanstalk retornará um erro se você tentar usar as duas opções.

Usar as opções de linha de comando

Use o parâmetro `--option-settings` para especificar as opções de configuração no 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"
  }
]
```

Para carregar o JSON de um arquivo, use o prefixo `file://`.

```
$ aws elasticbeanstalk create-environment --application-name my-app --environment-
name my-env --version-label v1 --template-name savedconfig --option-settings file://
healthcheckurl.json
```

O Elastic Beanstalk aplica as configurações de opção especificadas com a opção `--option-settings` diretamente ao ambiente. Se as mesmas opções forem especificadas em uma configuração salva ou arquivo de configuração, `--option-settings` substituirá esses valores.

Definir as opções de configuração depois da criação do ambiente

Você pode modificar as configurações de opções em um ambiente em execução, aplicando configurações salvas, carregando um novo pacote de origem com os arquivos de configuração (`.ebextensions`) ou usando um documento JSON. A CLI do EB e o console do Elastic Beanstalk também têm a funcionalidade específica do cliente para configurar e atualizar as opções de configuração.

Ao definir ou alterar uma opção de configuração, você pode acionar uma atualização completa do ambiente, dependendo da severidade da alteração. Por exemplo, alterações nas opções do `aws:autoscaling:launchconfiguration`, como `InstanceType`, exigem que as EC2 instâncias da Amazon em seu ambiente sejam reprovisionadas. Isso aciona uma atualização contínua. Outras alterações de configuração podem ser aplicadas sem qualquer interrupção ou novo provisionamento.

Você pode remover as configurações de opções de um ambiente com a CLI ou AWS CLI comandos do EB. Remover uma opção que foi definida diretamente em um ambiente em um nível de API permite que as configurações em arquivos de configuração, que são, caso contrário, mascaradas por configurações aplicadas diretamente a um ambiente, surjam e entrem em vigor.

As definições em configurações salvas e em arquivos de configuração podem ser substituídas definindo a mesma opção diretamente no ambiente com um dos outros métodos de configuração. No entanto, essas definições só podem ser removidas completamente aplicando uma atualização de uma configuração salva ou de um arquivo de configuração. Quando uma opção não for definida em

uma configuração salva, em um arquivo de configuração ou diretamente em um ambiente, o valor padrão será aplicado, se houver. Para mais detalhes, consulte Precedência.

Clientes

- O console do Elastic Beanstalk.
- A EB CLI
- O AWS CLI

O console do Elastic Beanstalk.

É possível atualizar as definições de opções de configuração no console do Elastic Beanstalk implantando um pacote de origem da aplicação que contenha os arquivos de configuração, aplicando uma configuração salva ou modificando o ambiente diretamente com a página Configuration (Configuração) no console de gerenciamento do ambiente.

Métodos

- Usar arquivos de configuração (.ebextensions)
- Usar uma configuração salva
- Usar o console do Elastic Beanstalk

Usar arquivos de configuração (`.ebextensions`)

Atualize os arquivos de configuração no diretório de origem, crie um novo pacote de origem e implante a nova versão no ambiente do Elastic Beanstalk para aplicar as alterações.

Para obter detalhes sobre arquivos de configuração, consulte .Ebextensions.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

As alterações feitas em arquivos de configuração não substituem as configurações de opção em configurações salvas ou em configurações aplicadas diretamente ao ambiente no nível de API. Consulte Precedência para obter mais detalhes.

Usar uma configuração salva

Aplique uma configuração salva a um ambiente em execução para aplicar as configurações de opção que ele define.

Como aplicar uma configuração salva a um ambiente em execução (console do Elastic Beanstalk)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja aplicar e escolha Load (Carregar).
5. Selecione um ambiente e escolha Load.

As configurações definidas em uma configuração salva substituem as configurações em arquivos de configuração e são substituídas pelas configurações definidas usando o console de gerenciamento do ambiente.

Consulte Configurações salvas para obter detalhes sobre como criar configurações salvas.

Usar o console do Elastic Beanstalk

O console do Elastic Beanstalk apresenta várias opções de configuração na página Configuration (Configuração) para cada ambiente.

Como alterar as opções de configuração em um ambiente em execução (console do Elastic Beanstalk)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Encontre a página de configuração a qual deseja editar:
   - Se você vir a opção que lhe interessa ou souber a categoria de configuração em que ela está, escolha Edit (Editar) na categoria de configuração dela.

- Para procurar uma opção, habilite a Table View (Exibição em tabela) e insira os termos de pesquisa na caixa de pesquisa. Conforme você digita, a lista diminui e mostra apenas as opções que correspondem aos seus termos de pesquisa.

  Quando vir a opção que você está procurando, escolha Edit (Editar) na categoria de configuração em que ela está.



5. Altere as configurações e selecione Save (Salvar).
6. Repita as duas etapas anteriores em outras categorias de configuração, conforme necessário.
7. Escolha Apply (Aplicar).

As alterações feitas nas opções de configuração no console de gerenciamento do ambiente são aplicadas diretamente ao ambiente. Essas alterações substituem as definições das mesmas opções em arquivos de configuração ou em configurações salvas. Para obter mais detalhes, consulte Precedence (Precedência).

Para obter detalhes sobre como alterar as opções de configuração em um ambiente em execução usando o console do Elastic Beanstalk, consulte os tópicos em Configurar ambientes do Elastic Beanstalk.

A EB CLI

Você pode atualizar as definições de opção de configuração com a EB CLI implantando um código-fonte que contenha os arquivos de configuração, aplicando as definições de uma configuração salva ou modificando a configuração do ambiente diretamente com o comando eb config.

Métodos

- Usar arquivos de configuração (.ebextensions)

- Usar uma configuração salva
- Usar o eb config
- Usar o eb setenv

Usar arquivos de configuração (**`.ebextensions`**)

Inclua os arquivos `.config` na pasta do seu projeto em `.ebextensions` para implantá-los com o código do seu aplicativo.

Para obter detalhes sobre arquivos de configuração, consulte .Ebextensions.

```
~/workspace/my-app/
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- .elasticbeanstalk
|   `-- config.yml
|-- index.php
`-- styles.css
```

Implante seu código-fonte com eb deploy.

```
~/workspace/my-app$ eb deploy
```

Usar uma configuração salva

Você pode usar o comando eb config para aplicar uma configuração salva a um ambiente em execução. Use a opção `--cfg` com o nome da configuração salva para aplicar suas configurações ao seu ambiente.

```
$ eb config --cfg v1
```

Neste exemplo, v1 é o nome de um criado anteriormente e o arquivo de configuração salvo.

As configurações aplicadas a um ambiente com esse comando substituem as configurações que foram aplicadas durante a criação do ambiente e as configurações definidas em arquivos de configuração em seu pacote de origem do aplicativo.

Usar o eb config

O comando eb config da EB CLI permite que você defina e remova configurações de opção diretamente em um ambiente usando um editor de texto.

Quando você executa eb config, a EB CLI exibe as configurações aplicadas em seu ambiente de todas as origens, inclusive de arquivos de configuração, configurações salvas, valores recomendados, opções definidas diretamente no ambiente e padrões de API.

> **Note**
>
> eb config não exibe as propriedades do ambiente. Para definir as propriedades do ambiente que você pode ler a partir do seu aplicativo, use eb setenv.

O exemplo a seguir mostra as configurações aplicadas no namespace `aws:autoscaling:launchconfiguration`. Essas configurações incluem:

- Dois valores recomendados, para `IamInstanceProfile` e `InstanceType`, aplicados pela EB CLI durante a criação do ambiente.
- A opção `EC2KeyName`definida diretamente no ambiente durante a criação com base na configuração do repositório.
- Os valores padrão da API para as outras opções.

```
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
...
```

Para definir ou alterar as opções de configuração com eb config

1. Execute eb config para visualizar a configuração de seu ambiente.

```
~/workspace/my-app/$ eb config
```

2. Altere qualquer um dos valores de configuração usando o editor de texto padrão.

```
aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: null
    EC2KeyName: my-key
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    ImageId: ami-1f316660
    InstanceType: t2.medium
```

3. Salve o arquivo de configuração temporário e saia.
4. A EB CLI atualiza a configuração do ambiente.

Definir as opções de configuração com eb config substitui as definições de todas as outras fontes.

Você também pode remover as opções de seu ambiente com eb config.

Para remover as opções de configuração (EB CLI)

1. Execute eb config para visualizar a configuração de seu ambiente.

```
~/workspace/my-app/$ eb config
```

2. Substitua qualquer valor mostrado com a string `null`. Você também pode excluir toda a linha que contém a opção que você deseja remover.

```
aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: null
    EC2KeyName: my-key
    IamInstanceProfile: aws-elasticbeanstalk-ec2-role
    ImageId: ami-1f316660
    InstanceType: null
```

3. Salve o arquivo de configuração temporário e saia.
4. A EB CLI atualiza a configuração do ambiente.

Remover as opções de seu ambiente com eb config permite que as configurações para as mesmas opções surjam a partir dos arquivos de configuração em seu pacote de origem do aplicativo. Consulte Precedência para obter mais detalhes.

Usar o eb setenv

Para definir as propriedades de ambiente com a EB CLI, use 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.
```

Esse comando define as propriedades de ambiente no `aws:elasticbeanstalk:application:environment` namespace. As propriedades do ambiente definidas com eb setenv ficam disponíveis para seu aplicativo após um curto processo de atualização.

Visualize as propriedades do ambiente definidas em seu ambiente com eb printenv.

```
~/workspace/my-app/$ eb printenv
 Environment Variables:
     ENVVAR = TEST
```

O AWS CLI

Você pode atualizar as configurações das opções de configuração com o AWS CLI implantando um pacote de origem que contém arquivos de configuração, aplicando uma configuração salva armazenada remotamente ou modificando o ambiente diretamente com o comando. `aws elasticbeanstalk update-environment`

Métodos

- Usar arquivos de configuração (.ebextensions)
- Usar uma configuração salva
- Usar as opções de linha de comando

Usar arquivos de configuração (**`.ebextensions`**)

Para aplicar arquivos de configuração a um ambiente em execução com o AWS CLI, inclua-os no pacote de origem do aplicativo que você carrega no Amazon S3.

Para obter detalhes sobre arquivos de configuração, consulte .Ebextensions.

```
~/workspace/my-app-v1.zip
|-- .ebextensions
|   |-- environmentvariables.config
|   `-- healthcheckurl.config
|-- index.php
`-- styles.css
```

Para fazer upload de um pacote de fonte da aplicação e aplicá-lo em um ambiente em execução (AWS CLI)

1. Se você ainda não tiver um bucket do Elastic Beanstalk no Amazon S3, crie um com `create-storage-location`:

   ```
   $ aws elasticbeanstalk create-storage-location
   {
       "S3Bucket": "elasticbeanstalk-us-west-2-123456789012"
   }
   ```

2. Faça upload do pacote de origem da aplicação para o Amazon S3.

   ```
   $ aws s3 cp sourcebundlev2.zip s3://elasticbeanstalk-us-west-2-123456789012/my-app/
   sourcebundlev2.zip
   ```

3. Crie a versão do aplicativo.

   ```
   $ 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. Atualize o ambiente.

   ```
   $ aws elasticbeanstalk update-environment --environment-name my-env --version-label
    v2
   ```

Usar uma configuração salva

Você pode aplicar uma configuração salva em um ambiente em execução com a opção `--template-name` no comando `aws elasticbeanstalk update-environment`.

A configuração salva deve estar no bucket do Elastic Beanstalk em um caminho com nome baseado na aplicação em `resources/templates`. Por exemplo, o modelo `v1` para a aplicação `my-app` na região Oeste dos EUA (Oregon) (us-west-2) para a conta 123456789012 está localizado em `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/v1`

Para aplicar uma configuração salva a um ambiente em execução (AWS CLI)

- Especifique a configuração salva em uma chamada `update-environment` com a opção `--template-name`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --template-name v1
```

O Elastic Beanstalk coloca as configurações salvas neste local quando são criadas com `aws elasticbeanstalk create-configuration-template`. Você também pode modificar as configurações salvas localmente e colocá-las nesse local por conta própria.

Usar as opções de linha de comando

Para alterar as opções de configuração com um documento JSON (AWS CLI)

1. Defina as configurações de opção no formato JSON em um arquivo local.
2. Execute `update-environment` com a opção `--option-settings`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --option-settings file://~/ebconfigs/as-zero.json
```

Neste exemplo, `as-zero.json` define as opções que configuram o ambiente com um mínimo e um máximo de instâncias zero. Isso interrompe as instâncias no ambiente sem encerrar o 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**
>
> Definir as opções de configuração com `update-environment` substitui as definições de todas as outras fontes.

Você também pode remover as opções de seu ambiente com `update-environment`.

Para remover as opções de configuração (AWS CLI)

- Execute o comando `update-environment` com a opção `--options-to-remove`.

```
$ aws elasticbeanstalk update-environment --environment-name my-env --options-to-
remove Namespace=aws:autoscaling:launchconfiguration,OptionName=InstanceType
```

Remover as opções de seu ambiente com `update-environment` permite que as configurações para as mesmas opções surjam a partir dos arquivos de configuração em seu pacote de origem do aplicativo. Se uma opção não for configurada usando qualquer um desses métodos, o valor padrão da API será aplicado, se houver. Consulte Precedência para obter mais detalhes.

Opções gerais para todos os ambientes

Namespaces

- 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: aplicação: segredos do meio ambiente
- 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

Configure o grupo de Auto Scaling do ambiente. Para obter mais informações, consulte the section called “Auto Scaling group (Grupo do Auto Scaling)”.

Namespace: **`aws:autoscaling:asg`**

Name (Nome)	Descrição	Padrão	Valores válidos

Availability Zones	As zonas de disponibilidade (AZs) são locais distintos dentro de uma AWS região que são projetados para serem isolados de falhas em outras AZs. Eles fornecem conectividade de rede barata e de baixa latência para outras pessoas AZs na mesma região. Escolha o número de AZs para suas instâncias.	`Any`	`Any`
			`Any 1`
			`Any 2`
			`Any 3`

Cooldown	Os períodos de espera ajudam a evitar que o Amazon EC2 Auto Scaling inicie atividades adicionais de escalabilidade antes que os efeitos das atividades anteriores sejam visíveis. O período de desaquecimento é a quantidad e de tempo, em segundos, após a conclusão de uma ação de escalabil idade antes que uma outra ação de escalabilidade possa ser iniciada.	`360`	`0` para `10000`

Name (Nome)	Descrição	Padrão	Valores válidos

Custom Availability Zones	Defina o AZs para suas instâncias.	Nenhum	`us-east-1a`
			`us-east-1b`
			`us-east-1c`
			`us-east-1d`
			`us-east-1e`
			`eu-central-1`

EnableCapacityRebalancing	Especifica se o recurso Rebalanceamento de capacidade deve ser habilitado para instâncias spot no seu grupo do Auto Scaling. Para obter mais informações, consulte Rebalanceamento de capacidade no Guia do usuário do Amazon Auto EC2 Scaling. Essa opção só é relevante quando `EnableSpot` é definido como `true` no namespace- `aws:ec2:instances` e quando há pelo menos uma instância spot no grupo do Auto Scaling.	`false`	`true` `false`
MinSize	O número mínimo de instâncias que você deseja no grupo de Auto Scaling.	`1`	1 para `10000`
MaxSize	O número máximo de instâncias desejadas no grupo de Auto Scaling.	`4`	1 para `10000`

aws:autoscaling:launchconfiguration

Configure as instâncias do Amazon Elastic Compute Cloud (Amazon EC2) para seu ambiente.

As instâncias usadas para seu ambiente são criadas usando um modelo de EC2 lançamento da Amazon ou um recurso de configuração de lançamento de grupo do Auto Scaling. As opções a seguir funcionam com ambos os tipos de recursos.

Para obter mais informações, consulte the section called “ EC2 Instâncias da Amazon”. Você também pode consultar mais informações sobre o Amazon Elastic Block Store (EBS) no capítulo Amazon EBS no Guia do usuário da Amazon. EC2

Namespace: **`aws:autoscaling:launchconfiguration`**

Name (Nome)	Descrição	Padrão	Valores válidos

DisableDefaultEC2SecurityGroup	Quando definido como o valor padrão `defalse`, o Elastic Beanstalk cria um grupo de segurança padrão que permite o tráfego da Internet ou do balanceador de carga nas portas padrão para HTTP (80). Ele anexa esse grupo de segurança às EC2 instâncias do ambiente ao criar o ambiente.
	
	Quando definido como `true` Elastic Beanstalk, não atribuirá o grupo de segurança padrão EC2 às instância s de um novo ambiente. Para um ambiente existente, o Elastic Beanstalk cancelará a atribuição do grupo de segurança padrão EC2 das instâncias do seu ambiente. EC2 Como resultado, você também deve definir as seguintes configurações:
		`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
	• A `SecurityGroups` opção desse namespace exigirá pelo menos um valor para definir seus grupos de segurança personalizados. • Para ambientes com um balancead or de carga, você também precisará definir as `SecurityGroups` opções em outro namespace para configurar grupos de segurança personalizados para o balancead or de carga. Para balanceadores de carga de aplicativos, defina a opção no aws:elbv2:loadbalancer namespace. Para balanceadores de carga clássicos, defina a opção no aws:elb:loadbalancer namespace. • Para obter mais informações, consulte Gerenciando grupos EC2 de segurança. Se um valor for especificado EC2KeyName em um ambiente `DisableDefaultEC2SecurityGr oup` definido como `true` um grupo de segurança padrão, não será associado às EC2 instâncias.		

Name (Nome)	Descrição	Padrão	Valores válidos

DisableIMDSv1	Defina como `true` para desativar o Serviço de Metadados de Instância Versão 1 (IMDSv1) e aplicar IMDSv2.
	
	Defina como `false` para ativar IMDSv1 tanto IMDSv2 e.
	
	As instâncias do seu ambiente são padronizadas da seguinte forma, com base no sistema operacional da plataforma:
	
	• Servidor Windows AL2 e versões anteriores — habilite ambos IMDSv1 e IMDSv2 (o DisableIMDSv1 padrão é) `false`
	• AL2023 — habilitar somente IMDSv2 (o DisableIMDSv1 padrão é) `true`
	
	Para obter mais informações, consulte Configurar o serviço de metadados de instância.
	
	Important
	Essa configuração de opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissõe
		`false`— plataformas baseadas em servidor Windows, Amazon Linux 2 e versões anteriores
		
		`true`— plataformas baseadas no Amazon Linux 2023
			`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
	s estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulte Migrando seu ambiente do Elastic Beanstalk para modelos de lançamento .		

Name (Nome)	Descrição	Padrão	Valores válidos
EC2KeyName	Você pode usar um key pair para fazer login com segurança na sua EC2 instância. Se um valor for especificado `EC2KeyName` em um ambiente `DisableDefaultEC2SecurityGroup` definido como `true` um grupo de segurança padrão, não será associado às EC2 instâncias. **Note** Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e substitui essa opção por um valor recomendado.	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos
IamInstanceProfile	Um perfil de instância permite que usuários e AWS serviços AWS Identity and Access Management (IAM) acessem credenciais de segurança temporárias para fazer chamadas de AWS API. Especifique o nome do perfil de instância ou o ARN dele. Exemplos: • `aws-elasticbeanstalk-ec2-role` • `arn:aws:iam::123456789012:instance-profile/aws-elasticbeanstalk-ec2-role` **Note** Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.	Nenhum	Nome do perfil de instância ou ARN.

Name (Nome)	Descrição	Padrão	Valores válidos
ImageId	Você pode substituir a Imagem de máquina da Amazon (AMI) padrão especificando o seu próprio ID de AMI personalizado. Example: `ami-1f316660`	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos
InstanceType	O tipo de instância usado para executar a aplicação em um ambiente do Elastic Beanstalk. **Important** A opção `InstanceType` está obsoleta. Ela foi substituída pela opção mais nova e mais eficiente `InstanceTypes` no namespace `aws:ec2:instances`. Você pode usar essa nova opção para especificar uma lista de um ou mais tipos de instância para seu ambiente. O primeiro valor nessa lista é equivalente ao valor da opção `InstanceType`, incluída no namespace `aws:autoscaling:launchconfiguration` que é descrito aqui. Recomendamos que você especifique os tipos de instância usando a nova opção. Se especificada, a nova opção tem precedência sobre a anterior. Para obter mais informações, consulte the section called “O namespace aws:ec2:instances”. Os tipos de instância disponíveis dependem das zonas de disponibi	Varia por conta e região.	Um tipo de EC2 instância. Varia de acordo com a conta, região e zona de disponibilidade. Você pode obter uma lista dos tipos de EC2 instância da Amazon filtrados por esses valores. Para obter mais informações, consulte Tipos de instância disponíveis no Guia EC2 do usuário da Amazon.

Name (Nome)	Descrição	Padrão	Valores válidos
	lidade e da região usadas. Se você escolher uma sub-rede, a zona de disponibilidade que contém essa sub-rede determinará os tipos de instância disponíveis. • O Elastic Beanstalk não é EC2 compatível com os tipos de instância do Amazon Mac. • Para obter mais informações sobre famílias e tipos de EC2 instâncias da Amazon, consulte Tipos de instância no Guia EC2 do usuário da Amazon. • Para obter mais informações sobre os tipos de instância disponíveis em todas as regiões, consulte Tipos de instância disponíveis no Guia EC2 do usuário da Amazon. **Note** Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.		

Name (Nome)	Descrição	Padrão	Valores válidos

LaunchTemplateTagPropagationEnabled	Defina como `true` para habilitar a propagação de tags de ambiente para os modelos de execução de recursos específicos provisionados para o ambiente.
	
	O Elastic Beanstalk só pode propagar tags para modelos de execução para os seguintes recursos:
	
	• Volumes do EBS
	• EC2 instâncias
	• EC2 interfaces de rede
	• AWS CloudFormation modelos de lançamento que definem um recurso
	
	Essa restrição existe porque CloudFormation só permite tags na criação de modelos para recursos específicos. Para obter mais informações, consulte TagSpecification no Guia AWS CloudFormation do usuário.
	
	⚠ Important
	
	• Alterar o valor dessa opção de `false` para `true` para um ambiente existente pode ser uma alteração significativa para tags já existentes.
	• Quando esse recurso estiver ativado, a propagação das tags precisará ser
		`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
	EC2 substituída, o que pode resultar em tempo de inatividade. É possível habilitar atualizações contínuas para aplicar alterações de configuração em lotes e evitar tempo de inatividade durante o processo de atualização. Para obter mais informações, consulte Alterações de configuração. Para obter mais informações sobre modelos de execução, consulte: • Modelos de lançamento no Guia do usuário do Amazon EC2 Auto Scaling • Trabalhar com modelos no Guia do usuário do AWS CloudFormation • Trechos de modelos do Elastic Beanstalk no Guia do usuário do AWS CloudFormation Para obter mais informações sobre essa opção, consulte Propagação de tags para modelos de execução.		

MonitoringInterval	O intervalo (em minutos) em que você deseja que CloudWatch as métricas da Amazon sejam retornadas.	`5 minute`	`1 minute`
			
			`5 minute`

Name (Nome)	Descrição	Padrão	Valores válidos
SecurityGroups	Lista o grupo EC2 de segurança da Amazon IDs a ser atribuído às EC2 instâncias no grupo Auto Scaling para definir regras de firewall para as instâncias. Use essa opção junto com `DisableDefaultEC2SecurityGroup` para anexar seus próprios grupos de segurança personalizados que definem as regras de firewall para as EC2 instâncias. Para obter mais informações, consulte Ambientes com balanceamento de carga (várias instâncias). **Important** Talvez seja necessário concluir algumas configurações adicionais para evitar que o tráfego de entrada em suas EC2 instâncias seja bloqueado. Isso se aplica somente a ambientes de várias instâncias com grupos EC2 de segurança personalizados. Os grupos EC2 de segurança devem incluir uma regra de entrada que conceda acesso ao tráfego roteado do balanceador de carga. Para obter mais informações, consulte Gerenciamento de	`elasticbeanstalk-default`	

Name (Nome)	Descrição	Padrão	Valores válidos
	grupos EC2 de segurança em ambientes de várias instâncias. Você pode fornecer uma única sequência de valores separados por vírgula que contenha grupos de EC2 segurança existentes da Amazon IDs ou referências a AWS::EC2::Security Group recursos criados no modelo. Você deve fornecer pelo menos um valor para essa opção se `DisableDefaultEC2SecurityGroup` o namespace estiver definido como. `true`		

Name (Nome)	Descrição	Padrão	Valores válidos
SSHSource Restriction	Usado para bloquear o acesso SSH a um ambiente. Por exemplo, você pode bloquear o acesso SSH às EC2 instâncias para que somente um bastion host possa acessar as instâncias na sub-rede privada. Essa string tem o seguinte formato: *protocol*, *fromPort*, *toPort*, *source_restriction* *protocol* O protocolo para a regra de entrada. *fromPort* O número da porta inicial. *toPort* O número da porta final. *source_restriction* O intervalo de Classless Inter-Domain Routing (CIDR) ou o grupo de segurança pelo qual o tráfego deve ser roteado. Especifique o grupo de segurança com o ID do grupo de segurança. Para especificar um grupo de segurança de outra conta, inclua o ID da conta da AWS antes do ID do grupo de segurança, separado	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos
	por uma barra. A outra conta deve estar na mesma AWS região. Observe a sintaxe: *aws-account-id* /*security-group-id*. Por exemplo: **123456789012 /sg-99999999** Exemplos: • `tcp, 22, 22, 54.240.196.185/32` • `tcp, 22, 22, my-security-group-id` • `tcp, 22, 22, 123456789012/their-security-group-id`		

Name (Nome)	Descrição	Padrão	Valores válidos
BlockDeviceMappings	Anexe os volumes adicionais do Amazon EBS ou volumes de armazenamento de instância em todas as instâncias no grupo de Auto Scaling. **Important** Essa configuração de opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulte Migrando seu ambiente do Elastic Beanstalk para modelos de lançamento . Ao mapear volumes de armazenamento de instâncias, é necessário	Nenhum	• tamanho: deve ser entre 500 e 16384 GiB • throughput: deve ser entre 125 a 1000 mebibytes por segundo (MiB/s)

Name (Nome)	Descrição	Padrão	Valores válidos
	mapear apenas o nome do dispositivo para um nome de volume. No entanto, recomendamos que, ao mapear volumes do Amazon EBS, você também especifique alguns ou todos os campos a seguir (cada campo deve ser separado por dois pontos): • ID do snapshot • tamanho, em GB • excluir no encerramento (`true` ou `false`) • tipo de armazenamento (somente para `gp3`, `gp2`, `standard`, `st1`, `sc1` ou `io1`) • IOPS (somente para `gp3` ou `io1`) • throughput (somente para `gp3`) O exemplo a seguir anexa três volumes do Amazon EBS, um volume vazio de 100 GB gp2 e um snapshot, um volume io1 vazio de 20 GB com 2.000 IOPS provisionadas, e um volume de armazenamento de instâncias `ephemeral0` . Vários volumes de armazenamento de instâncias podem ser anexados se o tipo de instância for compatível com ele. `/dev/sdj=:100:true:gp2,/dev/sdh=snap-51eef269,/dev/`		

Name (Nome)	Descrição	Padrão	Valores válidos
	`sdi=:20:true:io1:2000,/dev/sdb=ephemeral0`		
RootVolumeType	Tipo de volume (magnético, SSD de uso geral ou SSD IOPS provisionado) a ser usado para o volume raiz do Amazon EBS anexado às instâncias do seu ambiente. EC2 **Important** Essa configuração de opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulte Migrando seu ambiente do Elastic Beanstalk para modelos de lançamento .	Varia de acordo com a plataforma.	`standard` para armazenamento magnético. `gp2` ou `gp3` para SSD de uso geral. `io1` para SSD com IOPS provisionadas.

Name (Nome)	Descrição	Padrão	Valores válidos
RootVolumeSize	A capacidade de armazenamento do volume raiz do Amazon EBS em GB. Obrigatório se você definir `RootVolumeType` para SSD IOPS provisionadas. Por exemplo, ."64"	Varia por plataforma para armazenamento magnético e SSD de uso geral. Nenhum para SSD IOPS provisionadas.	`10` a `16384` GB para SSD de uso geral e SSD IOPS provisionadas. `8` a `1024` GB para magnético.
RootVolumeIOPS	As input/output operações por segundo (IOPS) desejadas para um volume raiz SSD de IOPS provisionado ou para um volume raiz SSD de uso geral. `gp3` A razão máxima entre IOPS e tamanho de volume é de 500 para 1. Por exemplo, um volume com 3000 IOPS deve ter pelo menos 6 GiB.	Nenhum	`100` para `20000` para volumes raízes SSD io1 com IOPS provisionadas. `3000` para `16000` para volumes raizes SSD `gp3` de uso geral.
RootVolumeThroughput	A taxa de transferência desejada de mebibytes por segundo (MiB/s) para provisionar o volume raiz do Amazon EBS conectado à instância do seu ambiente. EC2 **Note** Essa opção só é aplicável a armazenamento do tipo `gp3`.	Nenhum	`125` para `1000`

aws:autoscaling:scheduledaction

Configure ações programadas para o grupo de Auto Scaling do ambiente. Para cada ação, especifique um opção `resource_name` além do nome, namespace e valor da opção para cada configuração. Consulte O namespace aws:autoscaling:scheduledaction para ver exemplos.

Namespace: **`aws:autoscaling:scheduledaction`**

Name (Nome)	Descrição	Padrão	Valores válidos
StartTime	Para ações únicas, escolha a data e a hora para executar a ação. Para ações recorrentes, escolha quando ativar a ação.	Nenhum	Um time stamp da ISO-8601 exclusivo em todas as ações de escalabilidade agendadas.
EndTime	Uma data e hora no futuro (no fuso UTC/GMT horário) em que você deseja que a ação de escalabilidade programada pare de se repetir. Se você não especificar um EndTime, a ação será recorrente de acordo com a `Recurrence` expressão. Example: `2015-04-28T04:07:2Z` Quando uma ação programada termina, o Amazon EC2 Auto Scaling não volta automaticamente às configurações anteriores. Configure uma segunda ação programada para retornar às definições originais, conforme necessário.	Nenhum	Um time stamp da ISO-8601 exclusivo em todas as ações de escalabilidade agendadas.
MaxSize	A contagem máxima de instância s a ser aplicada quando a ação é executada.	Nenhum	`0` para `10000`

Name (Nome)	Descrição	Padrão	Valores válidos
MinSize	A contagem mínima de instância s a ser aplicada quando a ação é executada.	Nenhum	`0` para `10000`
DesiredCapacity	Defina a capacidade inicial desejada para o grupo de Auto Scaling. Depois que a ação programada for aplicada, os triggers ajustarão a capacidade desejada com base em suas configura ções.	Nenhum	`0` para `10000`
Recurrence	A frequência com que você deseja que a ação programada ocorra. Se você não especificar uma recorrênc ia, a ação de escalabilidade ocorrerá somente uma vez, conforme especific ado pelo `StartTime` .	Nenhum	Uma expressão Cron.
Suspend	Configure como `true` para desativar uma ação recorrente programada temporariamente.	`false`	`true` `false`

aws:autoscaling:trigger

Configure os triggers de escalabilidade para o grupo de Auto Scaling do ambiente.

> **Note**
>
> Três opções nesse namespace determinam por quanto tempo uma métrica de acionador pode permanecer além dos limites definidos antes que o acionador seja disparado. Essas opções são relacionadas da seguinte forma:
> `BreachDuration = Period * EvaluationPeriods`
> Os valores padrão para essas opções (5, 5 e 1, respectivamente) resolvem essa equação. Se você especificar valores inconsistentes, o Elastic Beanstalk poderá modificar um dos valores para que a equação ainda seja resolvida.

Namespace: **`aws:autoscaling:trigger`**

Name (Nome)	Descrição	Padrão	Valores válidos
BreachDuration	A quantidade de tempo, em minutos, que uma métrica pode estar além do limite definido (conforme especificado no `UpperThreshold` e `LowerThreshold`) antes que o gatilho seja invocado.	`5`	`1` para `600`
LowerBreachScaleIncrement	Quantas EC2 instâncias da Amazon devem ser removidas ao realizar uma atividade de escalabilidade.	`-1`	
LowerThreshold	Se a medida ficar abaixo desse número pela duração da violação, um gatilho é invocado.	`2000000`	`0` para `20000000`

MeasureName	A métrica usada para o acionador de Auto Scaling.
	
	Note
	`HealthyHostCount`, `UnhealthyHostCount` e `TargetResponseTime` são aplicáveis apenas a ambientes com um balanceador de carga dedicado. Estes não são valores de métrica válidos para ambientes configurados com um balanceador de carga compartilhado. Para obter mais informações sobre os tipos de balanceador de carga, consulte Balancead
		`NetworkOut`	`CPUUtilization`
			
			`NetworkIn`
			
			`NetworkOut`
			
			`DiskWriteOps`
			
			`DiskReadBytes`
			
			`DiskReadOps`
			
			`DiskWriteBytes`
			
			`Latency`
			
			`RequestCount`

Name (Nome)	Descrição	Padrão	Valores válidos

	or de carga do ambiente do Elastic Beanstalk.		`HealthyHostCount`
			`UnhealthyHostCount`
			`TargetResponseTime`

Period	Especifica com que frequência a Amazon CloudWatch mede as métricas do seu gatilho. O valor é o número de minutos entre dois períodos consecutivos.	`5`	1 para `600`
EvaluationPeriods	O número de períodos de avaliação consecutivos usados para determinar se está ocorrendo uma falha.	`1`	1 para `600`

Statistic	A estatística que o acionador deve usar, como `Average`.	`Average`	`Minimum`
			`Maximum`
			`Sum`
			`Average`

Name (Nome)	Descrição	Padrão	Valores válidos

Unit	Unidade de medição do gatilho, como `Bytes`.	`Bytes`	`Seconds`
			`Percent`
			`Bytes`
			`Bits`
			`Count`
			`Bytes/Second`
			`Bits/Second`
			`Count/Second`
			`None`

UpperBreachScaleIncrement	Especifica quantas EC2 instâncias da Amazon devem ser adicionadas ao realizar uma atividade de escalabilidade.	1	
UpperThreshold	Se a medida ficar acima desse número pela duração da violação, um gatilho será invocado.	`6000000`	`0` para `20000000`

aws:autoscaling:updatepolicy:rollingupdate

Configure atualizações contínuas para o grupo de Auto Scaling do ambiente.

Namespace: **`aws:autoscaling:updatepolicy:rollingupdate`**

Name (Nome)	Descrição	Padrão	Valores válidos
MaxBatchSize	O número de instâncias incluídas	Um terço do tamanho mínimo do grupo de Auto	1 para `10000`

Name (Nome)	Descrição	Padrão	Valores válidos
	em cada lote da atualização contínua.	Scaling arredonda do para o próximo valor inteiro.	
MinInstancesInService	O número mínimo de instâncias que devem estar em serviço dentro do grupo de Auto Scaling enquanto outras instâncias são encerradas.	O tamanho mínimo do grupo de Auto Scaling ou um inferior ao tamanho máximo do grupo de Auto Scaling, qualquer que seja o número menor.	`0` para 9999

Name (Nome)	Descrição	Padrão	Valores válidos

RollingUpdateEnabled	Se `true`, ele permite atualizações contínuas para um ambiente. As atualizações contínuas são úteis quando você precisa fazer pequenas atualizações frequentes na aplicação de software do Elastic Beanstalk e quer evitar tempo de inatividade da aplicação.
	
	A configuração desse valor para verdadeiro ativará automaticamente as opções `MaxBatchSize` , `MinInstancesInService` e `PauseTime` . A definição de qualquer uma dessas opções também define automaticamente a opção de valor `RollingUpdateEnabled` para `true`. A definição dessa opção para `false`
		`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
	desativa as atualizaç ões contínuas. **Note** Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.		

Name (Nome)	Descrição	Padrão	Valores válidos

RollingUpdateType	Isso inclui três tipos: atualizações contínuas baseadas em tempo, atualizações contínuas baseadas em saúde e atualizações imutáveis.
	
	As atualizações contínuas com base no tempo se aplicam PauseTime entre os lotes. Atualizações contínuas baseadas na integridade aguardam até que as novas instâncias passem nas verificações de integridade de antes de mudar para o próximo lote. Atualizações imutáveis iniciam um conjunto completo de instâncias em um novo grupo de Auto Scaling.
	
	Note
	Se você usar o console do Elastic Beanstalk ou
		`Time`	`Time`
			
			`Health`
			
			`Immutable`

Name (Nome)	Descrição	Padrão	Valores válidos
	a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.		
PauseTime	A quantidade de tempo (em segundos, minutos ou horas) que o serviço do Elastic Beanstalk aguarda após a conclusão das atualizações para um lote de instância s e antes que ele prossiga para o próximo lote.	Calculado automaticamente com base no tipo de instância e contêiner.	`PT0S`* (0 segundos) a `PT1H` (1 hora)

Name (Nome)	Descrição	Padrão	Valores válidos
Timeout	O tempo máximo de espera (em minutos ou horas) para que todas as instâncias em um lote de instâncias passem pelas verificações de integridade antes de cancelar a atualização.	`PT30M` (30 minutos)	`PT5M`* (5 minutos) a `PT1H` (1 hora) * Formato de duração ISO86 01: `PT#H#M#S` onde cada # é o número de horas, minutos e and/or segundos, respectivamente.

aws:ec2:instances

Configure as instâncias do seu ambiente, incluindo opções de spot. Este namespace complementa `aws:autoscaling:launchconfiguration` e `aws:autoscaling:asg`.

Para obter mais informações, consulte the section called “Auto Scaling group (Grupo do Auto Scaling)”.

Namespace: **`aws:ec2:instances`**

Name (Nome)	Descrição	Padrã	Valores válidos

EnableSpot	Habilite solicitações de instância spot para seu ambiente. Quando `false`, algumas opções neste namespace não entram em vigor.
	
	Important
	Essa configuração de opção pode fazer com que o Elastic Beanstalk migre um ambiente existente com configurações de execução para modelos de execução. Isso requer
		`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrã	Valores válidos
	as permissões necessárias para gerenciar os modelos de lançamento. Essas permissões estão incluídas em nossa política gerenciada. Se você usar políticas personalizadas em vez de nossas políticas gerenciadas, a criação ou as atualizações do ambiente poderão falhar ao atualizar a configuração do ambiente. Para obter mais informações e outras considerações, consulteMigrando seu ambiente do Elastic Beanstalk para modelos de lançamento .		

Name (Nome)	Descrição	Padrã	Valores válidos
InstanceTypes	Uma lista separada por vírgulas dos tipos de instância que você deseja que o ambiente use (por exemplo, `t2.micro,t3.micro`). Quando `EnableSpot` é `true` e `SpotAllocationStrategy` está definido como `capacity-optimized-prioritized` , a lista de valores especificada nessa opção determina a prioridade do tipo de instância para a estratégia de alocação de Instâncias Spot. Quando as instâncias Spot não estão ativadas (`EnableSpot` é `false`), apenas o primeiro tipo de instância na lista é usado. O primeiro tipo de instância na lista nesta opção é equivalente ao valor da opção `InstanceType` no namespace `aws:autoscaling:launchconfiguration` . Não recomendamos usar essa última opção porque ela está obsoleta. Se você especificar ambos, o primeiro tipo de instância na lista na opção `InstanceTypes` será usado e `InstanceType` será ignorado. Os tipos de instância disponíveis dependem das zonas de disponibilidade e da região usadas. Se você escolher uma sub-rede, a zona de disponibilidade que contém essa sub-	Uma lista de dois tipos de instânc . Varia por conta e região.	De um a quarenta tipos de EC2 instância . Recomendamos pelo menos dois Varia de acordo com a conta, região e zona de disponibilidade. Você pode obter uma lista dos tipos de EC2 instância da Amazon filtrados por esses valores. Para obter mais informaçõ es, consulte Tipos de instância disponíveis no Guia EC2 do usuário da Amazon. Os tipos de instância devem fazer parte da mesma arquitetura (`arm64`, `x86_64`, `i386`). `Supported Architectures` também faz parte desse namespace . Se você fornecer valores para `Supported Architectures` , o(s) valor(es) inserido(s) para `InstanceTypes` deve(m) pertencer a uma, e apenas uma, arquitetu

Name (Nome)	Descrição	Padrã	Valores válidos
	rede determinará os tipos de instância disponíveis. • O Elastic Beanstalk não é EC2 compatível com os tipos de instância do Amazon Mac. • Para obter mais informações sobre famílias e tipos de EC2 instância s da Amazon, consulte Tipos de instância no Guia EC2 do usuário da Amazon. • Para obter mais informações sobre os tipos de instância disponíveis em todas as regiões, consulte Tipos de instância disponíveis no Guia EC2 do usuário da Amazon. **Note** Algumas AWS contas mais antigas podem fornecer ao Elastic Beanstalk tipos de instância padrão que não são compatíveis com instâncias spot (por exemplo, t1.micro) . Se você ativar solicitações de instância Spot e receber um erro sobre um tipo de instância que não é compatíve l com Spot, configure tipos de instância que garantam essa compatibilidade. Para escolher os tipos de instância spot, use o Spot Instance Advisor.		ra que você fornece para `Supported Architectures` .

Name (Nome)	Descrição	Padrã	Valores válidos
	Quando você atualiza a configuração do seu ambiente e remove um ou mais tipos de instância da `InstanceTypes` opção, o Elastic Beanstalk encerra EC2 todas as instâncias da Amazon em execução em qualquer um dos tipos de instância removidos. O grupo d Auto Scaling do seu ambiente executa novas instâncias, conforme necessário, para completar a capacidade desejada, usando os tipos de instância especificados atuais.		

SpotAllocationStrategy	Especifica a estratégia de alocação de instâncias spot que determina como as instâncias spot são alocadas a partir dos pools de capacidade spot disponíveis.
	
	Se definido como`capacity-optimized-prioritized`, a ordem dos valores em `InstanceTypes` define a prioridade do tipo de instância para alocação.
	
	Esta opção é relevante somente quando `EnableSpot` é `true`.
		`capac`
		`optim`
			`capacity-optimized`
			
			`price-capacity-optimized`
			
			`capacity-optimized-prioritized`
			
			`lowest-price`

Name (Nome)	Descrição	Padrã	Valores válidos
SpotFleet OnDemandBase	O número mínimo de instâncias sob demanda que o grupo de Auto Scaling provisiona antes de considera r as instâncias spot à medida que o ambiente é dimensionado. Esta opção é relevante somente quando `EnableSpot` é `true`.	`0`	`0` para a opção `MaxSize` no namespace `aws:autos caling:asg`
SpotFleet OnDemandA boveBaseP ercentage	O percentual de instâncias sob demanda como parte da capacidad e adicional que o grupo de Auto Scaling provisiona além das instâncias `SpotOnDemandBase` . Esta opção é relevante somente quando `EnableSpot` é `true`.	`0` para um ambier de instânc única `70` para um ambier com balanc ento de carga	`0` para `100`

Name (Nome)	Descrição	Padrã	Valores válidos

SpotMaxPrice	O preço máximo por hora, em US$, que você está disposto a pagar por uma instância spot. Para recomendações sobre as opções de preço máximo para instâncias spot, consulte o histórico de preços de instâncias spot no Guia EC2 do usuário da Amazon.
	
	Esta opção é relevante somente quando `EnableSpot` é `true`.
		Preço sob demand para cada tipo de instânc. O valor da opção neste caso é `null`.	`0.001` para `20.0`
			
			`null`

Name (Nome)	Descrição	Padrã	Valores válidos

SupportedArchitectures	Uma lista separada por vírgulas dos tipos de arquitetura de EC2 instância que você usará em seu ambiente.
	
	O Elastic Beanstalk é compatível com tipos de instância baseados nas seguintes arquiteturas de processador:
	
	• AWS Arquitetura Arm Graviton de 64 bits (arm64)
	• Arquitetura de 64 bits (x86_64)
	• Arquitetura de 32 bits (i386)
	
	Para obter mais informações sobre a arquitetura do processador e os tipos de EC2 instância da Amazon, consultethe section called “Tipos de EC2 instância da Amazon”.
		Nenhu	`arm64`
			
			`x86_64`
			
			`i386`
			
			Note
			A arquitetura de 32 bits `i386` não é compatível com a maioria das plataformas do Elastic Beanstalk. Recomendamos que você escolha os tipos de arquitetura `x86_64` ou `arm64` em vez disso.

aws:ec2:vpc

Configure seu ambiente para iniciar recursos em uma Amazon Virtual Private Cloud (Amazon VPC) personalizada. Se você não definir configurações neste namespace, o Elastic Beanstalk executará recursos na VPC padrão.

Namespace: **`aws:ec2:vpc`**

Name (Nome)	Descrição	Padrão	Valores válidos
VPCId	O ID da sua Amazon VPC.	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos
Subnets	A IDs da sub-rede ou sub-redes do grupo Auto Scaling. Se você tiver várias sub-redes, especifique o valor como uma única sequência de caracteres de sub-rede separada por vírgula IDs (por exemplo,). `"subnet-11111111,subnet-22222222"`	Nenhum	
ELBSubnets	A IDs da sub-rede ou sub-redes do balanceador de carga elástico. Se você tiver várias sub-redes, especifique o valor como uma única sequência de caracteres de sub-rede separada por vírgula IDs (por exemplo,). `"subnet-11111111,subnet-222 22222"`	Nenhum	
ELBScheme	Especifique `internal` se você deseja criar um balanceador de carga interno na Amazon VPC para que a aplicação do Elastic Beanstalk não seja acessada de fora dela. Se você especificar um valor diferente de `public` ou `internal`, o Elastic Beanstalk ignorará o valor.	`public`	`public` `internal`
DBSubnets	Contém as sub-redes IDs do banco de dados. Opção usada somente se você deseja adicionar uma instância de banco de dados do Amazon RDS como parte do seu aplicativo. Se você tiver várias sub-redes , especifique o valor como uma única sequência de caracteres de sub-rede separada por vírgula IDs (por exemplo,). `"subnet-11111111,subnet-222 22222"`	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos

AssociatePublicIpAddress	Especifica se é para executar instâncias com endereços IP públicos na Amazon VPC. Instâncias com endereços IP públicos não exigem dispositivo NAT para comunicação com a Internet. Você deve definir o valor como `true` para incluir o balanceador de carga e as instâncias em uma única sub-rede pública.
	
	Essa opção não tem efeito em um ambiente de instância única, que sempre tem uma única EC2 instância da Amazon com um endereço IP elástico. A opção é relevante para ambientes escaláveis e com balanceamento de carga.
		Nenhum	`true`
			
			`false`

aws:elasticbeanstalk:application

Configure um caminho de verificação de integridade para seu aplicativo. Para obter mais informações, consulte Relatórios de integridade básica.

Namespace: **`aws:elasticbeanstalk:application`**

Name (Nome)	Descrição	Padrã	Valores válidos

URL de verificação de integridade do aplicativo	O caminho para o qual as solicitações de verificação de integridade são enviadas. Se esse caminho não estiver definido, o balanceador de carga tentará fazer uma conexão TCP na porta 80 para verificar o status de integridade do aplicativo. Configure para um caminho iniciando com / para enviar uma solicitação HTTP GET para esse caminho. Você também pode incluir um protocolo (HTTP, HTTPS, TCP ou SSL) e uma	Nenhu	Os valores válidos são:
			
			/ (HTTP GET para caminho raiz)
			
			`/health`
			
			`HTTPS:443/`
			
			`HTTPS:443/ health`

Name (Nome)	Descrição	Padrã	Valores válidos
	porta antes do caminho para verificar a conectividade HTTPS ou usar uma porta não padrão. **Note** Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e substitui essa opção por um valor recomendado.		

A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

aws:elasticbeanstalk:application:environment

Configure propriedades do ambiente para seu aplicativo.

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name (Nome)	Descrição	Padrão	Valores válidos
Qualquer nome de variável de ambiente.	Passe em pares de chave-valor.	Nenhum	Qualquer valor de variável de ambiente.

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

aws: elasticbeanstalk: aplicação: segredos do meio ambiente

Configure variáveis de ambiente para servir como segredos de ambiente para seu aplicativo. Os segredos do ambiente armazenam AWS Secrets Manager segredos ou AWS Systems Manager parâmetros do Parameter Store.

Namespace: **`aws:elasticbeanstalk:application:environmentsecrets`**

Name (Nome)	Descrição	Padrã	Valores válidos
OptionName	Especifica o nome da variável de ambiente para armazenar o valor do armazenamento secreto ou do armazenamento de parâmetros.	Nenhu	Qualquer nome de variável de ambiente.
Valor	Especifica o ARN para o valor armazenado em AWS Systems Manager ou AWS Secrets Manager no Parameter Store. Durante a inicializ ação da instância, o Elastic Beanstalk inicia a variável de ambiente com o valor armazenado nesse recurso ARN. **Note** Certifique-se de que as permissões necessárias estejam em vigor para que a função do perfil da EC2 instância do seu ambiente acesse o segredo e o parâmetro ARNs. Para obter mais informações, consulte Permissões obrigatórias do IAM.	Nenhu	Valor ARN válido para um valor AWS Secrets Manager secreto ou de AWS Systems Manager parâmetro do Parameter Store.

Para obter mais informações, consulte Configurando segredos como variáveis de ambiente.

aws:elasticbeanstalk:cloudwatch:logs

Configure o streaming de logs de instâncias para seu aplicativo.

Namespace: **`aws:elasticbeanstalk:cloudwatch:logs`**

Name (Nome)	Descrição	Padrão	Valores válidos
StreamLogs	Especifica se você deve criar grupos em CloudWatch Logs para registros de proxy e implantação e transmitir registros de cada instância em seu ambiente.	`false`	`true` `false`
DeleteOnTerminate	Especifica se os grupos de log devem ser excluídos quando o ambiente é encerrado. Se for `false`, os logs serão mantidos por `RetentionInDays` dias.	`false`	`true` `false`
RetentionInDays	O número de dias que os eventos de log serão mantidos antes que expirem.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:cloudwatch:logs:health

Configure o streaming de logs de integridade do ambiente para seu aplicativo.

Namespace: **`aws:elasticbeanstalk:cloudwatch:logs:health`**

Name (Nome)	Descrição	Padrão	Valores válidos
HealthStreamingEnabled	Para ambientes com relatórios de integridade aprimorados habilitados, especifica se é necessário criar um grupo no CloudWatch Logs para a integridade do ambiente e arquivar os dados de integridade do ambiente do Elastic Beanstalk. Para obter informações sobre como habilitar a integridade avançada, visite `aws:elasticbeanstalk:healthreporting:system`.	`false`	`true` `false`
DeleteOnTerminate	Especifica se o grupo de logs deve ser excluído quando o ambiente for encerrado. Se `false`, os dados de integridade são mantidos por `RetentionInDays` dias.	`false`	`true` `false`
RetentionInDays	O número de dias para manter os dados de integridade arquivados antes expirem.	7	1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 3653

aws:elasticbeanstalk:command

Configure a política de implantação para o código do aplicativo. Para obter mais informações, consulte the section called “Opções de implantação”.

Namespace: **`aws:elasticbeanstalk:command`**

Name (Nome)	Descrição	Padrão	Valores válidos

DeploymentPolicy	Escolha uma política de implantação para implantações de versões do aplicativo.
	
	Note
	Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e substitui essa opção por um valor recomendado.
		`AllAtOnce`	`AllAtOnce`
			
			`Rolling`
			
			`RollingWithAdditionalBatch`
			
			`Immutable`
			
			`TrafficSplitting`

Timeout	A quantidade de tempo, em segundos, para aguardar uma instância para concluir a execução de comandos. O Elastic Beanstalk adiciona internamente 240 segundos (quatro minutos) ao valor `Timeout`. Por exemplo, o tempo limite efetivo por padrão é 840 segundos (600 + 240) ou 14 minutos.	`600`	1 para `3600`

BatchSizeType	O tipo de número especificado em BatchSize.
	
	Note
	Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB
		`Percentage`	`Percentage`
			
			`Fixed`

Name (Nome)	Descrição	Padrão	Valores válidos
	substituem essa opção por um valor recomendado.		
BatchSize	A porcentagem ou o número fixo de EC2 instâncias da Amazon no grupo Auto Scaling nas quais realizar implantações simultane amente. Os valores válidos variam de acordo com a BatchSizeTypeconfiguração usada. **Note** Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.	`100`	1 para `100` (`Percentage`). 1para aws:autoscaling:asg::() MaxSize `Fixed`

IgnoreHealthCheck	Não cancele uma implantação devido a reprovação nas verificações de integridade.	`false`	`true`
			
			`false`

aws:elasticbeanstalk:environment

Configure a arquitetura e a função de serviço de seu ambiente.

Namespace: **`aws:elasticbeanstalk:environment`**

Name (Nome)	Descrição	Padrão	Valores válidos

EnvironmentType	Defina como `SingleInstance` para iniciar uma EC2 instância sem balanceador de carga.	`LoadBalanced`	`SingleInstance`
			
			`LoadBalanced`

ServiceRole	O nome de uma função do IAM que o Elastic Beanstalk usa para gerenciar recursos do ambiente. Especifique um nome de função (opcionalmente acompanhados de prefixo com um caminho personalizado) ou o ARN dele. Exemplos: • `aws-elasticbeanstalk-service-role` • *`custom-path`* `/`*`custom-role`* • `arn:aws:iam::123456789012:role/aws-elasticbeanstalk-service-role` **Note** Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.	Nenhum	Nome da função do IAM, caminho/nome ou ARN

Name (Nome)	Descrição	Padrão	Valores válidos

LoadBalancerType	O tipo de balanceador de carga para o seu ambiente. Para obter mais informações, consulte the section called “Load balancer”.	`classic`	`classic`
			`application`
			`network`

LoadBalancerIsShared	Especifica se o balanceador de carga do ambiente é dedicado ou compartilhado. Essa opção só pode ser definida para um Application Load Balancer. Ela não pode ser alterada após a criação do ambiente. Se `false`, o ambiente terá seu próprio balanceador de carga dedicado, criado e gerenciado pelo Elastic Beanstalk. Se `true`, o ambiente usará um balanceador de carga compartilhado, criado por você e especificado na opção `SharedLoadBalancer` do namespace aws:elbv2:loadbalancer.	`false`	`true` `false`

aws:elasticbeanstalk:environment:process:default

Configure o processo padrão de seu ambiente.

Namespace: **`aws:elasticbeanstalk:environment:process:default`**

Name (Nome)	Descrição	Padrão	Valores válidos
DeregistrationDelay	A quantidade de tempo, em segundos, para aguardar a conclusão das solicitações ativas	`20`	`0` para `3600`

Name (Nome)	Descrição	Padrão	Valores válidos
	antes de cancelar o registro.		
HealthCheckInterval	O intervalo de tempo, em segundos, em que o Elastic Load Balancing verifica a integridade das EC2 instâncias da Amazon do seu aplicativo.	Com Classic ou Application Load Balancer: `15` Com Network Load Balancer: `30`	Com Classic ou Application Load Balancer: `5` a `300` Com Network Load Balancer: `10`, `30`
HealthCheckPath	O caminho para o qual as solicitações HTTP para verificaç ões de integridade são enviadas.	`/`	Um caminho roteável.
HealthCheckTimeout	A quantidade de tempo, em segundos, para aguardar uma resposta durante uma verificação de integridade. Essa opção só é aplicável a ambientes com um Application Load Balancer.	`5`	1 para `60`
HealthyThresholdCo unt	O número de solicitaç ões consecutivas bem-sucedidas antes de o Elastic Load Balancing alterar o status de integridade da instância.	Com Classic ou Application Load Balancer: `3` Com Network Load Balancer: `5`	2 para `10`

Name (Nome)	Descrição	Padrão	Valores válidos
MatcherHTTPCode	Uma lista separada por vírgulas de códigos HTTP que indicam que uma instância está íntegra. Essa opção só é aplicável a ambientes com um Network ou Application Load Balancer.	`200`	Com Application Load Balancer: `200` a `499` Com Network Load Balancer: `200` a `399`
Port	Porta que o processo escuta.	`80`	1 para `65535`

Protocol	O protocolo que o processo usa.
	
	Com Application Load Balancer, você só pode definir essa opção como `HTTP` ou `HTTPS`.
	
	Com Network Load Balancer, você só pode definir essa opção como `TCP`.
		Com Classic ou Application Load Balancer: `HTTP`
		
		Com Network Load Balancer: `TCP`
			`TCP`
			
			`HTTP`
			
			`HTTPS`

Name (Nome)	Descrição	Padrão	Valores válidos

StickinessEnabled	Defina como verdadeiro para ativar sticky sessions.
	
	Essa opção só é aplicável a ambientes com um Application Load Balancer.
		`'false'`	`'false'`
			
			`'true'`

StickinessLBCookie Duration	A duração, em segundos, do cookie de sessão do sticky. Essa opção só é aplicável a ambientes com um Application Load Balancer.	`86400` (um dia)	1 para `604800`
StickinessType	Defina como `lb_cookie` para usar cookies nas sticky sessions. Essa opção só é aplicável a ambientes com um Application Load Balancer.	`lb_cookie`	`lb_cookie`
UnhealthyThreshold Count	Solicitações consecutivas sem êxito antes de o Elastic Load Balancing alterar o status de integridade da instância.	`5`	2 para `10`

aws:elasticbeanstalk:environment:process:process_name

Configure processos adicionais para o seu ambiente.

Namespace: **`aws:elasticbeanstalk:environment:process:`*`process_name`***

Name (Nome)	Descrição	Padrão	Valores válidos
DeregistrationDelay	A quantidade de tempo, em segundos, para aguardar a conclusão das solicitações ativas antes de cancelar o registro.	`20`	`0` para `3600`
HealthCheckInterval	O intervalo, em segundos, em que o Elastic Load Balancing verifica a integridade das EC2 instâncias da Amazon para seu aplicativo.	Com Classic ou Application Load Balancer: `15` Com Network Load Balancer: `30`	Com Classic ou Application Load Balancer: `5` a `300` Com Network Load Balancer: `10`, `30`
HealthCheckPath	O caminho para o qual as solicitações HTTP para verificaç ões de integridade são enviadas.	`/`	Um caminho roteável.
HealthCheckTimeout	A quantidade de tempo, em segundos, para aguardar uma resposta durante uma verificação de integridade. Essa opção só é aplicável a ambientes	`5`	`1` para `60`

Name (Nome)	Descrição	Padrão	Valores válidos
	com um Application Load Balancer.		
HealthyThresholdCount	O número de solicitações consecutivas bem-sucedidas antes de o Elastic Load Balancing alterar o status de integridade da instância.	Com Classic ou Application Load Balancer: `3` Com Network Load Balancer: `5`	2 para `10`
MatcherHTTPCode	Uma lista separada por vírgulas de códigos HTTP que indicam que uma instância está íntegra. Essa opção só é aplicável a ambientes com um Network ou Application Load Balancer.	`200`	Com Application Load Balancer: `200` a `499` Com Network Load Balancer: `200` a `399`
Port	A porta que o processo escuta.	`80`	1 para `65535`

Name (Nome)	Descrição	Padrão	Valores válidos

Protocol	O protocolo que o processo usa.
	
	Com Application Load Balancer, você só pode definir essa opção como `HTTP` ou `HTTPS`.
	
	Com Network Load Balancer, você só pode definir essa opção como `TCP`.
		Com Classic ou Application Load Balancer: `HTTP`
		
		Com Network Load Balancer: `TCP`
			`TCP`
			
			`HTTP`
			
			`HTTPS`

StickinessEnabled	Defina como verdadeiro para ativar sticky sessions.
	
	Essa opção só é aplicável a ambientes com um Application Load Balancer.
		`'false'`	`'false'`
			
			`'true'`

StickinessLBCookie Duration	A duração, em segundos, do cookie de sessão do sticky. Essa opção só é aplicável a ambientes com um Application Load Balancer.	`86400` (um dia)	1 para `604800`

Name (Nome)	Descrição	Padrão	Valores válidos
StickinessType	Defina como `lb_cookie` para usar cookies nas sticky sessions. Essa opção só é aplicável a ambientes com um Application Load Balancer.	`lb_cookie`	`lb_cookie`
UnhealthyThreshold Count	Solicitações consecutivas sem êxito antes de o Elastic Load Balancing alterar o status de integridade da instância.	`5`	2 para `10`

aws:elasticbeanstalk:environment:proxy:staticfiles

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

> **Note**
>
> Esse namespace se aplica a ramificações da plataforma baseadas no Amazon Linux 2 e versões posteriores. Se o ambiente usar uma versão de plataforma baseada na AMI do Amazon Linux (anterior ao Amazon Linux 2), consulte the section called “Opções específicas para a plataforma” para os namespaces de arquivos estáticos específicos da plataforma.

Namespace: **`aws:elasticbeanstalk:environment:proxy:staticfiles`**

Name (Nome)	Valor
O caminho no qual o servidor proxy serve os arquivos. Inicie o valor com `/`. Por exemplo, especifique `/images` para servir arquivos em *`subdomain`* `.elasticbeanstalk.com/images` .	O nome da pasta que contém os arquivos. Por exemplo, especifique `staticimages` para servir arquivos de uma pasta denominada `staticimages` no nível mais alto do seu pacote de origem.

aws:elasticbeanstalk:healthreporting:system

Configure relatórios de integridade aprimorados para o seu ambiente.

Namespace: **`aws:elasticbeanstalk:healthreporting:system`**

Name (Nome)	Descrição	Padrão	Valores válidos

SystemType	O sistema de relatórios de integridade (básicos ou aprimorados). Os relatórios de integridade aprimorados requerem uma função de serviço e uma versão da plataforma versão 2 ou mais nova.
	
	Note
	Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.
		`basic`	`basic`
			
			`enhanced`

ConfigDocument	Um documento JSON que descreve as métricas do ambiente e da instância para CloudWatch publicar.	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos

EnhancedHealthAuthEnabled	Permite a autorização para a API interna que o Elastic Beanstalk usa para comunicar informações de integridade aprimorada das instâncias do ambiente para o serviço do Elastic Beanstalk.
	
	Para obter mais informações, consulte the section called “Funções de integridade aperfeiçoada”.
	
	Note
	Essa opção só é aplicável a relatórios de integridade aprimorados (como quando `SystemType` está definido como `enhanced`).
		`true`	`true`
			
			`false`

HealthCheckSuccessThreshold	Reduz o limite para que as instâncias passem nas verificações de integridade.
	
	Note
	Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console substitui essa opção por um valor recomendado.
		`Ok`	`Ok`
			
			`Warning`
			
			`Degraded`
			
			`Severe`

aws:elasticbeanstalk:hostmanager

Configure as EC2 instâncias em seu ambiente para fazer upload de logs rotativos para o Amazon S3.

Namespace: **`aws:elasticbeanstalk:hostmanager`**

Name (Nome)	Descrição	Padrão	Valores válidos
LogPublicationControl	Copie os arquivos de log das EC2 instâncias da Amazon do seu aplicativo para o bucket do Amazon S3 que está associado ao seu aplicativo.	`false`	`true` `false`

aws:elasticbeanstalk:managedactions

Configure as atualizações gerenciadas da plataforma para o ambiente.

Namespace: **`aws:elasticbeanstalk:managedactions`**

Name (Nome)	Descrição	Padrão	Valores válidos
ManagedActionsEnabled	Ative as atualizações gerenciadas da Plataforma. Ao definir essa opção para `true`, também é necessário especificar `PreferredStartTime` e `UpdateLevel`.	`false`	`true` `false`
PreferredStartTime	Configure uma janela de manutenção para ações gerenciadas em UTC. Por exemplo, `."Tue:09:00"`	Nenhum	Dia e hora no *day*:*hour*:*minute* format.
ServiceRoleForManagedUpdates	O nome de uma função do IAM que o Elastic Beanstalk usa para executar atualizações gerenciadas de plataforma para o ambiente.	Nenhum	Igual a `ServiceRole` ou

Name (Nome)	Descrição	Padrão	Valores válidos
	É possível usar a mesma função especificada para a opção `ServiceRole` do namespace `aws:elasticbeanstalk:environment` ou a função vinculada ao serviço de atualizações gerenciadas da conta. No último caso, se a conta ainda não tiver uma função vinculada ao serviço de atualizações gerenciadas, o Elastic Beanstalk a criará.		`AWSServiceRoleForElasticBeanstalkManagedUpdates`

aws:elasticbeanstalk:managedactions:platformupdate

Configure as atualizações gerenciadas da plataforma para o ambiente.

Namespace: **`aws:elasticbeanstalk:managedactions:platformupdate`**

Name (Nome)	Descrição	Padrão	Valores válidos
UpdateLevel	O nível mais alto de atualização a aplicar com atualizações gerenciadas da plataforma. As plataformas são *`major`* versionadas. *`minor`*. *`patch`*. Por exemplo, a versão 2.0.8 tem uma versão principal 2, uma versão secundária 0 e uma versão de patch 8.	Nenhum	`patch` somente para atualizações de versão de patch. `minor` para atualizações de versão secundária e de patch.

Name (Nome)	Descrição	Padrão	Valores válidos

InstanceRefreshEnabled	Ative substituição de instância semanal.
	
	Isso requer que `ManagedActionsEnabled` seja definido como `true`.
		`false`	`true`
			
			`false`

aws:elasticbeanstalk:monitoring

Configure seu ambiente para encerrar EC2 instâncias que falham nas verificações de integridade.

Namespace: **`aws:elasticbeanstalk:monitoring`**

Name (Nome)	Descrição	Padrão	Valores válidos

Automatically Terminate Unhealthy Instances	Encerre uma instância se ela falhar nas verificações de integridade.
	
	Note
	Essa opção só é compatível com ambientes legados. Ela determina a integridade de uma instância com base em poder acessá-la e em outras métricas com base em instâncias.
	O Elastic Beanstalk não fornece uma maneira de encerrar instâncias automaticamente com
		`true`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
	base na integridade da aplicação.		

aws:elasticbeanstalk:sns:topics

Configure as notificações de seu ambiente.

Namespace: **`aws:elasticbeanstalk:sns:topics`**

Name (Nome)	Descrição	Padrão	Valores válidos
Notification Endpoint	O endpoint em que você deseja ser notificado sobre eventos importantes que afetam a aplicação. **Note** Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e substitui essa opção por um valor recomendado.	Nenhum	

Notification Protocol	O protocolo usado para enviar notificações ao endpoint.	`email`	`http`
			
			`https`
			
			`email`
			
			`email-json`

Name (Nome)	Descrição	Padrão	Valores válidos
			`sqs`
Notification Topic ARN	O nome de recurso da Amazon (ARN) do tópico no qual você se inscreveu.	Nenhum	
Notification Topic Name	O nome do tópico que você assinou.	Nenhum	

aws:elasticbeanstalk:sqsd

Configure a fila do Amazon SQS para um ambiente de operador.

Namespace: **`aws:elasticbeanstalk:sqsd`**

Name (Nome)	Descrição	Padrão	Valores válidos
WorkerQueueURL	A URL da fila em que o daemon no nível de ambiente de trabalho lê as mensagens **Note** Quando você não especific a um valor, a fila que o Elastic Beanstalk cria automaticamente é uma fila padrão do Amazon SQS. Quando você fornece um valor, pode informar o URL de uma fila padrão ou FIFO do Amazon SQS. Lembre-se de que, se você fornecer uma fila FIFO,	gerado automatic amente	Se você não especificar um valor, o Elastic Beanstalk criará automaticamente uma fila.

Name (Nome)	Descrição	Padrão	Valores válidos
	as tarefas periódicas não serão compatíveis.		
HttpPath	O caminho relativo para a aplicação para a qual as mensagens HTTP POST são enviadas.	/	

MimeType	O tipo MIME da mensagem enviada na solicitação HTTP POST.	`application/json`	`application/json`
			`application/x-www-form-urlencoded`
			`application/xml`
			`text/plain`
			Tipo MIME personalizado.

HttpConnections	O número máximo de conexões simultâneas com qualquer aplicativo que esteja dentro de uma EC2 instância da Amazon. **Note** Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e substitui essa opção por um valor recomendado.	`50`	1 para `100`

Name (Nome)	Descrição	Padrão	Valores válidos
ConnectTimeout	O tempo, em segundos, a aguardar por conexões bem-sucedidas com um aplicativo.	5	1 para 60
InactivityTimeout	O tempo, em segundos, a aguardar por uma resposta para uma conexão existente com um aplicativo. A mensagem é reprocessada até que o daemon receba uma resposta 200 (OK) do aplicativo no nível do ambiente de operador ou `RetentionPeriod` expire.	299	1 para 36000
VisibilityTimeout	O tempo, em segundos, em que uma mensagem recebida da fila do Amazon SQS fica bloqueada para processamento. Após o término do tempo configurado, a mensagem ficará novamente visível na fila para qualquer outro daemon ler.	300	0 para 43200
ErrorVisibilityTimeout	O tempo, em segundos, que o Elastic Beanstalk deve aguardar para retornar uma mensagem à fila do Amazon SQS após uma falha com erro explícito ao tentar processá-la.	2 segundos	0 a 43200 segundos
RetentionPeriod	A quantidade de tempo, em segundos, em que uma mensagem é válida e é ativamente processada.	345600	60 para 1209600

Name (Nome)	Descrição	Padrão	Valores válidos
MaxRetries	O número máximo de vezes em que o Elastic Beanstalk tenta enviar a mensagem à aplicação Web que vai processá-la antes de movê-la para a dead-letter queue.	`10`	1 para `100`

aws:elasticbeanstalk:trafficsplitting

Configure implantações de divisão de tráfego para o ambiente.

Esse namespace se aplica quando você define a opção `DeploymentPolicy` do namespace aws:elasticbeanstalk:command como `TrafficSplitting`. Para obter mais informações sobre políticas de implantação, consulte the section called “Opções de implantação”.

Namespace: **`aws:elasticbeanstalk:trafficsplitting`**

Name (Nome)	Descrição	Padrão	Valores válidos
NewVersionPercent	A porcentagem inicial de tráfego de entrada do cliente que o Elastic Beanstalk desloca para instâncias de ambiente executando a nova versão da aplicação que está sendo implantada.	`10`	1 para `100`
EvaluationTime	O período, em minutos, que o Elastic Beanstalk aguarda após uma implantação íntegra inicial antes de prosseguir para deslocar todo o tráfego de entrada de cliente para a nova versão da aplicação que está sendo implantada.	`5`	`3` para `600`

aws:elasticbeanstalk:xray

Execute o AWS X-Ray daemon para retransmitir informações de rastreamento do seu aplicativo integrado X-Ray.

Namespace: **`aws:elasticbeanstalk:xray`**

Name (Nome)	Descrição	Padrão	Valores válidos
`XRayEnabled`	Defina como `true` para executar o daemon do X-Ray nas instâncias do ambiente.	`false`	`true` `false`

aws:elb:healthcheck

Configure as verificações de integridade de um Classic Load Balancer.

Namespace: **`aws:elb:healthcheck`**

Name (Nome)	Descrição	Padrão	Valores válidos
HealthyThreshold	O número de solicitações consecutivas bem-sucedidas antes de o Elastic Load Balancing alterar o status de integridade da instância.	`3`	2 para `10`
Interval	O intervalo em que o Elastic Load Balancing verifica a integridade das EC2 instâncias da Amazon do seu aplicativo.	`10`	5 para `300`
Timeout	A quantidade de tempo, em segundos, que o Elastic Load Balancing aguardará por uma resposta antes de considerar a instância não responsiva.	`5`	2 para `60`
UnhealthyThreshold	Solicitações consecutivas sem êxito antes de o Elastic Load Balancing alterar o status de integridade da instância.	`5`	2 para `10`
(obsoleto) Target	O destino em uma instância de back-end para a qual as verificações de integridade são enviadas. Em vez dessa	`TCP:80`	Alvo no formato*PROTOCOL*

Name (Nome)	Descrição	Padrão	Valores válidos
	opção, use `Application Healthcheck URL` no namespace `aws:elasticbeanstalk:application`.		*PORT / PATH*

aws:elb:loadbalancer

Configure o Classic Load Balancer do ambiente.

Várias das opções neste namespace foram substituídas por opções específicas do listener no namespace aws:elb:listener. Como essas opções que não são mais suportadas, você só pode configurar dois listeners (um seguro e um inseguro) em portas padrão.

Namespace: **`aws:elb:loadbalancer`**

Name (Nome)	Descrição	Padrão	Valores válidos

CrossZone	Configure o balanceador de carga para rotear o tráfego uniformemente em todas as instâncias em todas as zonas de disponibilidade, em vez de apenas dentro de cada região.
	
	Note
	Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substituem essa opção por um valor recomendado.
		`false`	`true`
			
			`false`

SecurityGroups	Atribua um ou mais grupos de segurança que você criou ao balanceador de carga.	Nenhum	Um ou mais grupos de

Name (Nome)	Descrição	Padrão	Valores válidos
	Obrigatório se `DisableDefaultEC2SecurityGroup` (aws:autoscaling:launchconfiguration) estiver definido como `true`. Ambientes com balanceamento de carga que optaram por não participar do grupo de segurança padrão do EC2 Elastic Beanstalk devem fornecer essa opção a um ou mais grupos de segurança. Para obter mais informações, consulte Gerenciando grupos EC2 de segurança.		segurança IDs.
ManagedSecurityGroup	Atribua um security group existente ao balanceador de carga do seu ambiente, em vez de criar um novo. Para usar essa configuração, atualize a configuração `SecurityGroups` neste namespace para incluir a ID do security group e remova a ID do grupo de segurança criada automaticamente, se uma foi criada. Para permitir o tráfego do balanceador de carga para as EC2 instâncias do seu ambiente, o Elastic Beanstalk adiciona uma regra ao grupo de segurança das instâncias que permite o tráfego de entrada do grupo de segurança gerenciado.	Nenhum	Um ID de security group.

(obsoleto) LoadBalancerHTTPPort	Porta de escuta do listener não seguro.	`80`	`OFF`
			
			`80`

(obsoleto) LoadBalancerPortProtocol	O protocolo para usar no listener não seguro.	`HTTP`	`HTTP`
			
			`TCP`

Name (Nome)	Descrição	Padrão	Valores válidos

(obsoleto) LoadBalancerHTTPSPort	A p\orta de escuta do listener seguro.	`OFF`	`OFF`
			`443`
			`8443`

(obsoleto) LoadBalancerSSLPortProtocol	O protocolo para usar no listener seguro.	`HTTPS`	`HTTPS` `SSL`
(obsoleto) SSLCertificateId	O Amazon Resource Name (ARN) de um certificado SSL para vincular ao listener seguro.	Nenhum	

aws:elb:listener

Configure o listener padrão (porta 80) em um Classic Load Balancer.

Namespace: **`aws:elb:listener`**

Name (Nome)	Descrição	Padrão	Valores válidos
ListenerProtocol	O protocolo usado pelo listener.	`HTTP`	`HTTP TCP`
InstancePort	A porta que esse ouvinte usa para se comunicar com as EC2 instâncias.	80	1 para 65535
InstanceProtocol	O protocolo que esse ouvinte usa para se comunicar com as EC2 instâncias. Deve estar na mesma camada de protocolo de internet que `ListenerProtocol` . Também deve ter o mesmo nível de segurança que qualquer outro listener usando o mesmo `InstancePort` que esse listener.	`HTTP` quando `ListenerProtocol` é `HTTP` `TCP` quando	`HTTP` ou `HTTPS` quando `ListenerProtocol` é `HTTP` ou `HTTPS`

Name (Nome)	Descrição	Padrão	Valores válidos
	Por exemplo, se `ListenerProtocol` for `HTTPS` (camada de aplicação, usando uma conexão segura), você poderá definir `InstanceProtocol` como `HTTP` (também na camada de aplicação, usando uma conexão não segura). Se, além disso, você definir `InstancePort` como `80`, precisará definir `InstanceProtocol` como `HTTP` em todos os outros listeners, com `InstancePort` definido como `80`.	`ListenerProtocol` é `TCP`	`TCP` ou `SSL` quando `ListenerProtocol` é `TCP` ou `SSL`
PolicyNames	Uma lista separada por vírgulas de nomes de políticas para aplicar à porta desse listener. Em vez disso, recomendamos que você use a LoadBalancerPorts opção do aws:elb:policies namespace.	Nenhum	
ListenerEnabled	Especifica se esse listener está ativado. Se você especificar `false`, o listener não será incluído no balanceador de carga.	`true`	`true` `false`

aws:elb:listener:listener_port

Configure listeners adicionais em um Classic Load Balancer.

Namespace: **`aws:elb:listener:`*`listener_port`***

Name (Nome)	Descrição	Padrão	Valores válidos
ListenerProtocol	O protocolo usado pelo listener.	`HTTP`	`HTTP` `HTTPS` `TCP` `SSL`

Name (Nome)	Descrição	Padrão	Valores válidos
InstancePort	A porta que esse ouvinte usa para se comunicar com as EC2 instâncias.	O mesmo que *listene* `port` .	1 para 65535

InstanceProtocol	O protocolo que esse ouvinte usa para se comunicar com as EC2 instâncias.
	
	Deve estar na mesma camada de protocolo de internet que `ListenerProtocol` . Também deve ter o mesmo nível de segurança que qualquer outro listener usando o mesmo `InstancePort` que esse listener.
	
	Por exemplo, se `ListenerProtocol` for `HTTPS` (camada de aplicação, usando uma conexão segura), você poderá definir `InstanceProtocol` como `HTTP` (também na camada de aplicação, usando uma conexão não segura). Se, além disso, você definir `InstancePort` como `80`, precisará definir `InstanceProtocol` como `HTTP` em todos os outros listeners, com `InstancePort` definido como `80`.
		`HTTP` quando `ListenerProtocol` é `HTTP` ou `HTTPS`
		
		`TCP` quando `ListenerProtocol` é `TCP` ou `SSL`
			`HTTP` ou `HTTPS` quando `ListenerProtocol` é `HTTP` ou `HTTPS`
			
			`TCP` ou `SSL` quando `ListenerProtocol` é `TCP` ou `SSL`

PolicyNames	Uma lista separada por vírgulas de nomes de políticas para aplicar à porta desse listener. Em vez disso, sugerimos que você use a LoadBalancerPorts opção do aws:elb:policies namespace.	Nenhum	
SSLCertificateId	O nome de recurso da Amazon (ARN) de um certificado SSL a ser vinculado ao listener.	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos
ListenerEnabled	Especifica se esse listener está ativado. Se você especificar `false`, o listener não será incluído no balanceador de carga.	`true` se qualquer outra opção estiver definida. `false`, se for o contrário.	`true` `false`

aws:elb:policies

Modifique o padrão global de perdurabilidade e as políticas globais de um Classic Load Balancer.

Namespace: **`aws:elb:policies`**

Name (Nome)	Descrição	Padrão	Valores válidos

ConnectionDraining Enabled	Especifica se o balanceador de carga mantém as conexões existentes com a instâncias que deixaram de ser íntegras ou tiveram o registro cancelado para concluir as solicitações em andamento.
	
	Note
	Se você usar o console do Elastic Beanstalk ou a CLI do EB para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e a CLI do EB substitue
		`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
	m essa opção por um valor recomendado.		
ConnectionDraining Timeout	O número máximo de segundos que o balanceador de carga mantém conexões existentes com uma instância durante a drenagem da conexão antes de forçar o fechamento das conexões. **Note** Se você usar o console do Elastic Beanstalk para criar um ambiente, não será possível definir essa opção em um arquivo de configuração. O console e substitui essa opção por um valor recomendado.	`20`	1 para `3600`
ConnectionSettingIdleTimeout	A quantidade de tempo, em segundos, que o balanceador de carga aguarda que todos os dados sejam enviados ou recebidos pela conexão. Se nenhum dado tiver sido enviado ou recebido após esse período, o balanceador de carga fechará a conexão.	`60`	1 para `3600`
LoadBalancerPorts	Uma lista separada por vírgulas das portas do listener em que a política padrão (`AWSEB-ELB-StickinessPolicy`) se aplica.	Nenhum	Você pode usar `:all` para indicar todas as portas do listener

Name (Nome)	Descrição	Padrão	Valores válidos
Stickiness Cookie Expiration	O tempo, em segundos, pelo qual cada cookie é válido. Usa a política padrão (`AWSEB-ELB-StickinessPolicy`).	`0`	`0` para `1000000`
Stickiness Policy	Vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor. Usa a política padrão (`AWSEB-ELB-StickinessPolicy`).	`false`	`true false`

aws:elb:policies:policy_name

Crie políticas adicionais de balanceador de carga para um Classic Load Balancer.

Namespace: **`aws:elb:policies:`*`policy_name`***

Name (Nome)	Descrição	Padrão	Valores válidos
CookieName	O nome do cookie gerado pelo aplicativo que controla os tempos de vida da sessão de uma política `AppCookieStickinessPolicyType` . Essa política só pode ser associada aos HTTP/HTTPS ouvintes.	Nenhum	
InstancePorts	Uma lista separada por vírgulas das portas de instâncias em que a política aplica.	Nenhum	Uma lista de portas, ou `:all`
LoadBalancerPorts	Uma lista separada por vírgulas das portas de listeners em que a política aplica.	Nenhum	Uma lista de portas, ou `:all`

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyProtocol	Para uma política `ProxyProtocolPolicyType`, especifique se deseja ou não incluir o endereço IP e a porta da solicitação de origem de mensagens TCP. Essa política só pode ser associada aos TCP/SSL ouvintes.	Nenhum	`true false`
PublicKey	O conteúdo de uma chave pública para uma política `PublicKeyPolicyType` a ser utilizada na autenticação do servidor ou servidores de backend. Esta política não pode ser aplicada diretamente a servidores de backend ou listeners. Ela deve fazer parte de uma política de `BackendServerAuthenticationPolicyType`.	Nenhum	
PublicKeyPolicyNames	Uma lista separada por vírgulas de nomes de políticas (com base em políticas `PublicKeyPolicyType`) para uma política `BackendServerAuthenticationPolicyType` que controla a autenticação em um ou mais servidores de backend. Essa política só pode ser associada a servidores de backend que estejam usando HTTPS/SSL.	Nenhum	
SSLProtocols	Uma lista separada por vírgulas de protocolos SSL a serem habilitados para uma política de `SSLNegotiationPolicyType` que define as cifras e os protocolos que são aceitos pelo balanceador de carga. Essa política só pode ser associada aos HTTPS/SSL ouvintes.	Nenhum	

Name (Nome)	Descrição	Padrão	Valores válidos
SSLReferencePolicy	O nome de uma política de segurança predefinida que segue as melhores práticas de AWS segurança e que você deseja ativar para uma `SSLNegotiationPolicyType` política que define as cifras e os protocolos aceitos pelo balanceador de carga. Essa política só pode ser associada aos HTTPS/SSL ouvintes.	Nenhum	
Stickiness Cookie Expiration	O tempo, em segundos, pelo qual cada cookie é válido.	`0`	`0` para `1000000`
Stickiness Policy	Vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.	`false`	`true false`

aws:elbv2:listener:default

Configure o listener padrão (porta 80) em um Application Load Balancer ou um Network Load Balancer.

Esse namespace não se aplica a um ambiente que usa um balanceador de carga compartilhado. Os balanceadores de carga compartilhados não têm um listener padrão.

Namespace: **`aws:elbv2:listener:default`**

Name (Nome)	Descrição	Padrão	Valores válidos
DefaultProcess	O nome do processo ao qual encaminhar o tráfego quando	`default`	Um nome de processo.

Name (Nome)	Descrição	Padrão	Valores válidos
	nenhuma regra se aplica.		
ListenerEnabled	Defina como `false` para desativar o listener. Você pode usar essa opção para desativar o listener padrão na porta 80.	`true`	`true` `false`
Protocol	Protocolo de tráfego a processar.	Com Application Load Balancer: `HTTP` Com Network Load Balancer: `TCP`	Com Application Load Balancer: `HTTP`, `HTTPS` Com Network Load Balancer: `TCP`
Rules	A lista de regras que se aplicam ao listener Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum	Uma lista separada por vírgulas de nomes de regras.
SSLCertificateArns	O nome de recurso da Amazon (ARN) do certificado SSL a ser vinculado ao listener. Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum	O ARN de um certifica do armazenado no IAM ou no ACM.

Name (Nome)	Descrição	Padrão	Valores válidos
SSLPolicy	Especifique uma política de segurança para aplicar ao listener. Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum (padrão do ELB)	O nome de uma política de segurança do load balancer.

aws:elbv2:listener:listener_port

Configure listeners adicionais em um Application Load Balancer ou um Network Load Balancer.

> **Note**
>
> Para um Application Load Balancer compartilhado, você pode especificar apenas a opção `Rule`. As outras opções não são aplicáveis aos balanceadores de carga compartilhados.

Namespace: **`aws:elbv2:listener:`*`listener_port`***

Name (Nome)	Descrição	Padrão	Valores válidos
DefaultProcess	O nome do processo ao qual encaminhar o tráfego quando não houver regras correspondentes.	`default`	Um nome de processo.

ListenerEnabled	Defina como `false` para desativar o listener. Você pode usar essa opção para desativar o listener padrão na porta 80.	`true`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
Protocol	Protocolo de tráfego a processar.	Com Application Load Balancer: `HTTP` Com Network Load Balancer: `TCP`	Com Application Load Balancer: `HTTP`, `HTTPS` Com Network Load Balancer: `TCP`
Rules	Lista de regras que se aplicam ao listener Essa opção só é aplicável a ambientes com um Application Load Balancer. Se o ambiente usar um Application Load Balancer compartil hado e você não especificar essa opção para nenhum listener, o Elastic Beanstalk associará automaticamente a regra `default` a um listener de porta 80.	Nenhum	Uma lista separada por vírgulas de nomes de regras.
SSLCertificateArns	O nome de recurso da Amazon (ARN) do certificado SSL a ser vinculado ao listener. Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum	O ARN de um certifica do armazenado no IAM ou no ACM.

Name (Nome)	Descrição	Padrão	Valores válidos
SSLPolicy	Especifique uma política de segurança para aplicar ao listener. Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum (padrão do ELB)	O nome de uma política de segurança do load balancer.

aws:elbv2:listenerrule:rule_name

Defina regras de listener para um Application Load Balancer. Se uma solicitação corresponder aos nomes ou caminhos do host em uma regra, o balanceador de carga a encaminhará para o processo especificado. Para usar uma regra, adicione-a a um listener com a opção `Rules` no namespace `aws:elbv2:listener:listener_port`.

> **Note**
>
> Este namespace não é aplicável a ambientes com um Network Load Balancer.

Namespace: **`aws:elbv2:listenerrule:rule_name`**

Name (Nome)	Descrição	Padrão	Valores válidos
HostHeaders	Uma lista de nomes de host para corresponder. Por exemplo, `.my.example.com`	Load balancer dedicado: nenhum Load balancer compartilhado: o	Cada nome pode conter até 128 caracteres. Um padrão pode incluir letras maiúsculas e minúsculas, números, hífens (–) e até três caracteres curinga (*

Name (Nome)	Descrição	Padrão	Valores válidos
		CNAME do ambiente	corresponde a zero ou mais caractere s; ? corresponde a exatamente um caractere). Você pode listar mais de um nome, cada um separado por uma vírgula. O Applicati on Load Balancer é compatível com até cinco regras `HostHeader` e `PathPattern` combinadas. Para obter mais informações, consulte Condições de host no Guia do Usuário para o Application Load Balancers.

Name (Nome)	Descrição	Padrão	Valores válidos
PathPatterns	Os padrões de caminho a serem adotados (por exemplo, `/img/*`). Essa opção só é aplicável a ambientes com um Application Load Balancer.	Nenhum	Cada padrão pode conter até 128 caracteres. Um padrão pode incluir letras maiúsculas e minúsculas, números, hífens (–) e até três caracteres curinga (* corresponde a zero ou mais caracteres; ? corresponde a exatamente um caractere). É possível adicionar vários padrões de caminho separados por vírgulas. O Application Load Balancer é compatível com até cinco regras `HostHeader` e `PathPattern` combinadas. Para obter mais informações, consulte as Condições de caminho no Guia do Usuário para Application Load Balancers.

Name (Nome)	Descrição	Padrão	Valores válidos
Priority	A precedência dessa regra quando há várias regras correspondentes. O menor número tem precedência. Duas regras não podem ter a mesma prioridade. Com um balanceador de carga compartilhado, o Elastic Beanstalk trata as prioridades de regras como relativas em ambientes de compartilhamento e as mapeia para prioridades absolutas durante a criação.	1	1 para `1000`
Process	O nome do processo ao qual encaminhar o tráfego quando essa regra corresponde à solicitação.	`default`	Um nome de processo.

aws:elbv2:loadbalancer

Configure um Application Load Balancer.

Para um balanceador de carga compartilhado, somente as opções `SharedLoadBalancer` e `SecurityGroups` são válidas.

> **Note**
>
> Este namespace não é aplicável a ambientes com um Network Load Balancer.

Namespace: **`aws:elbv2:loadbalancer`**

Name (Nome)	Descrição	Padrão	Valores válidos
AccessLogsS3Bucket	O bucket do Amazon S3 no qual os logs de acesso são armazenados. O bucket deve estar na mesma região	Nenhum	Nome de um bucket.

Name (Nome)	Descrição	Padrão	Valores válidos
	que o ambiente e permitir acesso de gravação ao balanceador de carga.		
AccessLogsS3Enabled	Habilite o acesso ao armazenamento de logs.	`false`	`true` `false`
AccessLogsS3Prefix	O prefixo para prefixar nomes de logs de acesso. Por padrão, o balanceador de carga carrega os registros em um diretório chamado AWSLogs no bucket que você especificar. Especifique um prefixo para colocar o AWSLogs diretório dentro de outro diretório.	Nenhum	
IdleTimeout	A quantidade de tempo, em segundos, para aguardar a conclusão de uma solicitação antes de fechar conexões com o cliente e a instância.	Nenhum	1 para `3600`

Name (Nome)	Descrição	Padrão	Valores válidos
ManagedSecurityGroup	Atribua um grupo de segurança existente ao balanceador de carga do ambiente em vez de criar um novo. Para usar essa configuração, atualize a definição `SecurityGroups` neste namespace para incluir o ID do security group e remover o ID do security group criado automaticamente, se houver um. Para permitir o tráfego do balanceador de carga para as EC2 instâncias do seu ambiente, o Elastic Beanstalk adiciona uma regra ao grupo de segurança de suas instâncias que permite o tráfego de entrada do grupo de segurança gerenciado.	O security group que o Elastic Beanstalk cria para o balanceador de carga.	Um ID de security group.

Name (Nome)	Descrição	Padrão	Valores válidos
SecurityGroups	Lista de security groups para anexar ao balanceador de carga. Obrigatório se `DisableDefaultEC2SecurityGroup` (aws:autoscaling:launchconfiguration) estiver definido como `true`. Ambientes com balanceamento de carga que optaram por não participar do grupo de segurança padrão do EC2 Elastic Beanstalk devem fornecer essa opção a um ou mais grupos de segurança . Para obter mais informações, consulte Gerenciando grupos EC2 de segurança. Para um balanceador de carga compartilhado, se você não especificar esse valor, o Elastic Beanstalk verificará se um grupo de segurança existente que ele gerencia já está conectado ao balanceador de carga. Se um não estiver conectado ao balanceador de carga, o Elastic Beanstalk cria um security group e o anexará ao balanceador de carga. O Elastic Beanstalk exclui esse grupo de segurança quando o último ambiente que compartilha o balanceador de carga é encerrado. Os grupos de segurança do balanceador de carga são usados para configurar a regra de entrada	O grupo de segurança que o Elastic Beanstalk cria para o balanceador de carga.	Lista separada por vírgulas do grupo de segurança. IDs

Name (Nome)	Descrição	Padrão	Valores válidos
	do grupo de segurança de EC2 instâncias da Amazon.		

Name (Nome)	Descrição	Padrão	Valores válidos
SharedLoadBalancer	O nome de recurso da Amazon (ARN) do balanceador de carga. Essa opção é relevante apenas para um Application Load Balancer. Ela é necessária quando a opção `LoadBalancerIsShared` do namespace aws:elasticbeanstalk:environment está definida como `true`. Não é possível alterar o ARN do balanceador de carga compartilhado após a criação do ambiente. Critérios para um valor válido: • Ele deve ser um balanceador de carga ativo e válido na AWS região em que o ambiente está localizado. • Ele deve estar na mesma Amazon Virtual Private Cloud (Amazon VPC) que o ambiente. • Ele não pode ser um balanceador de carga criado pelo Elastic Beanstalk como balanceador de carga dedicado para outro ambiente. Você pode identificar esses balanceadores de carga dedicados usando o prefixo `awseb-`. Exemplo: `arn:aws:elasticloadbalancing:us-east-2:123456789012:lo`	Nenhum	ARN de um balanceador de carga válido que atende a todos os critérios descritos aqui.

Name (Nome)	Descrição	Padrão	Valores válidos
	`adbalancer/app/Fro ntEndLB/0dbf78d8ad96abbc`		

aws:rds:dbinstance

Configure uma instância de banco de dados anexada do Amazon RDS.

Namespace: **`aws:rds:dbinstance`**

Name (Nome)	Descrição	Padrão	Valores válidos
DBAllocatedStorage	O tamanho de armazenamento do banco de dados alocado, especificado em gigabytes.	MySQL: `5` Oracle: `10` sqlserver-se: `200` sqlserver-ex: `30` sqlserver-web: `30`	MySQL: `5-1024` Oracle: `10-1024` sqlserver: não pode ser modificado

DBDeletionPolicy	Especifica se é necessário reter, excluir ou fazer um snapshot da instância de banco de dados quando um ambiente é terminado.
	Essa opção funciona em conjunto com `HasCoupledDatabase` , também uma opção desse namespace.
		`Delete`	`Delete`
			`Retain`
			`Snapshot`

Name (Nome)	Descrição	Padrão	Valores válidos
	Warning A exclusão de uma instância de banco de dados resulta em perda de dados permanente.		

DBEngine	O nome do mecanismo de banco de dados a utilizar nessa instância.	`mysql`	`mysql`
			`oracle-se1`
			`sqlserver-ex`
			`sqlserver-web`
			`sqlserver-se`
			`postgres`

DBEngineVersion	O número da versão do mecanismo de banco de dados.	`5.5`	
DBInstanceClass	O tipo de instância de banco de dados.	`db.t2.micro` (`db.m1.large` para um ambiente que não está em execução em uma Amazon VPC)	Para obter mais informações, consulte Classe de instância do banco de dados no Guia do usuário do Amazon Relational Database Service.

Name (Nome)	Descrição	Padrão	Valores válidos
DBPassword	O nome da senha mestra do usuário para a instância de banco de dados.	Nenhum	
DBSnapshotIdentifier	O identificador para restaurar o DB snapshot.	Nenhum	
DBUser	O nome do usuário mestre da instância do banco de dados.	ebroot	

Name (Nome)	Descrição	Padrão	Valores válidos

HasCoupledDatabase	Especifica se uma instância de banco de dados está acoplada ao ambiente. Se definida como `true`, o Elastic Beanstalk cria uma nova instância de banco de dados acoplada ao seu ambiente. Se definida como `false`, o Elastic Beanstalk inicia o desacoplamento da instância de banco de dados do seu ambiente.
	
	Essa opção funciona em conjunto com `DBDeletionPolicy` , também uma opção desse namespace.
	
	Note
	Nota: se você voltar esse valor para `true` após desacoplar o banco de dados anterior, o Elastic Beanstalk cria um novo banco de dados com as configurações de banco de dados anteriores. No entanto, para manter a segurança do seu ambiente, ele não retém as configurações existentes de `DBUser` e `DBPassword` . Você precisa especificar `DBUser` e `DBPassword` novamente.
		`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
MultiAZDatabase	Especifica se a implantação Multi-AZ de uma instância de banco de dados precisa ser criada. Para obter mais informações sobre implantações Multi-AZ com o Amazon Relational Database Service (RDS), consulte as Regiões e zonas de disponibilidade no Guia do usuário do Amazon Relational Database Service.	`false`	`true` `false`

Opções específicas para a plataforma

Algumas plataformas do Elastic Beanstalk definem namespaces de opção que são específicos da plataforma. Esses namespaces e suas opções estão listados abaixo para cada plataforma.

> **Note**
>
> Anteriormente, em versões de plataforma baseadas na AMI do Amazon Linux (anterior ao Amazon Linux 2), os dois recursos a seguir e seus respectivos namespaces eram considerados recursos específicos da plataforma e foram listados aqui por plataforma:
>
> - Configuração de proxy para arquivos estáticos – `aws:elasticbeanstalk:environment:proxy:staticfiles`
> - AWS X-Ray suporte — `aws:elasticbeanstalk:xray`
>
> Nas versões da plataforma do Amazon Linux 2, o Elastic Beanstalk implementa esses recursos de forma consistente em todas as plataformas de suporte. O namespace relacionado agora está listado na página the section called “Opções gerais”. Mantivemos somente a menção a eles nesta página para plataformas que tinham namespaces de nomes diferentes.

Plataformas

- Opções para a plataforma Docker

- Opções para a plataforma Go
- Opções para a plataforma Java SE
- Java com opções para a plataforma Tomcat
- Opções da plataforma .NET Core no Linux
- Opções para a plataforma .NET
- Opções para a plataforma Node.js
- Opções para a plataforma PHP
- Opções para a plataforma Python
- Opções para a plataforma Ruby

Opções para a plataforma Docker

As opções de configuração específicas do Docker a seguir são aplicáveis a plataformas do Docker e do Docker pré-configurado.

> **Note**
>
> Essas opções de configuração não se aplicam a:
>
> - Plataforma Docker (Amazon Linux 2) com o Docker Compose
> - Plataforma Docker de vários contêineres (Amazon Linux AMI AL1) — essa plataforma foi descontinuada

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Especifica o servidor web a ser usado como um proxy.	`nginx`	`nginx` `none`: AMI do Amazon Linux e Docker com DC somente

Opções para a plataforma Go

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **`aws:elasticbeanstalk:container:golang:staticfiles`**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

Nome	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos. Exemplo: `/images` para servir arquivos em *`subdomain`* `.eleasticbeanstalk.com/images` .	Nome da pasta que contém os arquivos. Exemplo: `staticimages` para servir arquivos de uma pasta denominada `staticimages` no nível mais alto do seu pacote de origem.

Opções para a plataforma Java SE

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **`aws:elasticbeanstalk:container:java:staticfiles`**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

Nome	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos. Exemplo: `/images` para servir arquivos em *`subdomain`* `.elasticbeanstalk.com/images` .	Nome da pasta que contém os arquivos. Exemplo: `staticimages` para servir arquivos de uma pasta denominada `staticimages` no nível mais alto do seu pacote de origem.

Java com opções para a plataforma Tomcat

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name (Nome)	Descrição	Padrão	Valores válidos
JDBC_CONNECTION_STRING	A string de conexão a um banco de dados externo.	n/a	n/a

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespace: **`aws:elasticbeanstalk:container:tomcat:jvmoptions`**

Name (Nome)	Descrição	Padrão	Valores válidos
JVM Options	Passe as opções da linha de comando para a JVM no startup.	n/a	n/a
Xmx	Tamanhos máximos de heaps da JVM.	`256m`	n/a
XX:MaxPermSize	Seção do heap da JVM usada para armazenar definições de classe e metadados associados. **Note** Esta opção só se aplica a versões Java anteriores ao Java 8 e não é	`64m`	n/a

Name (Nome)	Descrição	Padrão	Valores válidos
	compatível com plataformas Tomcat do Elastic Beanstalk baseadas no Amazon Linux 2 ou posterior.		
Xms	Tamanhos iniciais de heap da JVM.	`256m`	n/a
`optionName`	Especifique as opções arbitrárias de JVM além daqueles definidos pela plataforma Tomcat.	n/a	n/a

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name (Nome)	Descrição	Padrão	Valores válidos

GzipCompression	Defina como `false` para desativar a compactação de resposta.
	
	Válido somente em versões da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2).
		`true`	`true`
			
			`false`

ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente. Se você definir essa opção como `apache`, o Elastic Beanstalk usará o Apache 2.4.
	
	Defina como `apache/2.2` se seu aplicativo não estiver pronto para migrar do Apache 2.2 devido a definições de configuração de proxy incompatíveis. Esse valor só é válido nas versões da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2).
		`nginx` (Amazon Linux 2)
		
		`apache` (AMI do Amazon Linux)
			`apache`
			
			`apache/2.2` – somente AMI do Amazon Linux
			
			`nginx`

Name (Nome)	Descrição	Padrão	Valores válidos
	Defina como `nginx` para usar o nginx. Este é o padrão a partir das versões da plataforma do Amazon Linux 2. Para obter mais informações, consulte Configurar o servidor de proxy.		

Opções da plataforma .NET Core no Linux

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name (Nome)	Descrição	Padrão	Valores válidos

ProxyServer	Especifica o servidor web a ser usado como um proxy.	`nginx`	`nginx`
			
			`none`

Opções para a plataforma .NET

Namespace: **`aws:elasticbeanstalk:container:dotnet:apppool`**

Name (Nome)	Descrição	Padrão	Valores válidos

Target Runtime	Escolha a versão do. NET Framework para o seu aplicativo.	`4.0`	`2.0`
			
			`4.0`

Enable 32-bit Applications	Defina como `True` para executar aplicativos de 32 bits.	`False`	`True`
			
			`False`

Opções para a plataforma Node.js

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name (Nome)	Descrição	Padrão	Valores válidos

ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente.	`nginx`	`apache`
			
			`nginx`

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **`aws:elasticbeanstalk:container:nodejs`**

Name (Nome)	Descrição	Padrão	Valores válidos
NodeCommand	Comando usado para iniciar o aplicativo Node.js. Se uma string vazia for especificada, `app.js` será usado, seguido por `server.js` e `npm start`, nesta ordem.	""	n/a
NodeVersion	Versão do Node.js. Por exemplo, `4.4.6` As versões do Node.js compatíveis variam entre as versões da plataforma do Node.js. Consulte Node.js no documento Platforms do AWS Elastic Beanstalk para obter uma lista das versões compatíveis no momento. **Note** Quando o suporte oferecido para a versão do Node.js que você estiver usando for removido da plataforma, será necessário alterar ou remover a configuração da versão antes de	varia	varia

Name (Nome)	Descrição	Padrão	Valores válidos
	fazer a atualização da plataforma. Isso pode ocorrer quando uma vulnerabilidade de segurança é identificada para uma ou mais versões do Node.js. Quando isso acontece, a tentativa de atualizar para uma nova versão da plataforma que não suporta a configuração NodeVersionfalha. Para evitar a necessidade de criar um novo ambiente, altere a opção de NodeVersionconfiguração para uma versão do Node.js compatível com a versão antiga e a nova da plataforma, ou remova a configuração da opção e execute a atualização da plataforma.		
GzipCompression	Especifica se a compactação gzip está habilitada. Se ProxyServer estiver definido comonone, a compactação gzip será desativada.	`false`	`true` `false`

ProxyServer	Especifica qual servidor da web deve ser usado em conexões via proxy ao Node.js. Se ProxyServer estiver definido comonone, os mapeamentos de arquivos estáticos não entrarão em vigor e a compactação gzip será desativada.	`nginx`	`apache`
			`nginx`
			`none`

Namespace: **`aws:elasticbeanstalk:container:nodejs:staticfiles`**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

> **Note**
>
> As configurações de arquivos estáticos não são aplicáveis se `aws:elasticbeanstalk:container:nodejs::ProxyFiles` estiver definido como `none`.

Nome	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos. Exemplo: `/images` para servir arquivos em *`subdomain`* `.eleasticbeanstalk.com/images` .	Nome da pasta que contém os arquivos. Exemplo: `staticimages` para servir arquivos de uma pasta denominada `staticimages` no nível mais alto do seu pacote de origem.

Opções para a plataforma PHP

Namespace: **`aws:elasticbeanstalk:container:php:phpini`**

Name (Nome)	Descrição	Padrão	Valores válidos
document_root	Especifique o diretório filho do seu projeto que é tratado como raiz da web voltada para o público.	`/`	Uma string em branco é tratada como `/` ou especifica uma string começando com `/`

Name (Nome)	Descrição	Padrão	Valores válidos
memory_limit	Quantidade de memória alocada para o ambiente PHP.	`256M`	n/a

zlib.output_compression	Especifica se o PHP deve usar compactação para saída.	`Off`	`On`
			`Off`
			`true`
			`false`

allow_url_fopen	Especifica se as funções de arquivos do PHP têm permissão para recuperar dados de locais remotos, como sites ou servidores de FTP.	`On`	`On`
			`Off`
			`true`
			`false`

display_errors	Especifica se as mensagens de erro devem ser parte da saída.	`Off`	`On` `Off`
max_execution_time	Define o tempo máximo em segundos que um script tem permissão para ser executado antes que o ambiente o encerre.	`60`	`0` a `9223372036854775807` (PHP_INT_MAX)
composer_options	Define as opções personalizadas a serem usadas ao instalar dependências usando o Composer por meio do comando composer.phar install. Para obter mais informações, consulte installl no site getcomposer.org.	n/a	n/a

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name (Nome)	Descrição	Padrão	Valores válidos
ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente.	`nginx`	`apache` `nginx`

> **Note**
>
> Para obter mais informações sobre a plataforma PHP, consulte Usar a plataforma PHP do Elastic Beanstalk.

Opções para a plataforma Python

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name (Nome)	Descrição	Padrão	Valores válidos
DJANGO_SETTINGS_MODULE	Especifica qual arquivo de configurações deve ser usado.	n/a	n/a

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespace: **`aws:elasticbeanstalk:container:python`**

Name (Nome)	Descrição	Padrão	Valores válidos
WSGIPath	O arquivo que contém o aplicativo WSGI. Este arquivo deve ter um `application` chamável.	Nas versões da plataforma Python do Amazon Linux 2:	n/a

Name (Nome)	Descrição	Padrão	Valores válidos
		`application` Nas versões da plataforma Python da AMI do Amazon Linux: `application.py`	
NumProcesses	O número de processos daemon que deve ser iniciado para o grupo de processo ao executar aplicativos WSGI.	1	n/a
NumThreads	O número de threads a serem criados para lidar com as solicitações em cada processo daemon no processo de grupo ao executar aplicativos WSGI.	15	n/a

Namespace: **`aws:elasticbeanstalk:environment:proxy`**

Name (Nome)	Descrição	Padrão	Valores válidos

ProxyServer	Defina o proxy a ser usado nas instâncias do ambiente.	`nginx`	`apache`
			
			`nginx`

Opções de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2)

Namespace: **`aws:elasticbeanstalk:container:python:staticfiles`**

Você pode usar o namespace a seguir para configurar o servidor de proxy para atender arquivos estáticos. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo. Isso reduz o número de solicitações que seu aplicativo deve processar.

Mapeie um caminho fornecido pelo servidor de proxy para uma pasta no código-fonte que contenha ativos estáticos. Cada opção definida neste namespace mapeia um caminho diferente.

Por padrão, o servidor de proxy em um ambiente Python atende todos os arquivos de uma pasta chamada `static` no caminho `/static`.

Namespace: **`aws:elasticbeanstalk:container:python:staticfiles`**

Name (Nome)	Value (Valor)
Caminho em que o servidor de proxy servirá os arquivos. Exemplo: `/images` para servir arquivos em *`subdomain`* `.elasticbeanstalk.com/images` .	Nome da pasta que contém os arquivos. Exemplo: `staticimages` para servir arquivos de uma pasta denominada `staticimages` no nível mais alto do seu pacote de origem.

Opções para a plataforma Ruby

Namespace: **`aws:elasticbeanstalk:application:environment`**

Name (Nome)	Descrição	Padrão	Valores válidos

RAILS_SKIP_MIGRATIONS	Especifica se `` `rake db:migrate` `` deve ser executado em nome dos aplicativos dos usuários ou se deve ser ignorado. Aplicável somente para aplicativos Rails 3.	`false`	`true`
			
			`false`

Name (Nome)	Descrição	Padrão	Valores válidos
RAILS_SKIP_ASSET_COMPILATION	Especifica se o contêiner deve executar `rake assets:precompile` em nome dos aplicativos dos usuários ou se deve ser ignorado. Também aplicável somente para aplicativos Rails 3.	`false`	`true` `false`
BUNDLE_WITHOUT	Uma lista separada por dois pontos (`:`) de grupos a serem ignorados ao instalar dependências de um Gemfile.	`test:development`	n/a
RACK_ENV	Especifica em que estágio do ambiente um aplicativo pode ser executado. Exemplos de ambientes comuns incluem desenvolvimento, produção, teste.	`production`	n/a

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Opções personalizadas

Use o namespace `aws:elasticbeanstalk:customoption` para definir opções e valores que possam ser lidos em blocos `Resources` de outros arquivos de configuração. Use as opções personalizadas para coletar as configurações especificadas pelo usuário em um único arquivo de configuração.

Por exemplo, você pode ter um arquivo de configuração complexa que define um recurso que pode ser configurado pelo usuário que inicia o ambiente. Se você usar `Fn::GetOptionSetting` para recuperar o valor de uma opção personalizada, poderá colocar a definição dessa opção em um arquivo de configuração diferente, para que seja detectada e modificada com mais facilidade pelo usuário.

Além disso, como são opções de configuração, as opções personalizadas podem ser definidas no nível de API para substituir valores definidos em um arquivo de configuração. Para obter mais informações, consulte Precedência.

As opções personalizadas são definidas como qualquer outra opção:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    option name: option value
```

Por exemplo, o seguinte arquivo de configuração cria uma opção chamada `ELBAlarmEmail` e define o valor como `someone@example.com`:

```
option_settings:
  aws:elasticbeanstalk:customoption:
    ELBAlarmEmail: someone@example.com
```

Em outras páginas, um arquivo de configuração define um tópico SNS que lê a opção com `Fn::GetOptionSetting` para preencher o valor do atributo `Endpoint`:

```
Resources:
  MySNSTopic:
    Type: AWS::SNS::Topic
    Properties:
      Subscription:
        - Endpoint:
            Fn::GetOptionSetting:
              OptionName: ELBAlarmEmail
              DefaultValue: nobody@example.com
          Protocol: email
```

Encontre mais snippets demonstrativos usando `Fn::GetOptionSetting` em Adicionar e personalizar recursos do ambiente do Elastic Beanstalk.

Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`)

Você pode adicionar arquivos de AWS Elastic Beanstalk configuração (`.ebextensions`) ao código-fonte do seu aplicativo web para configurar seu ambiente e personalizar os AWS recursos que ele contém. Arquivos de configuração são documentos formatados em YAML ou JSON com uma extensão de arquivo `.config` que você coloca em uma pasta `.ebextensions` e implanta no pacote de origem da sua aplicação.

Example .ebextensions/ .config network-load-balancer

Este exemplo facilita uma alteração de configuração. Ele modifica uma opção de configuração para definir o tipo de balanceador de carga do ambiente como Network Load Balancer.

```
option_settings:
  aws:elasticbeanstalk:environment:
    LoadBalancerType: network
```

Recomendamos usar o YAML para seus arquivos de configuração, pois é mais legível do que o JSON. YAML é compatível com comentários, comandos de várias linhas, várias alternativas para usar aspas e muito mais. No entanto, é possível fazer qualquer alteração de configuração em arquivos de configuração do Elastic Beanstalk de forma idêntica usando YAML ou JSON.

> **Dica**
>
> Quando você estiver desenvolvendo ou testando novos arquivos de configuração, inicie um ambiente limpo executando o aplicativo padrão e implante nele. Arquivos de configuração formatados incorretamente causarão falha irrecuperável do início de um novo ambiente.

A seção `option_settings` de um arquivo de configuração define valores para as opções de configuração. As opções de configuração permitem que você configure seu ambiente do Elastic Beanstalk AWS , os recursos nele contidos e o software que executa seu aplicativo. Os arquivos de configuração são apenas uma das várias maneiras de definir as opções de configuração.

A `Resources`seção permite que você personalize ainda mais os recursos no ambiente do seu aplicativo e defina AWS recursos adicionais além da funcionalidade fornecida pelas opções de configuração. Você pode adicionar e configurar qualquer recurso suportado pelo AWS CloudFormation, que o Elastic Beanstalk usa para criar ambientes.

As outras seções de um arquivo de configuração (`packages`,`sources`,`files`,`users`,`groups`, `commands`,`container_commands`, `services`) permitem que você configure as EC2 instâncias que são executadas em seu ambiente. Sempre que um servidor é iniciado no ambiente, o Elastic Beanstalk executa as operações definidas nessas seções para preparar o sistema operacional e o sistema de armazenamento para a aplicação.

Para obter exemplos de .ebextensions usados com frequência, consulte os Arquivos de configuração do repositório do Elastic Beanstalk.

Requisitos

- Local: o Elastic Beanstalk processará todas as pastas `.ebextensions` presentes na sua implantação. No entanto, recomendamos que você coloque todos os seus arquivos de configuração em uma única pasta, denominada `.ebextensions`, na raiz do seu pacote de código-fonte. Pastas que começam com um ponto podem ser ocultas por navegadores de arquivos, portanto, certifique-se de que a pasta é adicionada quando você criar seu pacote de origem. Para obter mais informações, consulte Crie um pacote de origem da aplicação Elastic Beanstalk.
- Nomenclatura: os arquivos de configuração devem ter a extensão de arquivo `.config`.
- Formatação: os arquivos de configuração devem ser compatíveis com as especificações de YAML ou JSON.

  Ao usar o YAML, sempre use espaços para recuar as chaves em diferentes níveis de aninhamento. Para obter mais informações sobre YAML, consulte YAML Ain't Markup Language (YAML™) versão 1.1.
- Exclusividade: use cada chave somente uma vez em cada arquivo de configuração.

> **Aviso**
>
> Se você usar uma chave (por exemplo, `option_settings`) duas vezes no mesmo arquivo de configuração, uma das seções será descartada. Combine seções duplicadas em uma única seção ou coloque-as em arquivos de configuração separados.

O processo de implantação varia ligeiramente, dependendo do cliente que você usa para gerenciar seus ambientes. Para detalhes, consulte estas seções:

- Console do Elastic Beanstalk
- EB CLI
- AWS CLI

Tópicos

- Configurações de opção
- Personalizar software em servidores Linux
- Personalizar software em Windows Servers

- Adicionar e personalizar recursos do ambiente do Elastic Beanstalk

Configurações de opção

É possível usar a chave `option_settings` para modificar a configuração do Elastic Beanstalk e definir variáveis que podem ser recuperadas da aplicação usando variáveis de ambiente. Alguns namespaces permitem estender o número e especificar os nomes dos parâmetros. Para obter uma lista de namespaces e as opções de configuração, consulte Opções de configuração.

É possível também aplicar as configurações de opção diretamente a um ambiente durante a criação ou a atualização dele. As configurações aplicadas diretamente ao ambiente substituem aquelas referentes às mesmas opções nos arquivos de configuração. Se você remover as configurações de uma configuração do ambiente, as dos arquivos de configuração entrarão em vigor. Para mais detalhes, consulte Precedência.

Sintaxe

A sintaxe padrão para configurações de opção é uma matriz de objetos, cada um com uma chave `namespace`, `option_name` e `value`.

```
option_settings:
  - namespace:  namespace
    option_name:  option name
    value:  option value
  - namespace:  namespace
    option_name:  option name
    value:  option value
```

A chave `namespace` é opcional. Se você não especificar um namespace, o padrão usado será `aws:elasticbeanstalk:application:environment`:

```
option_settings:
  - option_name:  option name
    value:  option value
  - option_name:  option name
    value:  option value
```

O Elastic Beanstalk também é compatível com uma sintaxe abreviada para configurações de opção que permite especificar opções como pares de chave/valor sob o namespace:

```
option_settings:
  namespace:
    option name: option value
    option name: option value
```

Exemplos

Os exemplos a seguir definem uma opção específica da plataforma Tomcat no namespace `aws:elasticbeanstalk:container:tomcat:jvmoptions` e uma propriedade de ambiente chamada `MYPARAMETER`.

No formato YAML padrão:

Example .ebextensions/options.config

```
option_settings:
  - namespace:  aws:elasticbeanstalk:container:tomcat:jvmoptions
    option_name:  Xmx
    value:  256m
  - option_name: MYPARAMETER
    value: parametervalue
```

No formato abreviado:

Example .ebextensions/options.config

```
option_settings:
  aws:elasticbeanstalk:container:tomcat:jvmoptions:
    Xmx: 256m
  aws:elasticbeanstalk:application:environment:
    MYPARAMETER: parametervalue
```

Em JSON:

Example .ebextensions/options.config

```
{
  "option_settings": [
    {
      "namespace": "aws:elasticbeanstalk:container:tomcat:jvmoptions",
      "option_name": "Xmx",
      "value": "256m"
```

```
    },
    {
      "option_name": "MYPARAMETER",
      "value": "parametervalue"
    }
  ]
}
```

Personalizar software em servidores Linux

Esta seção descreve o tipo de informação que você pode incluir em um arquivo de configuração para personalizar o software em suas EC2 instâncias que executam Linux. Para obter informações gerais sobre como personalizar e configurar ambientes, consulte Configurar ambientes do Elastic Beanstalk. Para obter informações sobre a personalização do software em suas EC2 instâncias que executam o Windows, consultePersonalizar software em Windows Servers.

Você pode personalizar e configurar o software do qual seu aplicativo depende. É possível adicionar comandos a serem executados durante o provisionamento da instância, definir usuários e grupos do Linux e baixar ou criar arquivos diretamente nas instâncias do ambiente. Esses arquivos podem ser dependências exigidas pela aplicação, por exemplo, pacotes adicionais do repositório yum, ou podem ser arquivos de configuração, por exemplo, um arquivo de configuração de proxy para substituir configurações específicas usadas como padrão pelo Elastic Beanstalk.

> **Observações**
>
> - Em plataformas do Amazon Linux 2, em vez de fornecer arquivos e comandos em arquivos de configuração .ebextensions, é altamente recomendável usar Buildfile. Procfile, e hooks de plataforma sempre que possível para configurar e executar um código personalizado nas instâncias do ambiente durante o provisionamento da instância. Para obter detalhes sobre esses mecanismos, consulte the section called “Estender plataformas Linux”.
> - YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Os arquivos de configuração oferecem suporte às seguintes chaves que afetam o servidor Linux no qual seu aplicativo é executado.

Chaves

- Pacotes
- Grupos
- Usuários
- Origens
- Arquivos
- Comandos
- Serviços
- Comandos de contêiner
- Exemplo: uso de CloudWatch métricas personalizadas da Amazon

As chaves são processadas na ordem em que aparecem listadas aqui.

Assista a seus eventos de ambiente enquanto desenvolve e testa os arquivos de configuração. O Elastic Beanstalk ignora um arquivo de configuração que contém erros de validação, como uma chave inválida, e não processa nenhuma das outras chaves no mesmo arquivo. Quando isso acontece, o Elastic Beanstalk adiciona um evento de aviso ao log de eventos.

Pacotes

Use a chave `packages` para fazer download e instalar aplicativos e componentes predefinidos.

Sintaxe

```
packages:
  name of package manager:
    package name: version
    ...
  name of package manager:
    package name: version
    ...
  ...
```

Você pode especificar vários pacotes sob cada chave do gerenciador de pacotes.

Formatos de pacote com suporte

Atualmente, o Elastic Beanstalk é compatível com os seguintes gerenciadores de pacote: yum, rubygems, python e rpm. Os pacotes são processados na seguinte ordem: rpm, yum e, em seguida,

rubygems e python. Não há ordem entre rubygems e python. Dentro de cada gerenciador de pacotes, a ordem de instalação do pacote não é garantida. Use um gerenciador de pacotes compatível com o seu sistema operacional.

> **Note**
>
> O Elastic Beanstalk é compatível com dois gerenciadores de pacotes subjacentes para Python, pip e easy_install. No entanto, na sintaxe do arquivo de configuração, especifique o nome do gerenciador de pacotes como `python`. Quando você usa um arquivo de configuração para especificar um gerenciador de pacotes Python, o Elastic Beanstalk usa o Python 2.7. Se seu aplicativo conta com uma versão diferente do Python, especifique os pacotes a serem instalados em um arquivo `requirements.txt`. Para obter mais informações, consulte Especificação de dependências usando um arquivo de requisitos no Elastic Beanstalk.

Como especificar versões

Dentro de cada gerenciador de pacotes, cada pacote é especificado como um nome de pacote e uma lista de versões. A versão pode ser uma sequência, uma lista de versões ou uma sequência ou lista vazia. Uma sequência ou lista vazia indica que você deseja a versão mais recente. Para o gerenciador de rpm, a versão é especificada como um caminho para um arquivo no disco ou um URL. Não há suporte para caminhos relativos.

Se você especificar uma versão de um pacote, o Elastic Beanstalk tentará instalar essa versão, mesmo que uma versão mais recente do pacote já esteja instalada na instância. Se uma versão mais recente já estiver instalada, a implantação não terá êxito. Alguns gerenciadores de pacotes oferecem suporte a várias versões, mas outros não. Verifique a documentação do seu gerenciador de pacotes para obter mais informações. Se você não especificar uma versão e uma versão do pacote já estiver instalada, o Elastic Beanstalk não instalará uma nova versão, pois vai considerar que você deseja manter e usar a versão existente.

Trecho de exemplo

O trecho a seguir especifica um URL da versão para rpm, solicita a versão mais recente do yum e a versão 0.10.2 de chef do 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'
```

Grupos

Você pode usar a `groups` chave para criar grupos Linux/UNIX e atribuir grupos. IDs Para criar um grupo, adicione um novo par de chave/valor que mapeie um novo nome de grupo para um ID de grupo opcional. A chave de grupos pode conter um ou mais nomes de grupos. A tabela a seguir lista as chaves disponíveis.

Sintaxe

```
groups:
  name of group: {}
  name of group:
    gid: "group id"
```

Opções

`gid`

Um número de ID de grupo.

Se um ID de grupo for especificado e o grupo já existir por nome, a criação do grupo não terá êxito. Se outro grupo tiver o ID de grupo especificado, o sistema operacional poderá rejeitar a respectiva criação.

Trecho de exemplo

O trecho a seguir especifica um grupo chamado groupOne, sem atribuir um ID de grupo, e um grupo chamado groupTwo que especificou um valor de ID de grupo 45.

```
groups:
  groupOne: {}
  groupTwo:
    gid: "45"
```

Usuários

Você pode usar a `users` chave para criar usuários Linux/UNIX na instância. EC2

Sintaxe

```
users:
  name of user:
    groups:
      - name of group
    uid: "id of the user"
    homeDir: "user's home directory"
```

Opções

`uid`

Um ID de usuário. Haverá falha no processo de criação se o nome do usuário existir com outro ID de usuário. Se o ID de usuário já estiver atribuído a um usuário existente, o sistema operacional poderá rejeitar a solicitação de criação.

`groups`

Uma lista de nomes de grupos. O usuário é adicionado a cada grupo da lista.

`homeDir`

O diretório base de usuário.

Os usuários são criados como usuários de sistema não interativos com um shell de `/sbin/nologin`. Este é o design e não pode ser modificado.

Trecho de exemplo

```
users:
  myuser:
    groups:
      - group1
      - group2
    uid: "50"
    homeDir: "/tmp"
```

Origens

Você pode usar a `sources` chave para baixar um arquivo de um URL público e descompactá-lo em um diretório de destino na EC2 instância.

Sintaxe

```
sources:
  target directory: location of archive file
```

Formatos com compatibilidade

Os formatos com suporte são tar, tar+gzip, tar+bz2 e zip. Você pode fazer referência a locais externos, como o Amazon Simple Storage Service (Amazon S3) (por exemplo, `https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject`), desde que o URL seja publicamente acessível.

Trecho de exemplo

O exemplo a seguir faz download de um arquivo .zip público de um bucket do Amazon S3 e o descompacta em `/etc/myapp`:

```
sources:
  /etc/myapp: https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject
```

> **Note**
>
> Não reutilize o mesmo caminho de destino para várias extrações. A extração de outra origem para o mesmo caminho de destino substituirá em vez de anexar aos conteúdos.

Arquivos

Você pode usar a `files` chave para criar arquivos na EC2 instância. O conteúdo pode estar embutido no arquivo de configuração ou o conteúdo pode ser obtido de um URL. Os arquivos são gravados em disco em ordem lexicográfica.

É possível usar a chave `files` para fazer download de arquivos privados do Amazon S3 ao fornecer o perfil da instância para autorização.

Se o caminho do arquivo especificado já existir na instância, o arquivo existente será mantido com a extensão `.bak` anexada ao nome.

Sintaxe

```
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:
```

Opções

`content`

Conteúdo de sequência para adicionar ao arquivo. Especifique `content` ou `source`, mas não ambos.

`source`

URL de um arquivo para fazer download: Especifique `content` ou `source`, mas não ambos.

`encoding`

O formato de codificação da sequência especificada com a opção `content`.

Valores válidos: `plain` | `base64`

`group`

Grupo do Linux que possui o arquivo.

`owner`

Usuário do Linux que possui o arquivo.

`mode`

Um valor octal de seis dígitos que representa o modo para este arquivo. Sem suporte para sistemas Windows. Use os primeiros três dígitos para symlinks e os três últimos dígitos para configurar permissões. Para criar um symlink, especifique `120xxx`, em que `xxx` define as permissões do arquivo de destino. Para especificar permissões para um arquivo, use os últimos três dígitos, como `000644`.

`authentication`

O nome de um método de autenticação do AWS CloudFormation para usar. É possível adicionar métodos de autenticação aos metadados do grupo de Auto Scaling com a chave Resources. Veja um exemplo a seguir.

Trecho de exemplo

```
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
```

Exemplo usando um symlink. Isso cria um link `/tmp/myfile2.txt` que aponta para o arquivo existente `/tmp/myfile1.txt`.

```
files:
  "/tmp/myfile2.txt" :
    mode: "120400"
    content: "/tmp/myfile1.txt"
```

O exemplo a seguir usa a chave Resources para adicionar um método de autenticação chamado S3Auth e o usa para fazer download de um arquivo privado de um bucket do 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
```

Comandos

Você pode usar a `commands` chave para executar comandos na EC2 instância. Os comandos executados antes da configuração do aplicativo e do servidor web e da extração do arquivo de versão do aplicativo.

Os comandos especificados são executadas como o usuário raiz e processados em ordem alfabética por nome. Por padrão, os comandos são executados no diretório raiz. Para executar comandos de outro diretório, use a opção `cwd`.

Para solucionar problemas com seus comandos, você pode encontrar a saída deles nos logs da instância.

Sintaxe

```
commands:
  command name:
    command: command to run
    cwd: working directory
    env:
```

```
    variable name: variable value
  test: conditions for command
  ignoreErrors: true
```

Opções

`command`

Uma matriz (coleção de sequência em bloco em sintaxe de YAML) ou uma cadeia que especifica o comando a ser executado. Algumas observações importantes:

- Se você usar uma cadeia, você não precisa colocar a cadeia inteira entre aspas. Se você usar aspas, pule as ocorrências literais do mesmo tipo de citação.
- Se você usar uma matriz, não precisará de caracteres de espaço de escape ou de aspas para parâmetros de comandos. Cada elemento de matriz é um único argumento de comando. Não use uma matriz para especificar vários comandos.

Os exemplos a seguir são todos equivalentes:

```
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.
```

Para especificar múltiplos comandos, use um escalar de bloco literal, como exibido no exemplo a seguir.

```
commands:
  command block:
    command: |
      git commit -m "This is a comment."
      git push
```

`env`

(Opcional) Define variáveis de ambiente para o comando. Essa propriedade substitui, em vez de anexar, o ambiente existente.

`cwd`

(Opcional) O diretório de trabalho. Se não for especificado, os comandos serão executados a partir do diretório raiz (/).

`test`

(Opcional) Um comando que deve retornar o valor `true` (código de saída 0) para que o Elastic Beanstalk processe o comando, por exemplo, um script de shell, contido na chave `command`.

`ignoreErrors`

(Opcional) Um valor booliano que determina se outros comandos deverão ser executados se houver falha no comando incluído na chave `command` (retorna um valor diferente de zero). Defina esse valor como `true` para continuar executando os comandos mesmo se houver falha no comando. Defina-o como `false` para interromper a execução dos comandos em caso de falha no comando. O valor padrão é `false`.

Trecho de exemplo

O exemplo a seguir executa um trecho de script do Python.

```
commands:
  python_install:
    command: myscript.py
    cwd: /home/ec2-user
    env:
      myvarname: myvarvalue
    test: "[ -x /usr/bin/python ]"
```

Serviços

É possível usar a chave `services` para definir quais serviços deverão ser iniciados ou interrompidos quando a instância for executada. A chave `services` também permite especificar dependências de origens, pacotes e arquivos para que, se for necessária uma reinicialização devido à instalação de arquivos, o Elastic Beanstalk reinicie o serviço.

Sintaxe

```
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"
```

Opções

`ensureRunning`

Defina `true` para garantir que o serviço esteja em execução depois que o Elastic Beanstalk for concluído.

Defina `false` para garantir que o serviço não esteja em execução depois que o Elastic Beanstalk for concluído.

Omita essa chave para não fazer nenhuma alteração no estado do serviço.

`enabled`

Defina como `true` para garantir que o serviço seja iniciado automaticamente na inicialização.

Defina como `false` para garantir que o serviço não seja iniciado automaticamente na inicialização.

Omita essa chave para não fazer nenhuma alteração nessa propriedade.

`files`

Uma lista de arquivos. Se o Elastic Beanstalk alterar um diretamente por meio do bloco de arquivos, o serviço será reiniciado.

`sources`

Uma lista de diretórios. Se o Elastic Beanstalk expandir um arquivo em um desses diretórios, o serviço será reiniciado.

`packages`

Um mapa do gerenciador de pacotes para uma lista de nomes de pacotes. Se o Elastic Beanstalk instalar ou atualizar um desses pacotes, o serviço será reiniciado.

`commands`

Uma lista de nomes de comandos. Se o Elastic Beanstalk executar o comando especificado, o serviço será reiniciado.

Trecho de exemplo

Veja, a seguir, um trecho de exemplo:

```
services:
  sysvinit:
    myservice:
      enabled: true
      ensureRunning: true
```

Comandos de contêiner

É possível usar a chave `container_commands` para executar comandos que afetam o código-fonte do seu aplicativo. Os comandos de contêiner são executados depois que o aplicativo e o servidor Web são configurados e o arquivo de versão do aplicativo é extraído, mas antes da implantação da versão do aplicativo. Os comandos que não são de contêiner e outras operações de personalização são executados antes da extração do código-fonte do aplicativo.

Os comandos especificados são executadas como o usuário raiz e processados em ordem alfabética por nome. Os comandos de contêiner são executados no diretório de preparação, onde seu código-fonte é extraído antes de ser implantado no servidor de aplicativo. Quaisquer alterações que você fizer em seu código-fonte no diretório de preparação com um comando de contêiner serão incluídas quando a fonte for implantada em seu local final.

> **Note**
>
> A saída dos comandos de contêiner é registrada no log da instância `cfn-init-cmd.log`. Para obter mais informações sobre como recuperar e visualizar registros de instâncias, consulte Visualização de registros de EC2 instâncias da Amazon.

É possível usar `leader_only` para somente executar o comando em uma única instância ou configurar um `test` para somente executar o comando quando um comando de teste for avaliado como `true`. Os comandos de contêiner somente líderes são executados apenas durante a criação do ambiente e as implantações, enquanto outros comandos e operações de personalização de servidor são executados sempre que uma instância é provisionada ou atualizada. Os comandos de contêiner somente líderes não são executados devido a alterações na configuração de execução, como uma alteração no ID do AMI ou no tipo de instância.

Sintaxe

```
container_commands:
  name of container_command:
    command: "command to run"
    leader_only: true
  name of container_command:
    command: "command to run"
```

Opções

`command`

Uma string ou matriz de strings para execução.

`env`

(Opcional) Defina variáveis de ambiente antes de executar o comando, substituindo qualquer valor existente.

`cwd`

(Opcional) O diretório de trabalho. Por padrão, este é o diretório de preparação do aplicativo descompactado.

`leader_only`

(Opcional) Somente execute o comando em uma única instância escolhida pelo Elastic Beanstalk. Os comandos de contêiner somente líderes são executados antes dos outros comandos de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

`test`

(Opcional) Execute um comando de teste que deve retornar `true` para executar o comando de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

`ignoreErrors`

(Opcional) Não falhar implantações se o comando de contêiner retornar um valor diferente de 0 (êxito). Defina como `true` para habilitar.

Trecho de exemplo

Veja, a seguir, um trecho de exemplo.

```
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"
```

Exemplo: uso de CloudWatch métricas personalizadas da Amazon

Este tópico fornece um exemplo de configuração que integra as métricas do Elastic Beanstalk CloudWatch com o agente da Amazon para plataformas baseadas no Amazon Linux 2 e versões posteriores. O exemplo de configuração usa arquivos e comandos em um arquivo de configuração `.ebextensions`.

CloudWatch A Amazon é um serviço web que permite monitorar, gerenciar e publicar várias métricas, bem como configurar ações de alarme com base nos dados das métricas. Você pode definir métricas personalizadas para seu próprio uso, e o Elastic Beanstalk enviará essas métricas para a Amazon. CloudWatch Depois que a Amazon CloudWatch contiver suas métricas personalizadas, você poderá visualizá-las no CloudWatch console da Amazon.

O CloudWatch agente da Amazon

O CloudWatch agente da Amazon permite a coleta de CloudWatch métricas e registros de EC2 instâncias da Amazon e servidores locais em todos os sistemas operacionais. O agente oferece suporte a métricas coletadas no nível do sistema. Ele também oferece suporte para a coleta de métricas personalizadas de aplicações ou serviços. Para obter mais informações sobre o CloudWatch agente da Amazon, consulte Coleta de métricas e registros com o CloudWatch agente no Guia CloudWatch do usuário da Amazon.

> **Note**
>
> O Elastic Beanstalk Enhanced Health Reporting tem suporte nativo para publicar uma ampla variedade de métricas de instância e ambiente no. CloudWatch Para mais detalhes, consulte Publicação de métricas CloudWatch personalizadas da Amazon para um ambiente.

Tópicos

- Arquivo de configuração .Ebextensions
- Permissões
- Visualizando métricas no CloudWatch console

Arquivo de configuração .Ebextensions

Este exemplo usa arquivos e comandos em um arquivo de configuração.ebextensions para configurar e executar o CloudWatch agente da Amazon na plataforma Amazon Linux 2. O agente faz parte do pacote do Amazon Linux 2. Se você estiver usando um sistema operacional diferente, etapas adicionais para instalar o agente podem ser necessárias. Para obter mais informações, consulte Instalando o CloudWatch agente no Guia CloudWatch do usuário da Amazon.

Para usar esse exemplo, salve-o em um arquivo chamado `cloudwatch.config` em um diretório chamado `.ebextensions` no nível superior do diretório do projeto e implante a aplicação usando o

console do Elastic Beanstalk (inclua o diretório .ebextensions em seu pacote de origem) ou na CLI do EB.

Para obter mais informações sobre esses arquivos de configuração, consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`).

Este arquivo tem duas seções:

- `files`: essa seção adiciona o arquivo de configuração do agente. Ele indica quais métricas e registros o agente deve enviar para a Amazon CloudWatch. Neste exemplo, estamos enviando apenas a métrica mem_used_percent. Para obter uma lista completa das métricas em nível de sistema suportadas pelo CloudWatch agente da Amazon, consulte Métricas coletadas pelo CloudWatch agente no Guia CloudWatch do usuário da Amazon.
- `container_commands`: essa seção contém o comando que inicia o agente, passando o arquivo de configuração como um parâmetro. Para obter mais detalhes sobre o `container_commands`, consulte Comandos de contêiner.

.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
```

Permissões

As instâncias em seu ambiente precisam das permissões adequadas do IAM para publicar CloudWatch métricas personalizadas da Amazon usando o CloudWatch agente da Amazon. Você pode conceder permissões às instâncias do ambiente adicionando-as ao perfil da instância do ambiente. Você pode adicionar permissões ao perfil da instância antes ou depois da implantação de seu aplicativo.

Para conceder permissões para publicar CloudWatch métricas

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Perfis.
3. Escolha a função de perfil da instância de seu ambiente. Por padrão, ao criar um ambiente com o console do Elastic Beanstalk ou com a CLI do EB, ela é `aws-elasticbeanstalk-ec2-role`.
4. Escolha a aba Permissões.
5. Em Permissions Policies (Políticas de permissões), na seção Permissions (Permissões) escolha Attach policies (Anexar políticas).
6. Em Anexar permissões, escolha a política AWS gerenciada CloudWatchAgentServerPolicy. Clique em Attach Policy (Anexar política).

Para obter mais informações sobre o gerenciamento de políticas, consulte Trabalhar com políticas no Guia do usuário do IAM.

Visualizando métricas no CloudWatch console

Depois de implantar o arquivo de CloudWatch configuração em seu ambiente, verifique o CloudWatchconsole da Amazon para ver suas métricas. As métricas personalizadas estarão localizadas no CWAgentnamespace.

Para obter mais informações, consulte Visualização das métricas disponíveis no Guia CloudWatch do usuário da Amazon.

Personalizar software em Windows Servers

Você pode personalizar e configurar o software do qual seu aplicativo depende. Estes arquivos podem ser dependências exigidas pelo aplicativo, por exemplo, pacotes adicionais ou serviços que precisam ser executados. Para obter informações gerais sobre como personalizar e configurar ambientes do Elastic Beanstalk, consulte Configurar ambientes do Elastic Beanstalk.

> **Note**
>
> YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Os arquivos de configuração oferecem suporte às seguintes chaves que afetam o servidor Windows no qual seu aplicativo é executado.

Chaves

- Pacotes
- Origens
- Arquivos
- Comandos
- Serviços
- Comandos de contêiner

As chaves são processadas na ordem em que aparecem listadas aqui.

> **Note**
>
> Versões mais antigas (sem versionamento) da plataforma .NET não processam arquivos de configuração na ordem correta. Saiba mais em Migrar entre versões principais da plataforma do Windows Server do Elastic Beanstalk.

Assista a seus eventos de ambiente enquanto desenvolve e testa os arquivos de configuração. O Elastic Beanstalk ignora um arquivo de configuração que contém erros de validação, como uma chave inválida, e não processa nenhuma das outras chaves no mesmo arquivo. Quando isso acontece, o Elastic Beanstalk adiciona um evento de aviso ao log de eventos.

Pacotes

Use a chave `packages` para fazer download e instalar aplicativos e componentes predefinidos.

Em ambientes Windows, o Elastic Beanstalk é compatível com o download e a instalação de pacotes MSI. (Os ambientes do Linux são compatíveis com outros gerenciadores de pacotes. Para obter detalhes, consulte Pacotes na página Personalizar software em servidores Linux).

É possível fazer referência a locais externos, como o Amazon Simple Storage Service (Amazon S3), desde que o URL seja publicamente acessível.

Se você especificar vários pacotes `msi:`, sua ordem de instalação não será garantida.

Sintaxe

Especifique um nome de sua preferência para o pacote e, como valor, um URL para um arquivo MSI local. Você pode especificar vários pacotes sob a chave `msi:`.

```
packages:
  msi:
    package name: package url
    ...
```

Exemplos

O exemplo a seguir especifica um URL para fazer download do mysql em `https://dev.mysql.com/`.

```
packages:
  msi:
    mysql: https://dev.mysql.com/get/Downloads/Connector-Net/mysql-connector-net-8.0.11.msi
```

O exemplo a seguir especifica um objeto do Amazon S3 como o local do arquivo MSI.

```
packages:
  msi:
```

```
    mymsi: https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject.msi
```

Origens

Use a `sources` chave para baixar um arquivo de um URL público e descompactá-lo em um diretório de destino na EC2 instância.

Sintaxe

```
sources:
  target directory: location of archive file
```

Formatos com suporte

Em ambientes Windows, o Elastic Beanstalk é compatível com o formato .zip. (Os ambientes do Linux são compatíveis com outros formatos. Para obter detalhes, consulte Origens na página Personalizar software em servidores Linux.)

É possível fazer referência a locais externos, como o Amazon Simple Storage Service (Amazon S3), desde que o URL seja publicamente acessível.

Exemplo

O exemplo a seguir baixa um arquivo .zip público do Amazon S3 e o descompacta no `c:/myproject/myapp`.

```
sources:
  "c:/myproject/myapp": https://amzn-s3-demo-bucket.s3.amazonaws.com/myobject.zip
```

Arquivos

Use a `files` chave para criar arquivos na EC2 instância. O conteúdo pode ser incorporado no arquivo de configuração ou obtido de um URL. Os arquivos são gravados em disco em ordem lexicográfica. Para fazer download de arquivos privados do Amazon S3, forneça o perfil da instância para autorização.

Sintaxe

```
files:
  "target file location on disk":
    source: URL
```

```
    authentication: authentication name:

  "target file location on disk":
    content: |
      this is my content
    encoding: encoding format
```

Opções

`content`

(Opcional) Uma string.

`source`

(Opcional) O URL de onde o arquivo é carregado. Essa opção não pode ser especificada com a chave de conteúdo.

`encoding`

(Opcional) O formato de codificação. Essa opção só é usada para um valor de chave de conteúdo fornecido. O valor padrão é `plain`.

Valores válidos: `plain` | `base64`

`authentication`

(Opcional) O nome de um método de autenticação do AWS CloudFormation para usar. É possível adicionar métodos de autenticação aos metadados do grupo de Auto Scaling com a chave Resources.

Exemplos

O exemplo a seguir mostra as duas maneiras de fornecer conteúdo do arquivo: por meio de um URL ou em linha no arquivo de configuração.

```
files:
  "c:\\targetdirectory\\targetfile.txt":
    source: http://foo.bar/myfile

  "c:/targetdirectory/targetfile.txt":
    content: |
      # this is my file
      # with content
```

> **Note**
>
> Se você usar uma barra invertida (\) no caminho do arquivo, deverá precedê-la com outra barra invertida (caractere de escape), como mostrado no exemplo anterior.

O exemplo a seguir usa a chave de recursos para adicionar um método de autenticação chamado S3Auth e o utiliza para baixar um arquivo privado de um 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"
```

Comandos

Use a `commands` chave para executar comandos na EC2 instância. Os comandos são processados em ordem alfabética por nome e executados antes que o aplicativo e o servidor Web sejam configurados e o arquivo de versão do aplicativo seja extraído.

Os comandos especificados são executados como usuário administrador.

Para solucionar problemas com seus comandos, você pode encontrar a saída deles nos logs da instância.

Sintaxe

```
commands:
```

```
  command name:
    command: command to run
```

Opções

`command`

Uma matriz ou uma string que especifica o comando a ser executado. Se você usar uma matriz, não precisará de caracteres de espaço de escape ou de aspas para parâmetros de comandos.

`cwd`

(Opcional) O diretório de trabalho. Por padrão, o Elastic Beanstalk tenta encontrar o local do diretório do projeto. Se não for encontrado, ele usará `c:\Windows\System32` como padrão.

`env`

(Opcional) Define variáveis de ambiente para o comando. Essa propriedade substitui, em vez de anexar, o ambiente existente.

`ignoreErrors`

(Opcional) Um valor booliano que determina se outros comandos deverão ser executados se houver falha no comando incluído na chave `command` (retorna um valor diferente de zero). Defina esse valor como `true` para continuar executando os comandos mesmo se houver falha no comando. Defina-o como `false` para interromper a execução dos comandos em caso de falha no comando. O valor padrão é `false`.

`test`

(Opcional) Um comando que deve retornar o valor `true` (código de saída 0) para que o Elastic Beanstalk processe o comando incluído na chave `command`.

`waitAfterCompletion`

(Opcional) Quantos segundos aguardar para executar o próximo comando depois que o comando é concluído. O valor padrão é **`60`** segundos. Você também pode especificar**`forever`**.

> **Important**
>
> Não há suporte para reinicializações do sistema durante a implantação. Se o sistema for reinicializado como resultado de um comando, a inicialização da instância falhará, fazendo com que a implantação falhe.

> Como solução alternativa, você pode usar isso `.ebextensions configuration`para agendar reinicializações após a conclusão da implantação.

Exemplo

O exemplo a seguir salva a saída do comando `set` no arquivo especificado. Se houver um comando subsequente, o Elastic Beanstalk o executará logo após o término deste comando. Se esse comando exigir reinicialização, o Elastic Beanstalk reinicializará a instância logo após o término do comando.

```
commands:
  test:
    command: set > c:\\myapp\\set.txt
    waitAfterCompletion: 0
```

Serviços

Use a chave `services` para definir quais serviços deverão ser iniciados ou interrompidos quando a instância for executada. A chave `services` também permite especificar dependências em fontes, pacotes e arquivos para que, se for necessária uma reinicialização devido à instalação de arquivos, o Elastic Beanstalk reinicie o serviço.

Sintaxe

```
services:
  windows:
    name of service:
      files:
        - "file name"
      sources:
        - "directory"
      packages:
        name of package manager:
            "package name[: version]"
      commands:
        - "name of command"
```

Opções

`ensureRunning`

(Opcional) Defina `true` para garantir que o serviço esteja sendo executado após o Elastic Beanstalk ser concluído.

Defina `false` para garantir que o serviço não esteja em execução depois que o Elastic Beanstalk for concluído.

Omita essa chave para não fazer nenhuma alteração no estado do serviço.

`enabled`

(Opcional) Defina como `true` para garantir que o serviço seja iniciado automaticamente na inicialização.

Defina como `false` para garantir que o serviço não seja iniciado automaticamente na inicialização.

Omita essa chave para não fazer nenhuma alteração nessa propriedade.

`files`

Uma lista de arquivos. Se o Elastic Beanstalk alterar um diretamente por meio do bloco de arquivos, o serviço será reiniciado.

`sources`

Uma lista de diretórios. Se o Elastic Beanstalk expandir um arquivo em um desses diretórios, o serviço será reiniciado.

`packages`

Um mapa do gerenciador de pacotes para uma lista de nomes de pacotes. Se o Elastic Beanstalk instalar ou atualizar um desses pacotes, o serviço será reiniciado.

`commands`

Uma lista de nomes de comandos. Se o Elastic Beanstalk executar o comando especificado, o serviço será reiniciado.

Exemplo

```
services:
```

```
  windows:
    myservice:
      enabled: true
      ensureRunning: true
```

Comandos de contêiner

Use a chave `container_commands` para executar comandos que afetam o código-fonte do seu aplicativo. Os comandos de contêiner são executados depois que o aplicativo e o servidor Web são configurados e o arquivo de versão do aplicativo é extraído, mas antes da implantação da versão do aplicativo. Os comandos que não são de contêiner e outras operações de personalização são executados antes da extração do código-fonte do aplicativo.

Os comandos de contêiner são executados no diretório de preparação, onde seu código-fonte é extraído antes de ser implantado no servidor de aplicativo. Quaisquer alterações que você fizer em seu código-fonte no diretório de preparação com um comando de contêiner serão incluídas quando a fonte for implantada em seu local final.

Para solucionar problemas com seus comandos de contêiner, você pode encontrar a saída deles nos logs da instância.

Use a opção `leader_only` para somente executar o comando em uma única instância ou configure `test` para somente executar o comando quando um comando de teste é avaliado como `true`. Os comandos de contêiner somente líderes são executados apenas durante a criação do ambiente e as implantações, enquanto outros comandos e operações de personalização de servidor são executados sempre que uma instância é provisionada ou atualizada. Os comandos de contêiner somente líderes não são executados devido a alterações na configuração de execução, como uma alteração no ID do AMI ou no tipo de instância.

Sintaxe

```
container_commands:
  name of container_command:
    command: command to run
```

Opções

`command`

Uma string ou matriz de strings para execução.

`env`

(Opcional) Defina variáveis de ambiente antes de executar o comando, substituindo qualquer valor existente.

`cwd`

(Opcional) O diretório de trabalho. Por padrão, este é o diretório de preparação do aplicativo descompactado.

`leader_only`

(Opcional) Somente execute o comando em uma única instância escolhida pelo Elastic Beanstalk. Os comandos de contêiner somente líderes são executados antes dos outros comandos de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

`test`

(Opcional) Execute um comando de teste que deve retornar `true` para executar o comando de contêiner. Um comando pode ser somente líder ou ter um `test`, mas não as duas coisas (`leader_only` tem precedência).

`ignoreErrors`

(Opcional) Não falhar implantações se o comando de contêiner retornar um valor diferente de 0 (êxito). Defina como `true` para habilitar.

`waitAfterCompletion`

(Opcional) Quantos segundos aguardar para executar o próximo comando depois que o comando é concluído. O valor padrão é **`60`** segundos. Você também pode especificar**`forever`**.

> **Important**
>
> Não há suporte para reinicializações do sistema durante a implantação. Se o sistema for reinicializado como resultado de um comando, a inicialização da instância falhará, fazendo com que a implantação falhe.
> Como solução alternativa, você pode usar isso `.ebextensions configuration`para agendar reinicializações após a conclusão da implantação.

Exemplo

O exemplo a seguir salva a saída do comando `set` no arquivo especificado. O Elastic Beanstalk executa o comando em uma instância e reinicializa a instância logo após o término do comando.

```
container_commands:
  foo:
    command: set > c:\\myapp\\set.txt
    leader_only: true
    waitAfterCompletion: 0
```

Adicionar e personalizar recursos do ambiente do Elastic Beanstalk

Você também pode personalizar seus recursos de ambiente que são parte do ambiente do Elastic Beanstalk. Por exemplo, você pode querer adicionar uma fila do Amazon SQS e um alarme sobre a profundidade da fila, ou você pode querer adicionar um cluster da Amazon. ElastiCache Você pode personalizar facilmente seu ambiente ao mesmo tempo que implanta a versão do aplicativo incluindo um arquivo de configuração com o pacote de origem.

Você pode usar a `Resources` chave em um arquivo de configuração para criar e personalizar AWS recursos em seu ambiente. Os recursos definidos nos arquivos de configuração são adicionados ao AWS CloudFormation modelo usado para iniciar seu ambiente. Todos os tipos de AWS CloudFormation recursos são compatíveis.

> **Note**
>
> Sempre que você adicionar um recurso que não seja gerenciado pelo Elastic Beanstalk, não se esqueça de adicionar uma política de usuário com as permissões AWS Identity and Access Management apropriadas para seus usuários (IAM). As políticas de usuário gerenciadas que o Elastic Beanstalk fornece apenas permissões de cobertura para recursos gerenciados pelo Elastic Beanstalk.

Por exemplo, o arquivo de configuração a seguir adiciona um gancho de ciclo de vida do Auto Scaling ao grupo de Auto Scaling padrão criado pelo 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"] }
```

Este exemplo define três recursos: `hookrole`, `hooktopic` e `lifecyclehook`. Os dois primeiros recursos são uma função do IAM, que concede permissão ao Amazon EC2 Auto Scaling para publicar mensagens no Amazon SNS, e um tópico do SNS, que retransmite mensagens do grupo Auto Scaling para um endereço de e-mail. O Elastic Beanstalk cria esses recursos com as propriedades e os tipos especificados.

O recurso final, `lifecyclehook`, é o ciclo de vida do gancho propriamente dito:

```
lifecyclehook:
  Type: AWS::AutoScaling::LifecycleHook
  Properties:
    AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
    LifecycleTransition: autoscaling:EC2_INSTANCE_TERMINATING
    NotificationTargetARN: { "Ref" : "hooktopic" }
    RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

A definição de gancho do ciclo de vida usa duas funções para preencher valores para as propriedades do gancho. `{ "Ref" : "AWSEBAutoScalingGroup" }` recupera o nome do grupo de Auto Scaling criado pelo Elastic Beanstalk para o ambiente. `AWSEBAutoScalingGroup` é um dos nomes de recurso padrão fornecidos pelo Elastic Beanstalk.

Para `AWS::IAM::Role`, `Ref` retorna somente o nome da função, não o nome de região da Amazon (ARN). Para obter o ARN para o parâmetro `RoleARN`, use outra função intrínseca, `Fn::GetAtt`, que pode obter qualquer atributo de um recurso. `RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }` obtém o atributo `Arn` do recurso `hookrole`.

`{ "Ref" : "hooktopic" }` obtém o ARN do tópico do Amazon SNS criado anteriormente no arquivo de configuração. O valor retornado por `Ref` varia de acordo com o tipo de recurso e pode ser encontrado no tópico AWS CloudFormation do Guia do Usuário para o tipo de AWS::SNS::Topic recurso.

Modificar os recursos criados pelo Elastic Beanstalk para o seu ambiente

Os recursos que o Elastic Beanstalk cria para o seu ambiente têm nomes. Você pode usar esses nomes para obter informações sobre os recursos com uma funçãoou modificar as propriedades nos recursos para personalizar seu comportamento. Este tópico descreve os AWS recursos que o Elastic Beanstalk usa nos diferentes tipos de ambientes.

> **Note**
>
> O tópico anterior, Recursos personalizados, fornece alguns casos de uso e exemplos para personalizar recursos de ambientes. Você também pode encontrar mais exemplos de arquivos de configuração no tópico posterior: Exemplos de recursos personalizados.

Os ambientes de servidor da web tem os recursos a seguir.

Ambientes de servidor da web

- `AWSEBAutoScalingGroup`(AWS::AutoScaling::AutoScalingGrupo) — O grupo Auto Scaling anexado ao seu ambiente.
- Um dos dois recursos a seguir.
  - `AWSEBAutoScalingLaunchConfiguration`(AWS::AutoScaling::LaunchConfiguration) — A configuração de lançamento anexada ao grupo Auto Scaling do seu ambiente.
  - `AWSEBEC2LaunchTemplate`(AWS::EC2::LaunchTemplate) — O modelo de EC2 lançamento da Amazon usado pelo grupo de Auto Scaling do seu ambiente.

  > **Note**
  >
  > Se seu ambiente usa funcionalidades que exigem modelos de EC2 lançamento da Amazon e sua política de usuário não tem as permissões necessárias, a criação ou atualização do ambiente pode falhar. Use a política de usuário gerenciada AdministratorAccess- AWSElastic Beanstalk ou adicione as permissões necessárias à sua política personalizada.

- `AWSEBEnvironmentName`(AWS::ElasticBeanstalk::Environment) — Seu ambiente.
- `AWSEBSecurityGroup`(AWS::EC2::SecurityGroup) — O grupo de segurança anexado ao seu grupo de Auto Scaling.
- `AWSEBRDSDatabase`(AWS: :RDS:: DBInstance) — A instância de banco de dados Amazon RDS conectada ao seu ambiente (se aplicável).

Em um ambiente com balanceamento de carga, é possível acessar recursos adicionais relacionados ao load balancer. Os load balancers clássicos têm um recurso para o load balancer e outro para

o grupo de segurança anexado a ele. Os load balancers de rede e de aplicativos têm recursos adicionais para listener padrão do load balancer para a regra do listener e para o grupo de destino.

Ambientes de carga equilibrada

- `AWSEBLoadBalancer`(AWS::ElasticLoadBalancing::LoadBalancer) — O balanceador de carga clássico do seu ambiente.
- `AWSEBV2LoadBalancer`(AWS::ElasticLoadBalancingV2::LoadBalancer) — O balanceador de carga do aplicativo ou da rede do seu ambiente.
- `AWSEBLoadBalancerSecurityGroup`(AWS::EC2::SecurityGroup) — Somente em uma Amazon Virtual Private Cloud personalizada (Amazon VPC), o nome do grupo de segurança que o Elastic Beanstalk cria para o balanceador de carga. Em uma VPC padrão ou EC2 clássica, o Elastic Load Balancing atribui um grupo de segurança padrão ao balanceador de carga.
- `AWSEBV2LoadBalancerListener`(AWS::ElasticLoadBalancingV2::Listener) — Um ouvinte que permite que o balanceador de carga verifique as solicitações de conexão e as encaminhe para um ou mais grupos-alvo.
- `AWSEBV2LoadBalancerListenerRule`(AWS::ElasticLoadBalancingV2::ListenerRule) — Define em quais solicitações um ouvinte do Elastic Load Balancing age e a ação que ele executa.
- `AWSEBV2LoadBalancerTargetGroup`(AWS::ElasticLoadBalancingV2::TargetGroup) — Um grupo-alvo do Elastic Load Balancing que encaminha solicitações para um ou mais destinos registrados, como instâncias da Amazon EC2 .

Ambientes de operador têm recursos para a fila do SQS que armazena as solicitações de entrada, e uma tabela do Amazon DynamoDB que as instâncias usam para escolher o líder.

Ambientes de operador

- `AWSEBWorkerQueue`(AWS::SQS::Queue) — A fila do Amazon SQS da qual o daemon extrai solicitações que precisam ser processadas.
- `AWSEBWorkerDeadLetterQueue`(AWS::SQS::Queue) — A fila do Amazon SQS que armazena mensagens que não podem ser entregues ou que não foram processadas com sucesso pelo daemon.
- `AWSEBWorkerCronLeaderRegistry`(AWS::DynamoDB::Table) — A tabela do Amazon DynamoDB que é o registro interno usado pelo daemon para tarefas periódicas.

Outras chaves AWS CloudFormation de modelo

Já introduzimos chaves de arquivo de configuraçãoResources, AWS CloudFormation comofiles, packages e. O Elastic Beanstalk adiciona o conteúdo dos arquivos de configuração ao modelo que dá suporte AWS CloudFormation ao seu ambiente, para que você possa AWS CloudFormation usar outras seções para realizar tarefas avançadas em seus arquivos de configuração.

Chaves

- Parâmetros
- Saídas
- Mapeamentos

Parâmetros

Os parâmetros são uma alternativa para as opções personalizadas próprias do Elastic Beanstalk que podem ser usadas para definir valores utilizados em outros lugares nos arquivos de configuração. Do mesmo modo que as opções personalizadas, você pode usar parâmetros para reunir valores configuráveis pelo usuário em um único lugar. Ao contrário das opções personalizadas, você não pode usar a API do Elastic Beanstalk para definir valores de parâmetros, e o número de parâmetros que você pode definir em um modelo é limitado por. AWS CloudFormation

Um dos motivos pelos quais você pode querer usar parâmetros é fazer com que seus arquivos de configuração funcionem como AWS CloudFormation modelos. Se você usar parâmetros em vez de opções personalizadas, poderá usar o arquivo de configuração para criar o mesmo recurso em AWS CloudFormation sua própria pilha. Por exemplo, você pode ter um arquivo de configuração que adiciona um sistema de arquivos do Amazon EFS ao seu ambiente para fins de teste, e depois usar o mesmo arquivo para criar um sistema de arquivos independentes que não estejam vinculados ao ciclo de vida do seu ambiente para ser usado em produção.

O exemplo a seguir mostra a utilização de parâmetros para reunir valores configuráveis pelo usuário no início de um arquivo de configuração.

Example L oadbalancer-accesslogs-existingbucket .config — Parâmetros

```
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: ""
```

Saídas

Você pode usar um bloco de `Outputs` para exportar informações sobre os recursos criados para o AWS CloudFormation. Em seguida, você pode usar a `Fn::ImportValue` função para inserir o valor em um AWS CloudFormation modelo fora do Elastic Beanstalk.

O exemplo a seguir cria um tópico do Amazon SNS e exporta seu ARN para AWS CloudFormation com o nome. `NotificationTopicArn`

Example sns-topic.config

```
Resources:
  NotificationTopic:
    Type: AWS::SNS::Topic

Outputs:
  NotificationTopicArn:
    Description: Notification topic ARN
    Value: { "Ref" : "NotificationTopic" }
    Export:
      Name: NotificationTopicArn
```

Em um arquivo de configuração para um ambiente diferente ou em um AWS CloudFormation modelo fora do Elastic Beanstalk, você pode `Fn::ImportValue` usar a função para obter o ARN exportado. Este exemplo atribui o valor exportado a uma propriedade do ambiente chamada `TOPIC_ARN`.

Example env.config

```
option_settings:
  aws:elasticbeanstalk:application:environment:
    TOPIC_ARN: '`{ "Fn::ImportValue" : "NotificationTopicArn" }`'
```

Mapeamentos

Você pode usar um mapeamento para armazenar pares de chave-valor organizados por namespace. Um mapeamento pode ajudar você a organizar os valores que utiliza em toda as configurações, ou

alterar o valor de um parâmetro dependendo de outro valor. Por exemplo, a configuração a seguir define o valor de um parâmetro de ID de conta com base na região atual.

Example L oadbalancer-accesslogs-newbucket .config — Mapeamentos

```
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
```

Funções

É possível usar funções nos arquivos de configuração para preencher valores de propriedades de recursos com informações de outros recursos ou das definições de opção de configuração do Elastic Beanstalk. O Elastic AWS CloudFormation Beanstalk suporta `Ref` funções `Fn::GetAtt` (`Fn::Join`,,) e uma função específica do Elastic Beanstalk,. `Fn::GetOptionSetting`

Funções

- Ref.
- Fã:: GetAtt
- Fn::Join
- Fã:: GetOptionSetting

Ref.

Use `Ref` para recuperar a representação de string padrão de um AWS recurso. O valor retornado por `Ref`depende do tipo de recurso e, às vezes, depende também de outros fatores. Por exemplo, um grupo de segurança (AWS::EC2::SecurityGroup) retorna o nome ou a ID do grupo de segurança, dependendo se o grupo de segurança está em uma Amazon Virtual Private Cloud (Amazon VPC) padrão, EC2 clássica ou em uma VPC personalizada.

```
{ "Ref" : "resource name" }
```

> **Note**
>
> Para obter detalhes sobre cada tipo de recurso, incluindo os valores de retorno de `Ref`, consulte Referência de tipos de recursos da AWS no Guia do usuário do AWS CloudFormation .

A partir do gancho de ciclo de vida do Auto Scaling demonstrativo:

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
      AutoScalingGroupName: { "Ref" : "AWSEBAutoScalingGroup" }
```

Você também pode usar `Ref` para recuperar o valor de um AWS CloudFormation parâmetro definido em outro lugar no mesmo arquivo ou em um arquivo de configuração diferente.

Fã:: GetAtt

Use `Fn::GetAtt` para recuperar o valor de um atributo em um AWS recurso.

```
{ "Fn::GetAtt" : [ "resource name", "attribute name"] }
```

A partir do gancho de ciclo de vida do Auto Scaling demonstrativo:

```
Resources:
  lifecyclehook:
    Type: AWS::AutoScaling::LifecycleHook
    Properties:
```

```
      RoleARN: { "Fn::GetAtt" : [ "hookrole", "Arn"] }
```

Consulte Fn:: GetAtt para obter mais informações.

Fn::Join

Use `Fn::Join` para combinar strings com um delimitador. As strings podem ser codificadas ou usar a saída de `Fn::GetAtt` ou `Ref`.

```
{ "Fn::Join" : [ "delimiter", [ "string1", "string2" ] ] }
```

Consulte Fn::Join para obter mais informações.

Fã:: GetOptionSetting

Use `Fn::GetOptionSetting` para recuperar o valor de uma definição de opção de configuração aplicada ao ambiente.

```
"Fn::GetOptionSetting":
  Namespace: "namespace"
  OptionName: "option name"
  DefaultValue: "default value"
```

No exemplo de armazenamento de chaves privadas:

```
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"
```

Exemplos de recursos personalizados

Veja a seguir uma lista de arquivos de configuração demonstrativos que podem ser usados para personalizar os ambientes do Elastic Beanstalk:

- DynamoDB CloudWatch e SNS
- Elastic Load Balancing e CloudWatch
- ElastiCache
- RDS e CloudWatch
- SQS, SNS e CloudWatch

Os subtópicos desta página fornecem alguns exemplos estendidos para adicionar e configurar recursos personalizados em um ambiente do Elastic Beanstalk.

Exemplos

- Exemplo: ElastiCache.
- Exemplo: SQS e CloudWatch SNS
- Exemplo: DynamoDB CloudWatch e SNS

Exemplo: ElastiCache.

Os exemplos a seguir adicionam um ElastiCache cluster da Amazon às EC2 plataformas -Classic e EC2 -VPC (ambas padrão e personalizadas Amazon Virtual Private Cloud (Amazon VPC)). Para obter mais informações sobre essas plataformas e como você pode determinar quais são EC2 compatíveis com sua região e sua AWS conta, consultehttps://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html. Depois, consulte a seção neste tópico que se aplica à sua plataforma.

- EC2-plataformas clássicas
- EC2-VPC (padrão)
- EC2-VPC (personalizado)

EC2-plataformas clássicas

Este exemplo adiciona um ElastiCache cluster da Amazon a um ambiente com instâncias lançadas na plataforma EC2 -Classic. Todas as propriedades listadas neste exemplo são as propriedades mínimas necessárias que devem ser definidas para cada tipo de recurso. Você pode baixar o exemplo em ElastiCacheexample.

> **Note**
>
> Este exemplo cria AWS recursos, pelos quais você pode ser cobrado. Para obter mais informações sobre AWS preços, consultehttps://aws.amazon.com/pricing/. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte https://aws.amazon.com/free/ para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

   YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Crie um arquivo de configuração (por exemplo, `elasticache.config`) que define os recursos. Neste exemplo, criamos o ElastiCache cluster especificando o nome do recurso do ElastiCache cluster (`MyElastiCache`), declarando seu tipo e, em seguida, configurando as propriedades do cluster. O exemplo faz referência ao nome do recurso do grupo de ElastiCache segurança que é criado e definido nesse arquivo de configuração. Em seguida, criamos um grupo ElastiCache de segurança. Definimos o nome deste recurso, declaramos seu tipo e adicionamos uma descrição para o security group. Por fim, definimos as regras de entrada do grupo de ElastiCache segurança para permitir o acesso somente de instâncias dentro do grupo de ElastiCache segurança (`MyCacheSecurityGroup`) e do grupo de segurança do Elastic Beanstalk (). `AWSEBSecurityGroup` O nome do parâmetro, `AWSEBSecurityGroup`, é o nome do recurso fixo fornecido pelo Elastic Beanstalk. Você deve adicionar regras de entrada `AWSEBSecurityGroup` ao seu grupo de ElastiCache segurança para que seu aplicativo Elastic Beanstalk se conecte às instâncias em seu 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
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- AWS::ElastiCache::CacheCluster
- AWS::ElastiCache::SecurityGroup
- AWS::ElastiCache: SecurityGroupIngress

Crie um arquivo de configuração separado chamado `options.config` e defina as opções de configurações personalizadas.

```
option_settings:
```

```
  "aws:elasticbeanstalk:customoption":
     CacheNodeType : cache.m1.small
     NumCacheNodes : 1
     Engine : memcached
```

Essas linhas fazem com que o Elastic Beanstalk obtenha os valores das propriedades,, e Engine dos valores CacheNodeType, e Engine em um arquivo CacheNodeTypede NumCacheNodes configuração (options.config em nosso exemplo) que contém uma seção option_settings com uma seção aws:elasticbeanstalk:customoption que contém um par nome-valor que contém o valor real a ser usado. NumCacheNodes No exemplo acima, isso significa que cache.m1.small, 1 e Memcached seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte Funções.

EC2-VPC (padrão)

Este exemplo adiciona um ElastiCache cluster da Amazon a um ambiente com instâncias lançadas na plataforma EC2 -VPC. Especificamente, as informações nesta seção se aplicam a um cenário em que as EC2 instâncias são executadas na VPC padrão. Todas as propriedades neste exemplo são as mínimas necessárias que devem ser definidas para cada tipo de recurso. Para obter mais informações sobre o padrão VPCs, consulte Sua VPC e sub-redes padrão.

> **Note**
>
> Este exemplo cria AWS recursos, pelos quais você pode ser cobrado. Para obter mais informações sobre AWS preços, consultehttps://aws.amazon.com/pricing/. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte https://aws.amazon.com/free/ para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Agora nomeie o arquivo de configuração de recursos `elasticache.config`. Para criar o ElastiCache cluster, este exemplo especifica o nome do recurso do ElastiCache cluster (`MyElastiCache`), declara seu tipo e, em seguida, configura as propriedades do cluster. O exemplo faz referência ao ID do recurso do security group que criamos e definimos neste arquivo de configuração.

Em seguida, criamos um grupo EC2 de segurança. Definimos o nome deste recurso, declaramos o tipo, adicionamos uma descrição e definimos as regras de entrada no security group para permitir o acesso somente a partir de instâncias dentro do security group do Elastic Beanstalk (`AWSEBSecurityGroup`). (O nome do parâmetro,`AWSEBSecurityGroup`, é um nome de recurso fixo fornecido pelo Elastic Beanstalk. Você deve adicionar regras de entrada `AWSEBSecurityGroup` ao seu grupo de ElastiCache segurança para que seu aplicativo Elastic Beanstalk se conecte às instâncias em seu cluster.) ElastiCache

As regras de entrada para o grupo EC2 de segurança também definem o protocolo IP e os números das portas nas quais os nós de cache podem aceitar conexões. Para Redis, o número padrão da porta é 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"
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- AWS::ElastiCache::CacheCluster
- AWS::EC2::SecurityGroup

Depois, nomeie o arquivo de configuração de opções `options.config` e defina as opções de configurações personalizadas.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
    CacheNodeType : cache.t2.micro
```

```
    NumCacheNodes : 1
    Engine : redis
    CachePort : 6379
```

Essas linhas instruem o Elastic Beanstalk para obter os valores para as propriedades `CacheNodeType`, `NumCacheNodes`, `Engine` e `CachePort` dos valores `CacheNodeType`, `NumCacheNodes`, `Engine` e `CachePort` em um arquivo de configuração (`options.config` em nosso exemplo). Esse arquivo inclui uma seção de `aws:elasticbeanstalk:customoption` (em `option_settings`) que contém pares de nome e valor com o valores reais para serem usados. No exemplo anterior, `cache.t2.micro`, `1`, `redis` e `6379` seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte Funções.

EC2-VPC (personalizado)

Se você criar uma VPC personalizada na plataforma EC2 -VPC e especificá-la como a VPC na qual executa as instâncias, o processo de adicionar um ElastiCache cluster da Amazon ao seu ambiente EC2 será diferente daquele de uma VPC padrão. A principal diferença é que você precisa criar um grupo de sub-redes para o ElastiCache cluster. Todas as propriedades neste exemplo são as mínimas necessárias que devem ser definidas para cada tipo de recurso.

> **Note**
>
> Este exemplo cria AWS recursos, pelos quais você pode ser cobrado. Para obter mais informações sobre AWS preços, consultehttps://aws.amazon.com/pricing/. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte https://aws.amazon.com/free/ para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

   YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Agora nomeie o arquivo de configuração de recursos `elasticache.config`. Para criar o ElastiCache cluster, este exemplo especifica o nome do recurso do ElastiCache cluster (`MyElastiCache`), declara seu tipo e, em seguida, configura as propriedades do cluster. As propriedades no exemplo fazem referência ao nome do grupo de sub-redes do ElastiCache cluster, bem como ao ID do recurso do grupo de segurança que criamos e definimos nesse arquivo de configuração.

Em seguida, criamos um grupo EC2 de segurança. Definimos o nome deste recurso, declaramos o tipo, adicionamos uma descrição, o ID da VPC, e definimos as regras de entrada no security group para permitir o acesso somente a partir das instâncias dentro do security group do Elastic Beanstalk (`AWSEBSecurityGroup`). (O nome do parâmetro,`AWSEBSecurityGroup`, é um nome de recurso fixo fornecido pelo Elastic Beanstalk. Você deve adicionar regras de entrada `AWSEBSecurityGroup` ao seu grupo de ElastiCache segurança para que seu aplicativo Elastic Beanstalk se conecte às instâncias em seu cluster.) ElastiCache

As regras de entrada para o grupo EC2 de segurança também definem o protocolo IP e os números das portas nas quais os nós de cache podem aceitar conexões. Para Redis, o número padrão da porta é `6379`. Por fim, esse exemplo cria um grupo de sub-redes para o ElastiCache cluster. Definimos o nome desse recurso, declaramos seu tipo e adicionamos uma descrição e um ID da sub-rede no grupo de sub-rede.

> **Note**
>
> Recomendamos que você use sub-redes privadas para o ElastiCache cluster. Para obter mais informações sobre uma VPC com uma sub-rede privada, consulte 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"
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- AWS::ElastiCache::CacheCluster
- AWS::EC2::SecurityGroup
- AWS::ElastiCache::SubnetGroup

Depois, nomeie o arquivo de configuração de opções `options.config` e defina as opções de configurações personalizadas.

> **Note**
>
> No exemplo a seguir, substitua o exemplo `CacheSubnets` e os valores de `VpcId` com suas próprias sub-redes e VPC.

```
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
```

Essas linhas instruem o Elastic Beanstalk para obter os valores para as propriedades `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` e `VpcId` dos valores `CacheNodeType`, `NumCacheNodes`, `Engine`, `CachePort`, `CacheSubnets` e `VpcId` em um arquivo de configuração (`options.config` em nosso exemplo). Esse arquivo inclui uma seção de `aws:elasticbeanstalk:customoption` (em `option_settings`) que contém pares de nome e valor com os valores de exemplo. No exemplo acima, `cache.t2.micro`, `1`, `redis`, `6379`, `subnet-1a1a1a1a`, `subnet-2b2b2b2b`, `subnet-3c3c3c3c` e `vpc-4d4d4d4d` seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte Funções.

Exemplo: SQS e CloudWatch SNS

Este exemplo adiciona uma fila do Amazon SQS e um alarme no comprimento da fila para o ambiente. As propriedades que você vê neste exemplo são as mínimas necessárias a serem definidas para cada um desses recursos. Você pode baixar o exemplo em SQS, SNS e. CloudWatch

> **Note**
>
> Este exemplo cria AWS recursos, pelos quais você pode ser cobrado. Para obter mais informações sobre AWS preços, consultehttps://aws.amazon.com/pricing/. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte https://aws.amazon.com/free/ para obter mais informações.

Para usar esse exemplo, faça o seguinte:

1. Crie um diretório `.ebextensions` no diretório de nível superior do pacote de origem.
2. Crie dois arquivos de configuração com a extensão `.config` e os coloque no diretório `.ebextensions`. Um arquivo de configuração define os recursos, e o outro arquivo de configuração define as opções.
3. Implante a aplicação no Elastic Beanstalk.

   YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Crie um arquivo de configuração (por exemplo, sqs.config) que define os recursos. Neste exemplo, criamos uma fila SQS e definimos a propriedade `VisbilityTimeout` no recurso `MySQSQueue`. Em seguida, criamos um `Topic` SNS e especificamos que o e-mail seja enviado para `someone@example.com` quando o alarme for acionado. Por fim, criamos um CloudWatch alarme se a fila aumentar além de 10 mensagens. Na propriedade `Dimensions`, especificamos o nome da dimensão e o valor que representa a medida da dimensão. Usamos `Fn::GetAtt` para retornar o valor de `QueueName` do `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"]}
```

Para obter mais informações sobre os recursos usados neste exemplo de arquivo de configuração, consulte as seguintes referências:

- AWS::SQS::Queue
- AWS::SNS::Topic
- AWS::CloudWatch::Alarm

Crie um arquivo de configuração separado chamado `options.config` e defina as opções de configurações personalizadas.

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     VisibilityTimeout : 30
     AlarmEmail : "nobody@example.com"
```

Essas linhas instruem o Elastic Beanstalk a obter os valores das propriedades e Subscription Endpoint dos valores VisibilityTimeout e Subscription Endpoint em um arquivo VisibilityTimeout de configuração (options.config em nosso exemplo) que contém uma seção option_settings com uma seção aws:elasticbeanstalk:customoption que contém um par nome-valor que contém o valor real a ser usado. No exemplo acima, isso significa que 30 e "nobody@amazon.com" seriam usados para os valores. Para obter mais informações sobre o `Fn::GetOptionSetting`, consulte the section called “Funções”.

Exemplo: DynamoDB CloudWatch e SNS

Esse arquivo de configuração configura a tabela do DynamoDB como um manipulador de sessão para um aplicativo baseado em PHP usando o SDK for PHP 2. AWS Para usar este exemplo, você deve ter um perfil de instância do IAM, que é adicionado às instâncias em seu ambiente e usado para acessar a tabela do DynamoDB.

Você poderá fazer download do exemplo que usaremos nesta etapa em Suporte de sessão do DynamoDB demonstrativo. A amostra contém os seguintes arquivos:

- O aplicativo de exemplo, `index.php`
- Um arquivo de configuração,`dynamodb.config`, para criar e configurar uma tabela do DynamoDB e AWS outros recursos e instalar o software nas instâncias que hospedam EC2 o aplicativo em um ambiente do Elastic Beanstalk

- Um arquivo de configuração, `options.config`, que substitui os valores padrão em `dynamodb.config` por configurações específicas para essa determinada instalação

`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.*"
```

No arquivo de configuração de exemplo, primeiro crie a tabela do DynamoDB e configure a chave primária da tabela e as unidades de capacidade para alocar recursos suficientes para fornecer a taxa de transferência solicitada. Em seguida, criamos CloudWatch alarmes para `WriteCapacity` e. `ReadCapacity` Criamos um tópico do SNS que envia um e-mail para "nobody@amazon.com" caso os limites de alarme sejam violados.

Depois de criar e configurar nossos AWS recursos para nosso ambiente, precisamos personalizar as EC2 instâncias. Usamos a `files` chave para passar os detalhes da tabela do DynamoDB para EC2 as instâncias em nosso ambiente, bem como adicionar um “require” no arquivo `composer.json` do SDK AWS for PHP 2. Por fim, executamos comandos de contêiner para instalar o Composer, as dependências necessárias e, em seguida, remover o instalador.

`.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
```

Substitua o SessionAlarmEmail valor pelo e-mail para o qual você deseja que as notificações de alarme sejam enviadas. O arquivo `options.config` contém os valores usados para algumas das variáveis definidas em `dynamodb.config`. Por exemplo, `dynamodb.config` contém as seguintes linhas:

```
Subscription:
  - Endpoint:
      Fn::GetOptionSetting:
        OptionName: SessionAlarmEmail
        DefaultValue: "nobody@amazon.com"
```

Essas linhas que fazem com que o Elastic Beanstalk obtenha o valor da propriedade Endpoint a SessionAlarmEmailpartir do valor em um **`options.config`** arquivo de configuração (em nosso aplicativo de amostra) que contém uma seção option_settings com uma seção aws:elasticbeanstalk:customoption que contém um par nome-valor que contém o valor real a ser usado. No exemplo acima, isso significa que SessionAlarmEmailseria atribuído o valor`nobody@amazon.com`.

Para obter mais informações sobre os CloudFormation recursos usados neste exemplo, consulte as seguintes referências:

- AWS::DynamoDB::Table
- AWS::CloudWatch::Alarm
- AWS::SNS::Topic

Usar configurações salvas do Elastic Beanstalk

Você pode salvar a configuração de seu ambiente como um objeto no Amazon Simple Storage Service (Amazon S3) para aplicar a outros ambientes durante a criação ou para aplicar a um ambiente em execução. Configurações salvas são modelos formatados em YAML que definem a versão da plataforma, o nível, as definições das opções de configuração e as tags de um ambiente.

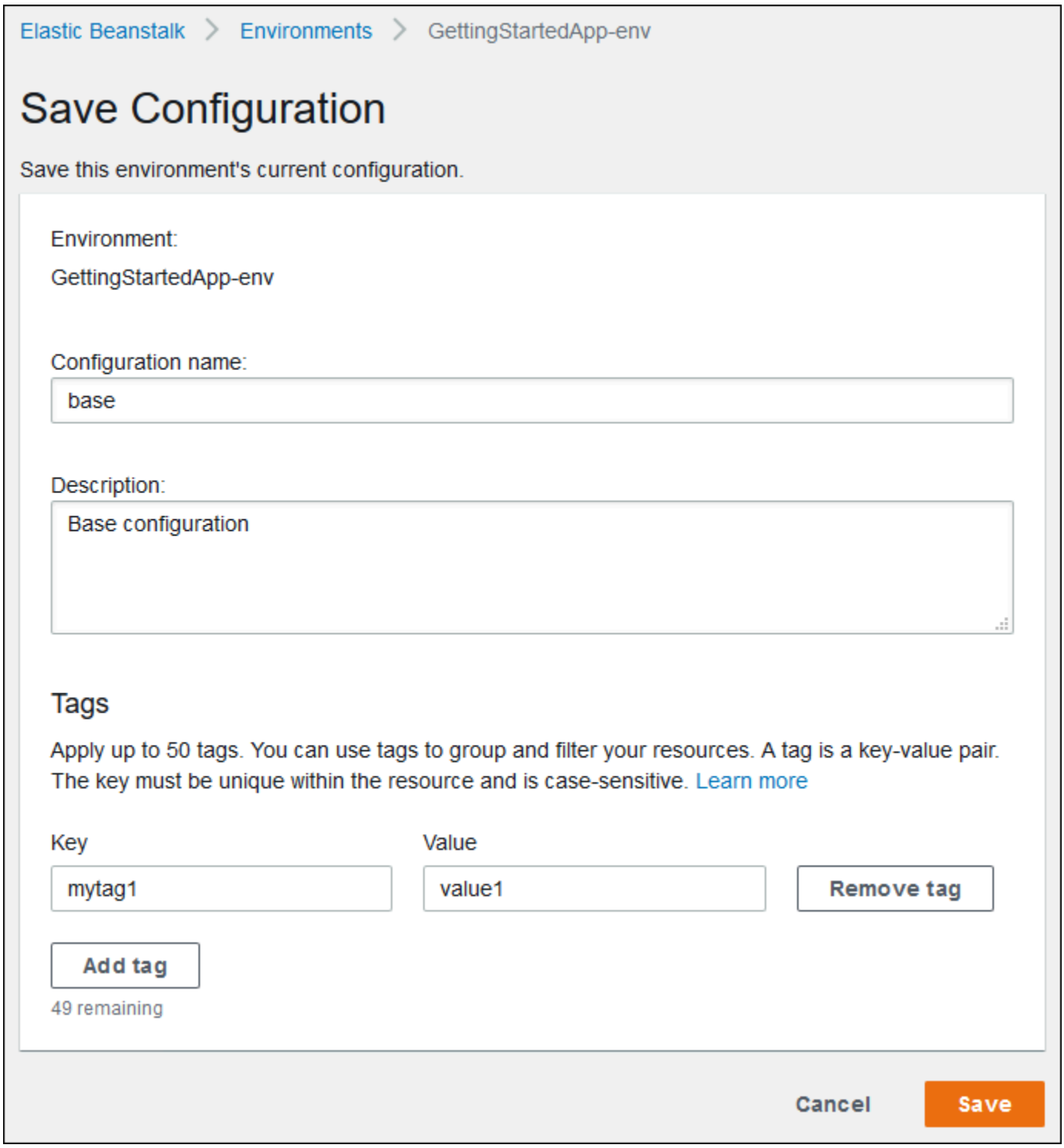
Você pode aplicar tags a uma configuração salva ao criá-la e editar tags de configurações salvas existentes. As tags aplicadas a uma configuração salva não estão relacionadas às tags especificadas em uma configuração salva que usa a chave `Tags:`. Os últimos são aplicados a um ambiente quando você aplica a configuração salva ao ambiente. Para obter detalhes, consulte Marcação de configurações salvas.

> **Note**
>
> Você pode criar e aplicar configurações salvas em seus ambientes do Elastic Beanstalk usando vários métodos. Isso inclui o console do Elastic Beanstalk, a CLI do EB e o AWS CLI. Consulte os seguintes tópicos para ver exemplos de métodos alternativos de criação e aplicação de configurações salvas:
>
> - Definir as opções de configuração antes da criação do ambiente
> - Definir as opções de configuração durante a criação do ambiente
> - Definir as opções de configuração depois da criação do ambiente

Crie uma configuração salva com base no estado atual do ambiente no console de gerenciamento do Elastic Beanstalk.

Para salvar a configuração de um ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Save Configuration (Salvar configuração).
4. Use o formulário na tela para nomear a configuração salva. Opcionalmente, forneça uma breve descrição e adicione chaves e valores de tag.
5. Escolha Salvar.



A configuração salva inclui todas as configurações aplicadas ao ambiente com o console ou qualquer outro cliente que usa a API do Elastic Beanstalk. Em seguida, você pode aplicar a configuração salva ao ambiente em uma data futura para restaurá-lo ao seu estado anterior ou aplicá-la a um novo ambiente durante a criação dele.

Você pode baixar uma configuração usando o the section called “eb config” comando EB CLI, conforme mostrado no exemplo a seguir. *NAME*é o nome da configuração salva.

```
eb config get NAME
```

Como aplicar uma configuração salva durante a criação do ambiente (console do Elastic Beanstalk)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.
3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja aplicar e escolha Launch environment (Iniciar ambiente).
5. Continue no assistente para criar seu ambiente.

As configurações salvas não incluem as definições aplicadas com os arquivos de configuração no código-fonte de seu aplicativo. Se a mesma configuração for aplicada tanto em um arquivo de configuração quanto em uma configuração salva, a da configuração salva terá precedência. Da mesma forma, as opções especificadas no console do Elastic Beanstalk substituem as das configurações salvas. Para obter mais informações, consulte Precedência.

As configurações salvas são armazenadas no bucket do S3 do Elastic Beanstalk em uma pasta com o mesmo nome da sua aplicação. Por exemplo, as configurações de uma aplicação chamada `my-app` na região us-west-2 da conta número 123456789012 podem ser encontradas em `s3://elasticbeanstalk-us-west-2-123456789012/resources/templates/my-app/`.

Visualize o conteúdo de uma configuração salva abrindo-a em um editor de texto. O exemplo a seguir mostra a configuração de um ambiente de servidor Web iniciado com o console de gerenciamento do 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
```

Você pode modificar o conteúdo de uma configuração salva e salvá-lo no mesmo local no Amazon S3. Qualquer configuração salva formatada adequadamente armazenada no local certo pode ser aplicada a um ambiente usando o console de gerenciamento do Elastic Beanstalk.

As seguintes chaves são compatíveis.

- AWSConfigurationTemplateVersion(obrigatório) — A versão do modelo de configuração (1.1.0.0).

  ```
  AWSConfigurationTemplateVersion: 1.1.0.0
  ```

- Plataforma: o nome de recurso da Amazon (ARN) da versão da plataforma do ambiente. Você pode especificar a plataforma pelo ARN ou pelo nome da pilha de soluções.

```
Platform:
  PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
 Amazon Linux/2.5.0
```

- SolutionStack— O nome completo da pilha de soluções usada para criar o ambiente.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings— Configurações das opções de configuração a serem aplicadas ao ambiente. Por exemplo, a seguinte entrada define o tipo de instância como t2.micro.

```
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: t2.micro
```

- Tags: até 47 tags a serem aplicadas aos recursos criados no ambiente.

```
Tags:
  Cost Center: WebApp Dev
```

- EnvironmentTier— O tipo de ambiente a ser criado. Para um ambiente de servidor Web, você pode excluir esta seção (servidor Web é o padrão). Para um ambiente de operador, use o seguinte.

```
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
```

> **Note**
>
> Você pode criar e aplicar configurações salvas em seus ambientes do Elastic Beanstalk usando vários métodos. Isso inclui o console do Elastic Beanstalk, a CLI do EB e o AWS CLI. Consulte os seguintes tópicos para ver exemplos de métodos alternativos de criação e aplicação de configurações salvas:
>
> - Definir as opções de configuração antes da criação do ambiente
> - Definir as opções de configuração durante a criação do ambiente
> - Definir as opções de configuração depois da criação do ambiente

Marcação de configurações salvas

Você pode aplicar tags às suas configurações AWS Elastic Beanstalk salvas. As tags são pares de valores-chave associados AWS aos recursos. Para obter informações sobre a atribuição de tags do recurso do Elastic Beanstalk, casos de uso, restrições de chave e valor de tag, além de tipos de recursos compatíveis, consulte Marcar recursos da aplicação do Elastic Beanstalk.

Você pode especificar tags quando criar uma configuração salva. Em uma configuração salva existente, você pode adicionar ou remover tags e atualizar os valores de tags existentes. Você pode adicionar até 50 tags para cada configuração salva.

Adicionar tags durante a criação de configurações salvas

Ao usar o console do Elastic Beanstalk para salvar uma configuração, é possível especificar chaves e valores de tags na página Save Configuration (Salvar configuração).

Se você usar a CLI do EB para salvar uma configuração, use a opção `--tags` com eb config para adicionar tags.

```
~/workspace/my-app$ eb config --tags mytag1=value1,mytag2=value2
```

Com o AWS CLI ou outros clientes baseados em API, adicione tags usando o `--tags` parâmetro no create-configuration-template comando.

```
$ 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
```

Gerenciar tags de uma configuração salva existente

É possível adicionar, atualizar e excluir tags em uma configuração salva existente do Elastic Beanstalk.

Como gerenciar as tags de uma configuração salva usando o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Aplicativos e escolha o nome do aplicativo na lista.

3. No painel de navegação, encontre o nome do aplicativo e escolha Saved configurations (Configurações salvas).
4. Selecione a configuração salva que deseja gerenciar.
5. Escolha Actions (Ações) e Manage tags (Gerenciar tags).
6. Use o formulário na tela para adicionar, atualizar ou excluir tags.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você usar a CLI do EB para atualizar a configuração salva, use eb tags para adicionar, atualizar, excluir ou listar tags.

Por exemplo, o comando a seguir lista as tags em uma configuração salva.

```
~/workspace/my-app$ eb tags --list --resource "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

O comando a seguir atualiza a tag `mytag1` e exclui a 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"
```

Para obter uma lista de opções e mais exemplos, consulte `eb tags`.

Com o AWS CLI ou outros clientes baseados em API, use o list-tags-for-resource comando para listar as tags de uma configuração salva.

```
$ aws elasticbeanstalk list-tags-for-resource --resource-arn
 "arn:aws:elasticbeanstalk:us-east-2:my-account-id:configurationtemplate/my-app/my-template"
```

Use o comando update-tags-for-resource para adicionar, atualizar ou excluir tags em uma configuração salva.

```
$ 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"
```

Especifique as tags a serem adicionadas e atualizadas no parâmetro `--tags-to-add` do update-tags-for-resource. Uma tag não existente é adicionada, e o valor de uma tag existente é atualizado.

> **Note**
>
> Para usar alguns AWS CLI comandos e CLI do EB com uma configuração salva do Elastic Beanstalk, você precisa do ARN da configuração salva. Para construir o ARN, primeiro use o comando a seguir para recuperar o nome da configuração salva.
>
> ```
> $ aws elasticbeanstalk describe-applications --application-names my-app
> ```
>
> Procure a chave `ConfigurationTemplates` na saída do comando. Esse elemento mostra o nome da configuração salva. Use esse nome, em que *`my-template`* é especificado nos comandos mencionados nessa página.

Manifesto de ambiente (**`env.yaml`**)

Você pode incluir um manifesto de ambiente formatado com YAML na raiz do seu pacote de origem do aplicativo para configurar o nome do ambiente, a pilha de soluções e os links do ambiente para usar ao criá-lo.

Este formato de arquivo inclui suporte para grupos do ambiente. Para usar grupos, especifique o nome do ambiente no manifesto com um símbolo + no final. Ao criar ou atualizar o ambiente, especifique o nome do grupo com `--group-name` (AWS CLI) ou `--env-group-suffix` (EB CLI). Para obter mais informações sobre grupos, consulte Criar e atualizar grupos de ambientes do Elastic Beanstalk.

O manifesto de exemplo a seguir define um ambiente de servidor Web com um link para um componente do ambiente de operador do qual ele depende. O manifesto usa grupos para permitir a criação de vários ambientes com o mesmo pacote de origem:

`~/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+"
```

As seguintes chaves são compatíveis.

- AWSConfigurationTemplateVersion(obrigatório) — A versão do modelo de configuração (1.1.0.0).

  ```
  AWSConfigurationTemplateVersion: 1.1.0.0
  ```

- Plataforma: o nome de recurso da Amazon (ARN) da versão da plataforma do ambiente. Você pode especificar a plataforma pelo ARN ou pelo nome da pilha de soluções.

  ```
  Platform:
    PlatformArn: arn:aws:elasticbeanstalk:us-east-2::platform/Java 8 running on 64bit
   Amazon Linux/2.5.0
  ```

- SolutionStack— O nome completo da pilha de soluções usada para criar o ambiente.

```
SolutionStack: 64bit Amazon Linux 2017.03 v2.5.0 running Java 8
```

- OptionSettings— Configurações das opções de configuração a serem aplicadas ao ambiente. Por exemplo, a seguinte entrada define o tipo de instância como t2.micro.

```
OptionSettings:
  aws:autoscaling:launchconfiguration:
    InstanceType: t2.micro
```

- Tags: até 47 tags a serem aplicadas aos recursos criados no ambiente.

```
Tags:
  Cost Center: WebApp Dev
```

- EnvironmentTier— O tipo de ambiente a ser criado. Para um ambiente de servidor Web, você pode excluir esta seção (servidor Web é o padrão). Para um ambiente de operador, use o seguinte.

```
EnvironmentTier:
  Name: Worker
  Type: SQS/HTTP
```

- CName— O CNAME para o ambiente. Incluir um caractere + no fim do nome para habilitar grupos.

```
CName: front-A08G28LG+
```

- EnvironmentName— O nome do ambiente a ser criado. Incluir um caractere + no fim do nome para habilitar grupos.

```
EnvironmentName: front+
```

  Com os grupos habilitados, você deve especificar o nome do grupo ao criar os ambientes. O Elastic Beanstalk anexa o nome do grupo ao nome do ambiente com um hífen. Por exemplo, com o nome do ambiente `front+` e o nome do grupo `dev`, o Elastic Beanstalk criará o ambiente com o nome `front-dev`.

- EnvironmentLinks— Um mapa de nomes de variáveis e nomes de ambientes de dependências. O exemplo a seguir torna o ambiente `worker+` uma dependência e instrui o Elastic Beanstalk a salvar as informações do link em uma variável denominada `WORKERQUEUE`.

```
EnvironmentLinks:
```

```
  "WORKERQUEUE" : "worker+"
```

O valor da variável de link muda de acordo com o tipo de ambiente vinculado. Para um ambiente de servidor Web, o link é o CNAME do ambiente. Para um ambiente de operador, o link é o nome da fila do Amazon Simple Queue Service (Amazon SQS) do ambiente.

As EnvironmentLinksteclas CName, EnvironmentNamee podem ser usadas para criar grupos de ambientes e links para outros ambientes. Atualmente, esses recursos são suportados ao usar o EB CLI AWS CLI ou um SDK.

Usar uma imagem de máquina Amazon personalizada (AMI) no ambiente Elastic Beanstalk

Esta seção explica quando considerar o uso de uma AMI personalizada e fornece os procedimentos para configurar e gerenciar a AMI personalizada no seu ambiente. Ao criar um AWS Elastic Beanstalk ambiente, você pode especificar uma Amazon Machine Image (AMI) para usar em vez da AMI padrão do Elastic Beanstalk incluída na versão da sua plataforma. Uma AMI personalizada pode melhorar os tempos de provisionamento quando as instâncias são lançadas em seu ambiente se você precisar instalar muitos softwares que não estão incluídos no padrão. AMIs

O uso de arquivos de configuração é eficaz para personalizar seu ambiente de maneira rápida e consistente. Embora a aplicação de configurações possa começar a levar muito tempo durante a criação e as atualizações do ambiente. Se você especificar muitas configurações de servidor em arquivos de configuração, poderá reduzir esse tempo ao criar uma AMI personalizada que já tenha o software e a configuração de que precisa.

Uma AMI personalizada também permite fazer alterações em componentes de baixo nível, como o kernel do Linux, que são difíceis de implantar ou que demoram muito tempo para aplicar em arquivos de configuração. Para criar uma AMI personalizada, execute uma AMI da plataforma Elastic Beanstalk EC2 na Amazon, personalize o software e a configuração de acordo com suas necessidades e, em seguida, interrompa a instância e salve uma AMI a partir dela.

Criar uma AMI do personalizada

Você pode usar o EC2 Image Builder para criar e gerenciar a personalização AMIs como uma alternativa a esses procedimentos. Para obter mais informações, consulte o Guia do usuário do Image Builder.

Como identificar a AMI base do Elastic Beanstalk

1. Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte describe-platform-version na Referência de comandos da AWS CLI .

   Especifique a AWS região em que você deseja usar sua AMI personalizada e substitua o ARN da plataforma e o número da versão pela plataforma Elastic Beanstalk na qual seu aplicativo se baseia.

   Example - Mac OS/SO 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 - SO 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. Anote o valor `ImageId`, que será semelhante a `ami-020ae06fdda6a0f66` no resultado.

O valor é a AMI padrão do Elastic Beanstalk para a EC2 versão da plataforma, a arquitetura da instância e a região que são AWS relevantes para seu aplicativo. Se você precisar criar AMIs para várias plataformas, arquiteturas ou AWS regiões, repita esse processo para identificar a AMI base correta para cada combinação.

> **Note**
>
> Não crie uma AMI a partir de uma instância que foi executada em um ambiente do Elastic Beanstalk. O Elastic Beanstalk faz alterações nas instâncias durante o provisionamento que podem causar problemas na AMI salva. Salvar uma imagem de uma instância em um ambiente do Elastic Beanstalk também faz com que a versão da aplicação que foi implantada na instância seja uma parte fixa da imagem.

Para o Linux, também é possível criar uma AMI personalizada a partir de uma AMI de comunidade que não foi publicada pelo Elastic Beanstalk. É possível usar a AMI mais recente do Amazon Linux como ponto de partida. Quando você executa um ambiente com uma AMI do Linux que não é gerenciada pelo Elastic Beanstalk, o Elastic Beanstalk tenta instalar o software da plataforma (idioma, estrutura, servidor de proxy, etc.) e os componentes adicionais para oferecer suporte a recursos como relatórios de integridade aprimorada.

> **Note**
>
> Os modelos personalizados AMIs baseados no Windows Server exigem que a AMI padrão do Elastic Beanstalk seja `describe-platform-version` retornada, conforme mostrado anteriormente na Etapa 1.

Embora o Elastic Beanstalk possa usar uma AMI não gerenciada pelo Elastic Beanstalk, o aumento no tempo de provisionamento que resulta da instalação pelo Elastic Beanstalk de componentes ausentes pode reduzir ou eliminar os benefícios da criação de uma AMI personalizada. Outras distribuições do Linux talvez funcionem com a solução de alguns problemas, mas não são oficialmente compatíveis. Se a aplicação exigir uma distribuição específica do Linux, uma alternativa

será criar uma imagem de Docker e executá-la na plataforma Docker ou na plataforma Docker de vários contêineres do Elastic Beanstalk.

Para criar uma AMI personalizada

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. Escolha Executar instância.
3. Se você identificou uma AMI base do Elastic Beanstalk (usando `describe-platform-version`) ou uma AMI do Amazon Linux, insira o ID da AMI na caixa de pesquisa. Em seguida, pressione Enter.

   Você também pode pesquisar a lista para outra comunidade AMI que atenda às suas necessidades.

   > **Note**
   >
   > Recomendamos que você escolha uma AMI que use a virtualização de HVM. Eles AMIs mostram o tipo de virtualização: hvm em sua descrição.
   > Para obter mais informações, consulte Tipos de virtualização no Guia do EC2 usuário da Amazon.

4. Escolha Select para selecionar a AMI.
5. Selecione um tipo de instância e, a seguir, escolha Próximo: Configurar detalhes da instância.
6. (Para plataformas Amazon Linux AMI (AL1) descontinuadas) Pule esta etapa se seu ambiente for executado em uma plataforma compatível baseada em Linux ou em uma plataforma Windows.

   Expanda a seção Detalhes avançados e cole o seguinte texto no campo Dados do usuário.

   ```
   #cloud-config
     repo_releasever: repository version number
     repo_upgrade: none
   ```

   O número de versão do repositório é a versão de ano e mês no nome da AMI. Por exemplo, AMIs com base na versão de março de 2015 do Amazon Linux, tenha um número `2015.03` de versão do repositório. Para uma imagem do Elastic Beanstalk, isso corresponde à data mostrada no nome da pilha de soluções para a versão da plataforma com base na AMI do Amazon Linux (anterior ao Amazon Linux 2).

> **Note**
>
> A `repo_releasever` configuração configura o lock-on-launch recurso para uma Amazon Linux AMI. Isso faz com que a AMI use uma versão fixa e específica do repositório quando é iniciada. Esse recurso não é compatível com o Amazon Linux 2, ou seja, não o especifique se seu ambiente usa uma ramificação de plataforma atual do Amazon Linux 2. A configuração será necessária se você estiver usando uma AMI personalizada com o Elastic Beanstalk somente nas ramificações da plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2).
> A configuração `repo_upgrade` desativa a instalação automática de atualizações de segurança. É necessário usar uma AMI personalizada com o Elastic Beanstalk.

7. Prossiga com o assistente para iniciar a EC2 instância. Quando solicitado, selecione um par de chaves ao qual tenha acesso para que você possa se conectar à instância para as próximas etapas.
8. Conecte-se à instância com SSH ou RDP.
9. Execute todas as personalizações desejadas.
10. (Plataformas Windows) Execute o serviço EC2 Config Sysprep. Para obter informações sobre o EC2 Config, consulte Como configurar uma instância do Windows usando o Config EC2 Service. Certifique-se de que o Sysprep está configurado para gerar uma senha aleatória que pode ser recuperada no AWS Management Console.
11. No EC2 console da Amazon, interrompa a EC2 instância. Em seguida, no menu Instance Actions (Ações da instância), escolha Create Image (EBS AMI) (Criar imagem (AMI do EBS)).
12. Para evitar AWS cobranças adicionais, encerre a EC2 instância.

Como usar sua AMI personalizada em um ambiente do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Em AMI ID (ID de AMI), insira seu ID de AMI personalizado.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Quando você cria um ambiente com a AMI personalizada, deve usar a mesma versão da plataforma usada como base para criar a AMI.

Gerenciar um ambiente com uma AMI personalizada

Atualizações da plataforma

Ao usar uma AMI personalizada, o Elastic Beanstalk continuará a usar a mesma AMI personalizada em um ambiente quando sua versão da plataforma for atualizada, independentemente de a atualização ser aplicada manualmente ou por meio de atualizações de plataformas gerenciadas. O ambiente não será redefinido para usar a AMI padrão da nova versão da plataforma.

Recomendamos criar uma nova AMI personalizada com base na AMI padrão da nova versão da plataforma. Isso aplicará os patches disponíveis na nova versão da plataforma e também minimizará as falhas de implementação devido a versões incompatíveis de pacotes ou bibliotecas.

Para obter mais informações sobre a criação de uma nova AMI personalizada, consulte o Criar uma AMI do personalizada anteriormente neste tópico.

Remover uma AMI personalizada

Para remover uma AMI personalizada de um ambiente e redefini-la para usar a AMI padrão da versão da plataforma do ambiente, use o seguinte comando da 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**
>
> Para evitar a interrupção do serviço, teste sua aplicação com uma AMI fixa antes de aplicar essa alteração ao seu ambiente de produção.

Limpeza de uma AMI personalizada

Quando tiver concluído com uma AMI personalizada e não precisar mais dela para iniciar ambientes do Elastic Beanstalk, considere limpá-la para minimizar o custo de armazenamento. Limpar uma AMI

personalizada envolve cancelar o registro dela na Amazon EC2 e excluir outros recursos associados. Para obter detalhes, consulte Cancelar o registro da AMI do Linux ou Cancelar o registro da AMI do Windows.

Preservar o acesso a uma imagem de máquina da Amazon (AMI) para uma plataforma descontinuada

O Elastic Beanstalk define o status de uma ramificação da plataforma como descontinuada quando o sistema operacional ou componente principal usado pela ramificação atinge o fim da vida útil. A AMI básica do Elastic Beanstalk para a ramificação da plataforma também pode ser tornada privada para evitar o uso dessa AMI. out-of-date Ambientes AMIs que tenham sido tornados privados não poderão mais iniciar instâncias.

Se você não conseguir migrar sua aplicação para um ambiente compatível antes de ser descontinuada, seu ambiente pode estar nessa situação. Pode ser necessário atualizar um ambiente para uma ramificação da plataforma Beanstalk, onde sua AMI básica do Elastic Beanstalk foi alterada para privada. Há uma abordagem alternativa. Você pode atualizar um ambiente existente com base em uma cópia da AMI básica do Elastic Beanstalk usada pelo seu ambiente.

Este tópico apresenta algumas etapas e um script independente para atualizar um ambiente existente com base em uma cópia da AMI básica do Elastic Beanstalk usada pelo seu ambiente. Depois de migrar sua aplicação para uma plataforma compatível, você pode continuar usando os procedimentos padrão para manter sua aplicação e ambientes compatíveis.

Etapas manuais

Para atualizar um ambiente com base em uma cópia da AMI básica do Elastic Beanstalk

1. Determine qual AMI seu ambiente está usando. Este comando retorna a AMI usada pelo ambiente do Elastic Beanstalk que você fornece nos parâmetros. O valor retornado é usado como source-ami-idna próxima etapa.

   Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte describe-configuration-settings na Referência de comandos da AWS CLI .

   Especifique a AWS região que armazena a AMI de origem que você deseja copiar. Substitua o nome da aplicação e o nome do ambiente por aqueles baseados na AMI de origem. Insira o texto para o parâmetro de consulta, conforme mostrado.

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. Copie a AMI em sua conta. Esse comando retorna a nova AMI resultante da cópia da source-ami-idque foi retornada na etapa anterior.

> **Note**
>
> Não se esqueça de anotar o novo ID da AMI que é gerado por esse comando. Você precisará inseri-lo na próxima etapa, substituindo-o copied-ami-idno comando de exemplo.

Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte copy-image na AWS CLI Command Reference.

Especifique a AWS região da AMI de origem que você deseja copiar (--source-region) e a região em que você deseja usar sua nova AMI personalizada (--region). source-ami-id Substitua pela AMI da imagem que você está copiando. O source-ami-idfoi retornado pelo comando na etapa anterior. new-ami-nameSubstitua por um nome para descrever a nova AMI na região de destino. O script que segue esse procedimento gera o novo nome da AMI anexando a string "Copy of" ao início do nome dosource-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. Atualize o ambiente para usar a AMI copiada. Depois de executar o comando, ele retornará o status do ambiente.

Em uma janela de comando, execute um comando semelhante ao seguinte. Para obter mais informações, consulte update-environment na Referência de comandos da AWS CLI .

Especifique a AWS região do ambiente e do aplicativo que você precisa atualizar. Substitua o nome do aplicativo e o nome do ambiente pelos que você precisa associar à etapa anterior. copied-ami-id Para o parâmetro --option-setttings, *copied-ami-id* substitua pelo ID da AMI que você anotou na saída do comando anterior.

```
>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**
>
> Para minimizar o custo de armazenamento, considere limpar a AMI personalizada quando não precisar mais dela para iniciar ambientes do Elastic Beanstalk. Para obter mais informações, consulte Limpeza de uma AMI personalizada.

Script autônomo

O script a seguir fornece os mesmos resultados das etapas manuais anteriores. Faça o download do script selecionando este link: copy_ami_and_update_env.zip.

Fonte do 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**
>
> Você deve ter o AWS CLI instalado para executar o script. Para obter instruções de atualização, consulte Instalar ou atualizar a versão mais recente da AWS CLI no Guia do usuário do AWS Command Line Interface .
> Depois de instalar o AWS CLI, você também deve configurá-lo para usar a AWS conta proprietária do ambiente. Para obter mais informações, consulte Configurar a AWS CLI no Guia do usuário da AWS Command Line Interface . A conta também deve ter permissões para criar uma AMI e atualizar o ambiente do Elastic Beanstalk.

Estas etapas descrevem o processo que o script segue.

1. Imprima a conta em uso.
2. Determine qual AMI é usada pelo ambiente (AMI de origem).
3. Verifique se a AMI de origem já pertence à conta. Se sim, saia.
4. Determine o nome da AMI de origem para que ela possa ser usada no novo nome da AMI. Isso também serve para confirmar o acesso à AMI de origem.
5. Verifique se a AMI de origem já foi copiada para a conta. Isso é feito pesquisando AMIs com o nome da AMI copiada de propriedade da conta. Se o nome da AMI tiver sido alterado entre as execuções do script, ela copiará a imagem novamente.
6. Se a AMI de origem ainda não tiver sido copiada, copie a AMI de origem para a conta e aguarde até que a nova AMI esteja disponível.
7. Atualize a configuração do ambiente para usar a nova AMI.
8. Aguarde a conclusão da atualização do ambiente.

Depois de extrair o script do arquivo copy_ami_and_update_env.zip, execute-o conforme o exemplo a seguir. Substitua o nome da aplicação e o nome do ambiente no exemplo por seus valores.

```
>sh copy_ami_and_update_env.sh \
  --application-name my-application \
  --environment-name my-environment \
  --region us-east-1
```

> **Note**
>
> Para minimizar o custo de armazenamento, considere limpar a AMI personalizada quando não precisar mais dela para iniciar ambientes do Elastic Beanstalk. Para obter mais informações, consulte Limpeza de uma AMI personalizada.

Fornecer arquivos estáticos

Para melhorar o desempenho, você pode configurar o servidor de proxy para fornecer arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro de seu aplicativo web. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

O Elastic Beanstalk é compatível com a configuração do proxy para atender a arquivos estáticos na maioria das ramificações da plataforma baseadas no Amazon Linux 2. A única exceção é o Docker.

> **Note**
>
> Nas plataformas Python e Ruby, o Elastic Beanstalk configura algumas pastas de arquivos estáticos por padrão. Para obter detalhes, consulte as seções de configuração de arquivo estático para Python e Ruby. É possível configurar pastas adicionais conforme explicado nesta página.

Configurar arquivos estáticos usando o console

Para configurar o servidor de proxy para fornecer arquivos estáticos

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Vá até a seção Platform software (Software da plataforma) e localize o grupo Static files (Arquivos estáticos).

    a. Para adicionar um mapeamento de arquivo estático, selecione Add static files (Adicionar arquivos estáticos). Na linha extra que aparece, você inserirá um caminho para servir arquivos estáticos e o diretório que contém os arquivos estáticos a serem servidos.

    - No campo Path (Caminho), inicie o nome do caminho com uma barra (/) (por exemplo, "/images “).
    - No campo Directory (Diretório), especifique um nome de diretório localizado na raiz do código-fonte do seu aplicativo. Não comece com uma barra (por exemplo, "static/image-files“).

> **Note**
>
> Se a seção Static files (Arquivos estáticos) não for exibida, adicione pelo menos um mapeamento usando um arquivo de configuração. Para obter detalhes, consulte the section called “Configurar arquivos estáticos usando opções de configuração” nesta página.

b. Para remover um mapeamento, selecione Remove (Remover).

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar arquivos estáticos usando opções de configuração

É possível usar um arquivo de configuração para configurar caminhos de arquivos estáticos e locais de diretório usando opções de configuração. É possível adicionar um arquivo de configuração ao pacote de origem do aplicativo e implantá-lo durante a criação do ambiente ou em uma implantação posterior.

Se o seu ambiente usar uma ramificação da plataforma baseada no Amazon Linux 2, use o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`.

O exemplo de arquivo de configuração a seguir informa ao servidor de proxy para servir arquivos na pasta `statichtml` no caminho `/html`, e arquivos na pasta `staticimages` no caminho `/images`.

Example .ebextensions/static-files.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:staticfiles:
    /html: statichtml
    /images: staticimages
```

Se o seu ambiente do Elastic Beanstalk usar uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as seguintes informações adicionais:

Namespaces específicos da plataforma da AMI do Amazon Linux

Nas ramificações da plataforma da AMI do Amazon Linux, os namespaces de configuração de arquivos estáticos variam de acordo com a plataforma. Para obter detalhes, consulte uma das seguintes páginas:

- Namespace de configuração do Go
- Namespace de configuração do Java SE
- Namespaces de configuração do Tomcat
- Namespace de configuração do Node.js
- Namespaces de configuração do Python

Configurar HTTPS para o seu ambiente do Elastic Beanstalk

Os tópicos desta seção explicam como configurar o HTTPS para o ambiente Elastic Beanstalk. O HTTPS é essencial para qualquer aplicativo que transmite dados de usuário ou informações de login.

Se você tiver adquirido e configurado um nome de domínio personalizado para o seu ambiente Elastic Beanstalk, poderá usar HTTPS para permitir que os usuários se conectem ao seu site com segurança.

Se você não tiver um nome de domínio, ainda poderá usar HTTPS com um certificado autoassinado para fins de desenvolvimento e teste. Para obter mais informações, consulte Certificados de servidor.

Configurar a terminação HTTPS no balanceador de carga

Um balanceador de carga distribui solicitações para as EC2 instâncias que executam seu aplicativo. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet. A maneira mais simples de usar o HTTPS em um ambiente de várias instâncias do Elastic Beanstalk é configurar um receptor seguro para o balanceador de carga. A conexão entre o cliente e o balanceador de carga permanece segura, então você pode configurar o balanceador de carga para encerrar HTTPS. As conexões de back-end entre o balanceador de carga e EC2 as instâncias usam HTTP, portanto, nenhuma configuração adicional das instâncias é necessária. Para obter instruções detalhadas sobre como configurar um receptor seguro, consulte Configurar a terminação HTTPS no balanceador de carga.

Configurando a terminação de HTTPS na instância EC2

Se você executa seu aplicativo em um ambiente de única instância ou precisa proteger a conexão até as EC2 instâncias por trás do balanceador de carga, você pode configurar o servidor proxy executado na instância para encerrar o HTTPS. Configurar suas instâncias para encerrar as conexões HTTPS exige o uso de arquivos de configuração para modificar o software em execução nas instâncias e para modificar os grupos de segurança para permitir conexões seguras. Para obter mais informações, consulte Configuração da terminação de HTTPS na instância.

Configurando HTTPS end-to-end

Para end-to-end HTTPS em um ambiente com balanceamento de carga, você pode combinar o encerramento da instância e do balanceador de carga para criptografar as duas conexões. Por padrão, se você configurar o load balancer para encaminhar o tráfego usando HTTPS, ele confiará em qualquer certificado apresentado a ele pelas instâncias de back-end. Para obter o máximo de segurança, você pode anexar políticas ao load balancer que impedem que ele se conecte a instâncias que não apresentarem um certificado público em que ele confia. Para obter mais informações, consulte Configurando a end-to-end criptografia em um ambiente Elastic Beanstalk com balanceamento de carga.

Configuração HTTPS com TCP Passthrough

Você também pode configurar o load balancer para retransmitir o tráfego HTTPS sem descriptografá-lo. Para obter mais informações, consulte Configurar o load balancer do ambiente para passagem de TCP.

> **Note**
>
> O Tem cobras? O aplicativo de amostra em GitHub inclui arquivos de configuração e instruções para cada método de configuração de HTTPS com um aplicativo web Tomcat. Consulte o arquivo readme e as instruções de HTTPS para obter mais detalhes.

Tópicos

- Certificados de servidor
- Configurar a terminação HTTPS no balanceador de carga
- Configuração da terminação de HTTPS na instância
- Configurando a end-to-end criptografia em um ambiente Elastic Beanstalk com balanceamento de carga
- Configurar o load balancer do ambiente para passagem de TCP
- Configurar o redirecionamento de HTTP para HTTPS

Certificados de servidor

Este tópico descreve os diferentes tipos de certificados que você pode usar para configurar o HTTPS e quando aplicar cada um deles. Os subtópicos desta seção fornecem instruções para criar seu próprio certificado e como carregá-lo.

AWS Certificate Manager (ACM)

O ACM é a ferramenta preferencial para provisionar, gerenciar e implantar seus certificados de servidor. Você pode fazer isso programaticamente ou usando o. AWS CLI Com o ACM, você pode criar um certificado confiável para seus nomes de domínio gratuitamente.

Os certificados ACM só podem ser usados com AWS balanceadores de carga e CloudFront distribuições da Amazon, e o ACM está disponível somente em determinadas regiões. AWS Para usar um certificado do ACM com o Elastic Beanstalk, consulte Configurar a terminação HTTPS no balanceador de carga. Para obter mais informações sobre o ACM, consulte o Guia do usuário do AWS Certificate Manager .

> **Note**
>
> Para obter uma lista das regiões em que o ACM está disponível, consulte Endpoints e cotas do ACM na seção Referência geral da Amazon Web Services.

Se o ACM não estiver disponível na sua AWS região, você poderá carregar um certificado de terceiros ou autoassinado e uma chave privada para AWS Identity and Access Management (IAM). Você pode usar o AWS CLI para carregar o certificado. Os certificados armazenados no IAM podem ser usados com balanceadores de carga e CloudFront distribuições. Para obter mais informações, consulte Upload de um certificado no IAM.

Certificado de terceiros

Se ACM não estiver disponível em sua região, você pode comprar um certificado confiável de terceiros. Um certificado de terceiros pode ser usado para descriptografar tráfego HTTPS no load balancer, nas instâncias de back-end ou em ambos.

Certificado autoassinado

Para desenvolvimento e teste, você pode criar e assinar um certificado por conta própria com ferramentas de código aberto. Os certificados autoassinados são gratuitos e fáceis de criar, mas

não podem ser usados para a descriptografia de front-end em sites públicos. Se você tentar usar um certificado autoassinado para uma conexão HTTPS com um cliente, o navegador do usuário exibirá uma mensagem de erro indicando que o seu site não é seguro. No entanto, você pode usar um certificado autoassinado para proteger as conexões de back-end sem problemas.

Criar e acessar um certificado X509

Você pode criar um certificado X509 para seu aplicativo com `OpenSSL`. OpenSSL é uma biblioteca de código aberto padrão que oferece suporte a uma grande variedade de funções de criptografia, incluindo a criação e a assinatura de certificados x509. Para obter mais informações sobre OpenSSL, visite www.openssl.org.

> **Note**
>
> Só será necessário criar um certificado localmente se você quiser usar HTTPS em um ambiente de instância única ou criptografar novamente no back-end com um certificado autoassinado. Se você possui um nome de domínio, pode criar um certificado AWS e usá-lo com um ambiente de balanceamento de carga gratuitamente usando AWS Certificate Manager (ACM). Consulte Solicite um certificado noAWS Certificate Manager Guia do usuário para obter instruções.

Execute `openssl version` na linha de comando para verificar se o OpenSSL já está instalado. Caso contrário, você pode criar e instalar o código-fonte usando as instruções no GitHub repositório público ou usar seu gerenciador de pacotes favorito. O OpenSSL também está instalado nas imagens Linux do Elastic Beanstalk, então uma alternativa rápida é conectar-se a uma instância em EC2 um ambiente em execução usando o comando da EB CLI: eb ssh

```
~/eb$ eb ssh
[ec2-user@ip-255-55-55-255 ~]$ openssl version
OpenSSL 1.0.1k-fips 8 Jan 2015
```

Você precisa criar uma chave privada RSA para criar sua solicitação de assinatura de certificado (CSR). Para criar sua chave privada, use o 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

O nome do arquivo onde deseja salvar a chave privada. Em geral, o comando openssl genrsa imprime o conteúdo da chave privada na tela, mas esse comando envia a saída para um arquivo. Escolha qualquer nome de arquivo e armazene o arquivo em um local seguro, de modo que possa recuperá-lo mais tarde. Se você perder a sua chave privada, não poderá usar seu certificado.

Um CSR é um arquivo que você envia a uma autoridade de certificação (CA) para solicitar um certificado de servidor digital. Para criar um CSR, use o 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.
```

Insira as informações solicitadas e pressione Enter. A tabela a seguir descreve e mostra exemplos para cada campo.

Nome	Descrição	Exemplo
Nome do país	A abreviação ISO de duas letras para seu país.	US = Estados Unidos
Estado	O nome do estado ou província onde sua organização está localizada. Você não pode abreviar esse nome.	Washington
Nome da localidade	O nome da cidade onde sua organização está localizada.	Seattle
Nome da organização	A razão social completa da sua organização. Não abrevie o nome de sua organização.	Corporação de exemplo

Nome	Descrição	Exemplo
Unidade Organizacional	Opcional, para fornecer informações adicionais sobre sua organização.	Marketing
Nome comum	O nome de domínio totalmente qualificado para o seu site. Isso deve corresponder ao nome de domínio que os usuários veem quando visitam seu site; caso contrário, serão mostrados erros de certificado.	www.exemplo.com
Endereço de e-mail	O endereço de e-mail do administrador do site.	someone@example.com

Você pode enviar a solicitação de assinatura para terceiros assinarem ou assine-a você mesmo para desenvolvimento e teste. Certificados autoassinados também podem ser usados para HTTPS de back-end entre um balanceador de carga e instâncias. EC2

Para assinar o certificado, use o comando openssl x509. O exemplo a seguir usa a chave privada da etapa anterior (*privatekey.pem*) e a solicitação de assinatura (*csr.pem*) para criar um certificado público chamado *public.crt* válido por *365* dias.

```
$ 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
```

Guarde a chave privada e o certificado público para uso posterior. Você pode descartar a solicitação de assinatura. Sempre armazene a chave privada em um local seguro e evite adicioná-la ao seu código-fonte.

Para usar o certificado com a plataforma Windows Server, converta-o em um formato PFX. Use o comando a seguir para criar um certificado PFX da chave privada e dos arquivos públicos do certificado:

```
$ openssl pkcs12 -export -out example.com.pfx -inkey privatekey.pem -in public.crt
Enter Export Password: password
```

```
Verifying - Enter Export Password: password
```

Agora que você tem um certificado, pode fazer upload para o IAM para uso com um load balancer ou configurar as instâncias em seu ambiente para encerrar o HTTPS.

Upload de um certificado no IAM

Para usar seu certificado com o balanceador de carga do seu ambiente Elastic Beanstalk, faça o upload do certificado e da chave privada para (IAM). AWS Identity and Access Management Você pode usar um certificado armazenado no IAM com balanceadores de carga do Elastic Load Balancing e distribuições da Amazon. CloudFront

> **Note**
>
> AWS Certificate Manager (ACM) é a ferramenta preferida para provisionar, gerenciar e implantar seus certificados de servidor. Para obter mais informações sobre como solicitar um certificado do ACM, consulte Solicitação de um certificado no Manual do usuário do AWS Certificate Manager . Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte Importação de certificados no Manual do usuário do AWS Certificate Manager . Use o IAM para fazer upload de um certificado somente se o ACM não estiver disponível na sua AWS região.

Você pode usar o AWS Command Line Interface (AWS CLI) para carregar seu certificado. O comando a seguir carrega um certificado autoassinado chamado *https-cert.crt* com uma chave privada chamada: *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"
    }
}
```

O `file://` prefixo diz ao AWS CLI para carregar o conteúdo de um arquivo no diretório atual. *`elastic-beanstalk-x509`*especifica o nome para chamar o certificado no IAM.

Se você adquiriu um certificado de uma autoridade de certificação e recebeu um arquivo de cadeia de certificados, faça upload dele incluindo a opção `--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
```

Anote o Nome de recurso da Amazon (ARN) do seu certificado. Você o usará quando atualizar as definições de configuração do load balancer para usar HTTPS.

> **Note**
>
> Um certificado carregado para o IAM permanece armazenado mesmo depois de deixar de ser usado em qualquer balanceador de carga do ambiente. Ele contém dados confidenciais. Quando não precisar mais do certificado para qualquer ambiente, lembre-se de excluí-lo. Para obter detalhes sobre como excluir um certificado do IAM, consulte https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html#delete-server-certificate.

Para obter mais informações sobre certificados de servidor no IAM, consulte Trabalhar com certificados de servidor no Guia do usuário do IAM.

Armazenar chaves privadas com segurança no Amazon S3

A chave privada que você usa para assinar seu certificado público é privada e não deve ser submetida ao código-fonte. Evite armazenar chaves privadas em arquivos de configuração carregando-as no Amazon S3 e configurando o Elastic Beanstalk para fazer download do arquivo do Amazon S3 durante a implantação da aplicação.

O exemplo a seguir mostra as seções Recursos e arquivos de um arquivo de configuração que faz download do arquivo de chave privada de um bucket do 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
```

Substitua o nome do bucket e o URL no exemplo pelos seus. A primeira entrada desse arquivo adiciona um método de autenticação chamado `S3Auth` aos metadados do grupo de Auto Scaling do ambiente. Se você tiver configurado um perfil de instância personalizado para o ambiente a ser usado, caso contrário, será aplicado o valor padrão `aws-elasticbeanstalk-ec2-role`. O perfil de instância padrão tem permissão de leitura do bucket de armazenamento do Elastic Beanstalk. Se você usar um outro bucket, adicione permissões ao perfil de instância.

A segunda entrada usa o método de autenticação `S3Auth` para fazer download da chave privada do URL especificado e salvá-la no `/etc/pki/tls/certs/server.key`. O servidor de proxy pode ler a chave privada a partir desse local para encerrar conexões HTTPS na instância.

O perfil da instância atribuído às EC2 instâncias do seu ambiente precisa ter permissão para ler o objeto chave do bucket especificado. Verifique se o perfil de instância tem permissão para ler o objeto no IAM e se as permissões no bucket e objeto não proíbem o perfil de instância.

Para visualizar as permissões de um bucket

1. Abra o Console de gerenciamento do Amazon S3.
2. Escolha um bucket.
3. Escolha Propriedades e, em seguida, Permissões.

4. Verifique se sua conta tem permissão de leitura no bucket.
5. Se uma política de bucket estiver anexada, escolha Bucket policy para visualizar as permissões atribuídas ao bucket.

Configurar a terminação HTTPS no balanceador de carga

Para atualizar seu AWS Elastic Beanstalk ambiente para usar HTTPS, você precisa configurar um ouvinte HTTPS para o balanceador de carga em seu ambiente. Dois tipos de balanceador de carga são compatíveis com um listener HTTPS: Classic Load Balancer e Application Load Balancer.

É possível usar o console do Elastic Beanstalk ou um arquivo de configuração para configurar um listener seguro e atribuir o certificado.

> **Note**
>
> Ambientes de instância única não têm um load balancer e não são compatíveis com o encerramento do HTTPS no load balancer.

Configurar um listener seguro usando o console do Elastic Beanstalk

Como atribuir um certificado ao load balancer do seu ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Load balancer, escolha Edit (Editar).

   > **Note**
   >
   > Se a categoria de configuração Load balancer não tem um botão Edit (Editar), seu ambiente não tem um load balancer.

5. Na página Modify load balancer (Modificar load balancer), o procedimento varia dependendo do tipo de load balancer associado ao seu ambiente.
   - Classic Load Balancer

     a. Escolha Add listener.

b. Na caixa de diálogo Classic Load Balancer listener (Listener do Classic Load Balancer), defina as seguintes configurações:

   - Para Listener port, digite a porta de tráfego de entrada, normalmente `443`.
   - Para Listener protocol, escolha HTTPS.
   - Para Instance port, digite `80`.
   - Para Instance protocol, escolha HTTP.
   - Para SSL certificate, choose your certificate.

c. Escolha Adicionar.

- Application Load Balancer

a. Escolha Add listener.

b. Na caixa de diálogo Application Load Balancer listener (Listener do Application Load Balancer), defina as seguintes configurações:

   - Para Port, digite a porta de tráfego de entrada, normalmente `443`.
   - Em Protocol, escolha HTTPS.
   - Para SSL certificate, choose your certificate.

c. Escolha Adicionar.

> **Note**
>
> Para Classic Load Balancer e Application Load Balancer, se o menu suspenso não mostrar nenhum certificado, será necessário criar ou carregar um certificado para seu nome de domínio personalizado no AWS Certificate Manager (ACM) (preferencial). Como alternativa, faça upload de um certificado para o IAM com a AWS CLI.

- Network Load Balancer

a. Escolha Add listener.

b. Na caixa de diálogo Network Load Balancer listener (Listener do Network Load Balancer) em Port (Porta), digite o número da porta de tráfego de entrada que, normalmente, é `443`.

c. Escolha Adicionar.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar um listener seguro usando um arquivo de configuração

É possível configurar um listener seguro em seu load balancer com um dos seguintes arquivos de configuração.

Example .ebextensions/securelistener-clb.config

Use este exemplo quando seu ambiente tiver um Classic Load Balancer. O exemplo usa as opções no namespace do `aws:elb:listener` para configurar um listener HTTPS na porta 443 com o certificado especificado e encaminhar o tráfego descriptografado para as instâncias em seu ambiente na porta 80.

```
option_settings:
  aws:elb:listener:443:
    SSLCertificateId: arn:aws:acm:us-east-2:1234567890123:certificate/
###################################
    ListenerProtocol: HTTPS
    InstancePort: 80
```

Substitua o texto destacado pelo ARN do seu certificado. O certificado pode ser aquele que você criou ou carregou no AWS Certificate Manager (ACM) (preferencial) ou um que você enviou para o IAM com o. AWS CLI

Para obter mais informações sobre as opções de configuração do Classic Load Balancer, consulte Namespaces de configuração do Classic Load Balancer.

Example .ebextensions/securelistener-alb.config

Use este exemplo quando seu ambiente tiver um Application Load Balancer. O exemplo usa opções no namespace `aws:elbv2:listener` configurar um listener HTTPS na porta 443 com o certificado especificado. O listener direciona o tráfego para o processo padrão.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
    Protocol: HTTPS
    SSLCertificateArns: arn:aws:acm:us-east-2:1234567890123:certificate/
###################################
```

Example .ebextensions/securelistener-nlb.config

Use este exemplo quando seu ambiente tiver um Network Load Balancer. O exemplo usa as opções no namespace `aws:elbv2:listener` para configurar um listener na porta 443. O listener direciona o tráfego para o processo padrão.

```
option_settings:
  aws:elbv2:listener:443:
    ListenerEnabled: 'true'
```

Configurar um grupo de segurança

Se você configurar o load balancer para encaminhar o tráfego para uma porta de instância diferente da porta 80, será necessário adicionar uma regra para o security group que permita o tráfego de entrada vindo de seu load balancer pela porta de instância configurada. Se você criar seu ambiente em uma VPC personalizada, o Elastic Beanstalk adicionará essa regra para você.

Você adiciona essa regra ao inserir uma chave `Resources` em um arquivo de configuração no diretório `.ebextensions` para seu aplicativo.

O arquivo de configuração de exemplo a seguir adiciona uma regra de entrada no security group `AWSEBSecurityGroup`. Isso permite tráfego na porta 1000 do security group do load balancer.

Example .ebextensions/sg-ingressfromlb.config

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 1000
      FromPort: 1000
      SourceSecurityGroupId: {"Fn::GetAtt" : ["AWSEBLoadBalancerSecurityGroup",
 "GroupId"]}
```

Configuração da terminação de HTTPS na instância

Use arquivos de configuração para configurar o servidor de proxy que passa o tráfego para o aplicativo para encerrar conexões HTTPS. Isso é útil se você quer usar o HTTPS com um ambiente

de uma única instância, ou se configurar o load balancer para passar o tráfego através sem descriptografá-lo.

Para habilitar o HTTPS, você deve permitir o tráfego de entrada na porta 443 para a EC2 instância em que seu aplicativo Elastic Beanstalk está sendo executado. Você faz isso usando a `Resources` chave no arquivo de configuração para adicionar uma regra para a porta 443 às regras de entrada do grupo de segurança do AWSEBSecurity Grupo.

O seguinte snippet adiciona uma regra de entrada para o security group `AWSEBSecurityGroup` que abre a porta 443 para todo o tráfego de um ambiente de uma única instância:

`.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
```

Em um ambiente com balanceamento de carga em um Amazon Virtual Private Cloud (Amazon VPC) padrão, é possível modificar essa política para aceitar apenas o tráfego do balanceador de carga. Consulte Configurando a end-to-end criptografia em um ambiente Elastic Beanstalk com balanceamento de carga para ver um exemplo.

Plataformas

- Encerramento de HTTPS em EC2 instâncias que executam o Docker
- Encerramento de HTTPS em EC2 instâncias que executam Go
- Encerramento de HTTPS em EC2 instâncias que executam Java SE
- Encerramento de HTTPS em EC2 instâncias que executam o Node.js
- Encerramento de HTTPS em EC2 instâncias que executam PHP
- Encerramento de HTTPS em EC2 instâncias que executam Python
- Encerramento de HTTPS em EC2 instâncias que executam Ruby
- Encerramento de HTTPS em EC2 instâncias que executam o Tomcat

- Encerramento de HTTPS em EC2 instâncias da Amazon que executam o.NET Core no Linux
- Encerramento de HTTPS em EC2 instâncias da Amazon que executam o.NET

Encerramento de HTTPS em EC2 instâncias que executam o Docker

Para os contêineres do Docker, use um arquivo de configuração para habilitar o HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

  `/etc/nginx/conf.d/https.conf`

  Configura o servidor nginx. Este arquivo é carregado quando o serviço do nginx é iniciado.

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

  > **Note**
  >
  > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

  Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

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**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam Go

Para os tipos de contêiner Go, você habilita o HTTPS com um arquivo de configuração e um arquivo de configuração nginx que configura o servidor nginx para usar HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo os espaços reservados do certificado e da chave privada como instruído, e salve-o no diretório `.ebextensions` do pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `Resources` habilita a porta 443 no security group usado pela instância do seu ambiente.
- A chave `files` cria os seguintes arquivos na instância:

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

  > **Note**
  >
  > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

  Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reinicia o servidor nginx depois que tudo é configurado para o servidor carregar o arquivo de configuração nginx.

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-----

container_commands:
  01restart_nginx:
    command: "service nginx restart"
```

> **Note**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Insira o seguinte em um arquivo com a extensão `.conf` no diretório `.ebextensions/nginx/conf.d/` do seu pacote de origem (ex. `.ebextensions/nginx/conf.d/https.conf`). *`app_port`*Substitua pelo número da porta que seu aplicativo escuta. Este exemplo configura o

servidor nginx para escutar na porta 443 usando SSL. Para obter mais informações sobre esses arquivos de configuração na plataforma Go, consulte Configurar o servidor de 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;
    }
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
```

```
    IpProtocol: tcp
    ToPort: 443
    FromPort: 443
    CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam Java SE

Para os tipos de contêiner Java SE, você habilita o HTTPS com um arquivo de configuração .ebextensions e um arquivo de configuração nginx que configura o servidor nginx para usar HTTPS.

Todas as AL2 plataformas AL2 023/ suportam um recurso de configuração de proxy uniforme. Para obter mais informações sobre como configurar o servidor proxy nas versões da sua plataforma executando AL2 023/AL2, consulte. Configuração de proxy reverso

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo os espaços reservados do certificado e da chave privada como instruído, e salve-o no diretório `.ebextensions`. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

  > **Note**
  >
  > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

  Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reinicia o servidor nginx depois que tudo é configurado para o servidor carregar o arquivo de configuração nginx.

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-----

container_commands:
  01restart_nginx:
    command: "service nginx restart"
```

> **Note**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para

fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Insira o seguinte em um arquivo com a extensão `.conf` no diretório `.ebextensions/nginx/conf.d/` do seu pacote de origem (ex. `.ebextensions/nginx/conf.d/https.conf`). *`app_port`*Substitua pelo número da porta que seu aplicativo escuta. Este exemplo configura o servidor nginx para escutar na porta 443 usando SSL. Para obter mais informações sobre esses arquivos de configuração na plataforma Java SE, consulte Configurar o servidor de 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;
    }
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam o Node.js

O arquivo de configuração de exemplo a seguir amplia a configuração padrão do nginx para escutar na porta 443 e encerrar conexões SSL/TLS com um certificado público e uma chave privada.

Se você configurou seu ambiente para relatórios de saúde aprimorados, precisará configurar o nginx para gerar logs de acesso. Para fazer isso, apague a escrita do bloco de linhas sob o comentário `# For enhanced health...` removendo o # inicial.

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-----
```

A chave `files` cria os seguintes arquivos na instância:

`/etc/nginx/conf.d/https.conf`

Configura o servidor nginx. Este arquivo é carregado quando o serviço do nginx é iniciado.

`/etc/pki/tls/certs/server.crt`

Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

> **Note**
>
> YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

> **Note**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está

funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam PHP

Para tipos de contêiner PHP, você usa um arquivo de configuração para habilitar o Apache HTTP Server a usar HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem.

O arquivo de configuração executa as seguintes tarefas:

- A chave `packages` usa o yum para instalar `mod24_ssl`.
- A chave `files` cria os seguintes arquivos na instância:

  `/etc/httpd/conf.d/ssl.conf`

  Configura o servidor Apache. Este arquivo é carregado quando o serviço do Apache é iniciado.

`/etc/pki/tls/certs/server.crt`

Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

> **Note**
>
> YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

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**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam Python

Para os tipos de contêiner Python que usam Apache HTTP Server com a Web Server Gateway Interface (WSGI), você usa um arquivo de configuração para habilitar o Apache HTTP Server a usar o HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `packages` usa o yum para instalar `mod24_ssl`.
- A chave `files` cria os seguintes arquivos na instância:

  `/etc/httpd/conf.d/ssl.conf`

  Configura o servidor Apache. Se o seu aplicativo não estiver nomeado como `application.py`, substitua o texto destacado no valor para `WSGIScriptAlias` pelo caminho local para seu aplicativo. Por exemplo, um aplicativo django pode estar em `django/wsgi.py`. A localização deve corresponder ao valor da opção `WSGIPath` que você definiu para seu ambiente.

  De acordo com os requisitos do seu aplicativo, você também pode precisar adicionar outros diretórios ao parâmetro python-path.

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

  > **Note**
  >
  > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

  Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` interrompe o serviço httpd após tudo estar configurado, para que ele use o novo certificado e o arquivo `https.conf`.

> **Note**
>
> O exemplo funciona apenas em ambientes usando a Plataforma Python.

Example .ebextensions/https-instance.config

```
packages:
  yum:
    mod24_ssl : []

files:
  /etc/httpd/conf.d/ssl.conf:
    mode: "000644"
    owner: root
    group: root
    content: |
      LoadModule wsgi_module modules/mod_wsgi.so
      WSGIPythonHome /opt/python/run/baselinenv
      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/ /opt/python/current/app/static/
        <Directory /opt/python/current/app/static>
        Order allow,deny
        Allow from all
```

```
        </Directory>

        WSGIScriptAlias / /opt/python/current/app/application.py

        <Directory /opt/python/current/app>
        Require all granted
        </Directory>

        WSGIDaemonProcess wsgi-ssl processes=1 threads=15 display-name=%{GROUP} \
          python-path=/opt/python/current/app \
          python-home=/opt/python/run/venv \
          home=/opt/python/current/app \
          user=wsgi \
          group=wsgi
        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**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam Ruby

Para tipos de contêiner do Ruby, o modo como o HTTPS é habilitado depende do tipo de servidor de aplicativos utilizado.

Tópicos

- Configurar HTTPS para Ruby com Puma
- Configurar o HTTPS para Ruby com o Passenger

Configurar HTTPS para Ruby com Puma

Para tipos de contêiner do Ruby que usam o Puma como servidor de aplicativos, você pode usar um arquivo de configuração para ativar o HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

  `/etc/nginx/conf.d/https.conf`

  Configura o servidor nginx. Este arquivo é carregado quando o serviço do nginx é iniciado.

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

  > **Note**
  >
  > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

  Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reiniciará o servidor nginx depois que tudo estiver configurado, assim o servidor usará o novo arquivo `https.conf`.

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-----

container_commands:
  01restart_nginx:
    command: "service nginx restart"
```

> **Note**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Configurar o HTTPS para Ruby com o Passenger

Para tipos de contêiner do Ruby que usam o Passenger como servidor de aplicativos, você pode usar um arquivo de configuração e um arquivo JSON para ativar o HTTPS.

Para configurar o HTTPS para Ruby com o Passenger

1. Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

   - A chave `files` cria os seguintes arquivos na instância:

     `/etc/pki/tls/certs/server.crt`

     Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

     > **Note**
     >
     > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

Example Snippet .Ebextensions para configuração do HTTPS para Ruby com 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**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela

está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

2. Crie um arquivo de texto e adicione o JSON a seguir ao arquivo. Salve-o no diretório raiz do pacote de origem com o nome `passenger-standalone.json`. Esse arquivo JSON configura o Passenger para usar HTTPS.

> **Important**
>
> Esse arquivo JSON não deve conter a marca de ordem de bytes (BOM). Se a resposta for sim, a biblioteca JSON do Passenger não lerá o arquivo corretamente e o serviço Passenger não será iniciado.

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"
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias que executam o Tomcat

Para os tipos de contêiner Tomcat, você usa um arquivo de configuração para habilitar o Apache HTTP Server a usar HTTPS ao atuar como proxy reverso para o Tomcat.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo o material de certificado e de chave privada como instruído, e salve-o no diretório `.ebextensions` do seu pacote de origem. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

  > **Note**
  >
  > YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

  `/etc/pki/tls/certs/server.key`

  Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

  `/opt/elasticbeanstalk/hooks/appdeploy/post/99_start_httpd.sh`

  Cria um gancho de implantação pós-script para reiniciar o serviço 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
```

Você também precisa configurar o servidor de proxy do ambiente para escutar na porta 443. A configuração do Apache 2.4 a seguir adiciona um ouvinte na porta 443. Para saber mais, consulte Configurar o servidor de 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>
```

O fornecedor do seu certificado pode incluir certificados intermediários que você pode instalar para melhorar a compatibilidade com clientes móveis. Configure o Apache com um pacote de certificate authority (CA – autoridade de certificação) intermediária adicionando o seguinte ao arquivo de configuração SSL (consulte Estendendo e substituindo a configuração padrão do Apache — Amazon Linux AMI () AL1 para saber o local):

- No conteúdo do arquivo `ssl.conf`, especifique o arquivo de cadeia:

```
SSLCertificateKeyFile "/etc/pki/tls/certs/server.key"
SSLCertificateChainFile "/etc/pki/tls/certs/gd_bundle.crt"
SSLCipherSuite        EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

- Adicione uma nova entrada à chave `files` com o conteúdo dos certificados intermediários:

```
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**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para

fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias da Amazon que executam o.NET Core no Linux

Para os tipos de contêiner .NET Core no Linux, habilite o HTTPS com um `.ebextensions` arquivo de configuração e um arquivo de configuração nginx que configura o servidor nginx para usar HTTPS.

Adicione o seguinte trecho ao seu arquivo de configuração, substituindo os espaços reservados do certificado e da chave privada como instruído, e salve-o no diretório `.ebextensions`. O arquivo de configuração executa as seguintes tarefas:

- A chave `files` cria os seguintes arquivos na instância:

  `/etc/pki/tls/certs/server.crt`

  Cria o arquivo de certificado na instância. *`certificate file contents`*Substitua pelo conteúdo do seu certificado.

> **Note**
>
> YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

Se você tiver certificados intermediários, inclua-os no `server.crt` depois do certificado do site.

```
    -----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`

Cria o arquivo de chave privada na instância. *`private key contents`*Substitua pelo conteúdo da chave privada usada para criar a solicitação de certificado ou o certificado autoassinado.

- A chave `container_commands` reinicia o servidor nginx depois que tudo é configurado para o servidor carregar o arquivo de configuração nginx.

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-----

container_commands:
  01restart_nginx:
    command: "systemctl restart nginx"
```

> **Note**
>
> Evite confirmar um arquivo de configuração que contenha sua chave privada para o controle de origem. Depois que você tiver testado a configuração e confirmado se ela está funcionando, armazene a chave privada no Amazon S3 e modifique a configuração para fazer download dele durante a implantação. Para obter instruções, consulte Armazenar chaves privadas com segurança no Amazon S3.

Insira o seguinte em um arquivo com a extensão `.conf` no diretório `.platform/nginx/conf.d/` do seu pacote de origem (por ex. `.platform/nginx/conf.d/https.conf`). *`app_port`*Substitua pelo número da porta que seu aplicativo escuta. Este exemplo configura o servidor nginx para escutar na porta 443 usando SSL. Para obter mais informações sobre esses arquivos de configuração na plataforma .NET Core no Linux, consulte the section called “Servidor de 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;
    }
}
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Encerramento de HTTPS em EC2 instâncias da Amazon que executam o.NET

O arquivo de configuração a seguir cria e executa um PowerShell script do Windows que executa as seguintes tarefas:

- Verifica se há um certificado HTTPS vinculado à porta 443.
- Obtém o certificado PFX de um bucket do Amazon S3.

> **Note**
>
> Adicione uma política `AmazonS3ReadOnlyAccess` ao `aws-elasticbeanstalk-ec2-role` para acessar o certificado SSL no bucket do Amazon S3.

- Obtém a senha de AWS Secrets Manager.

> **Note**
>
> Adicione uma declaração em `aws-elasticbeanstalk-ec2-role` que permita a ação `secretsmanager:GetSecretValue` para o segredo que contém a senha do certificado

- Instala o certificado.
- Vincula o certificado à porta 443.

> **Note**
>
> Para remover o endpoint HTTP (porta 80), inclua o comando `Remove-WebBinding` na seção Remova a vinculação HTTP do exemplo.

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"  ## AWS Secrets Manager 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
```

Em um ambiente de instância única, você também deve modificar o grupo de segurança da instância para habilitar o tráfego na porta 443. O arquivo de configuração a seguir recupera a ID do grupo de segurança usando uma AWS CloudFormation função e adiciona uma regra a ela.

Example .ebextensions/ .config https-instance-single

```
Resources:
  sslSecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId: {"Fn::GetAtt" : ["AWSEBSecurityGroup", "GroupId"]}
      IpProtocol: tcp
      ToPort: 443
```

```
      FromPort: 443
      CidrIp: 0.0.0.0/0
```

Para um ambiente com balanceamento de carga, você configura o balanceador de carga para passar tráfego seguro intocado ou descriptografar e recriptografar para criptografia. end-to-end

Configurando a end-to-end criptografia em um ambiente Elastic Beanstalk com balanceamento de carga

O encerramento de conexões seguras no load balancer e o uso de HTTP no back-end podem ser suficientes para seu aplicativo. O tráfego de rede entre os recursos da AWS pode não ser escutado por instâncias que não façam parte da conexão, mesmo que elas estejam em execução na mesma conta.

No entanto, se você está desenvolvendo um aplicativo que precisa estar em conformidade com as rigorosas normas externas, pode ser necessário proteger todas as conexões de rede. É possível usar o console do Elastic Beanstalk ou os arquivos de configuração para fazer o balanceador de carga do ambiente do Elastic Beanstalk se conectar a instâncias de back-end com segurança a fim de atender a esses requisitos. O procedimento a seguir se concentra nos arquivos de configuração.

Primeiro, adicione um listener seguro ao seu load balancer, caso ainda não o tenha feito.

Você também deve configurar as instâncias do ambiente para escutar na porta segura e encerrar conexões HTTPS. A configuração varia de acordo com a plataforma. Para obter instruções, consulte Configuração da terminação de HTTPS na instância. Você pode usar um certificado autoassinado para as EC2 instâncias sem problemas.

Em seguida, configure o listener para encaminhar o tráfego usando HTTPS na porta segura usada pelo seu aplicativo. Use um dos seguintes arquivos de configuração, com base no tipo de load balancer que seu ambiente utiliza.

`.ebextensions/https-reencrypt-clb.config`

Use este arquivo de configuração com um Classic Load Balancer. Além de configurar o load balancer, o arquivo de configuração também altera a verificação de integridade padrão para usar a porta 443 e HTTPS, garantindo que o load balancer possa se conectar com segurança.

```
option_settings:
  aws:elb:listener:443:
```

```
    InstancePort: 443
    InstanceProtocol: HTTPS
  aws:elasticbeanstalk:application:
    Application Healthcheck URL: HTTPS:443/
```

`.ebextensions/https-reencrypt-alb.config`

Use este arquivo de configuração com um 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`

Use este arquivo de configuração com um Network Load Balancer.

```
option_settings:
  aws:elbv2:listener:443:
    DefaultProcess: https
    ListenerEnabled: 'true'
  aws:elasticbeanstalk:environment:process:https:
    Port: '443'
```

A opção `DefaultProcess` recebe esse nome em virtude do Application Load Balancers, que podem ter listeners não padrão na mesma porta para o tráfego de caminhos específicos (consulte Application Load Balancer para obter detalhes). Para um Network Load Balancer, a opção especifica o único processo de destino para esse listener.

Neste exemplo, chamamos o processo de `https` porque ele faz o listening do tráfego seguro (HTTPS). O listener envia o tráfego para o processo na porta designada usando o protocolo TCP, pois o Network Load Balancer funciona apenas com TCP. Isso está correto, porque o tráfego de rede para HTTP e HTTPS é implementado na parte superior do TCP.

> **Note**
>
> A CLI do EB e o console do Elastic Beanstalk aplicam os valores recomendados para as opções anteriores. Se quiser usar arquivos de configuração para definir a mesma coisa, você precisa remover essas configurações. Para mais detalhes, consulte Valores recomendados.

Na próxima tarefa, você precisa modificar o security group do load balancer para permitir tráfego. Dependendo da Amazon Virtual Private Cloud (Amazon VPC) em que você inicia seu ambiente, a VPC padrão ou uma VPC personalizada, o grupo de segurança do balanceador de carga variará. Em uma VPC padrão, o Elastic Load Balancing fornece um grupo de segurança padrão que todos os balanceadores de carga podem usar. Em uma Amazon VPC criada, o Elastic Beanstalk cria um grupo de segurança para o balanceador de carga usar.

Para oferecer suporte aos dois cenários, crie um grupo de segurança e instrua o Elastic Beanstalk a usá-lo. O seguinte arquivo de configuração cria um security group e o anexa ao load balancer.

`.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
```

Substitua o texto destacado pelo seu ID de VPC padrão ou personalizada. O exemplo anterior inclui entrada e saída pela porta 80 para permitir conexões HTTP. Remova essas propriedades se você deseja somente permitir conexões seguras.

Por fim, adicione regras de entrada e saída que permitam a comunicação pela porta 443, entre o security group do load balancer e o security group das instâncias.

`.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"]}
```

Faça isso separadamente da criação do security group para restringir os grupos de segurança de origem e destino, sem criar uma dependência circular.

Depois de concluir todas as tarefas anteriores, o load balancer se conecta a instâncias de back-end com segurança usando HTTPS. O load balancer não se importa se o certificado da instância é autoassinado ou emitido por uma autoridade de certificação confiável, e aceitará qualquer certificado apresentado.

Esse comportamento pode ser alterado adicionando políticas ao load balancer que o instrua a confiar apenas em um certificado específico. O arquivo de configuração a seguir cria duas políticas. Uma política especifica um certificado público, e o outro instrui o load balancer a confiar apenas naquele certificado para conexões com a porta 443 da instância.

`.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-----
```

Substitua o texto destacado pelo conteúdo do certificado público da sua EC2 instância.

Configurar o load balancer do ambiente para passagem de TCP

Se você não quiser que o balanceador de carga em seu AWS Elastic Beanstalk ambiente decifre o tráfego HTTPS, você pode configurar o ouvinte seguro para retransmitir solicitações para instâncias de back-end no estado em que se encontram.

> **Important**
>
> A configuração do balanceador de carga para retransmitir o tráfego HTTPS sem descriptografá-lo apresenta uma desvantagem. O balanceador de carga não pode ver as solicitações criptografadas e, portanto, não pode otimizar o roteamento ou relatar métricas de resposta.

Primeiro, configure as EC2 instâncias do seu ambiente para encerrar o HTTPS. Teste a configuração em um ambiente de instância única para garantir que tudo esteja funcionando antes de adicionar um load balancer à combinação.

Adicione um arquivo de configuração ao seu projeto para configurar um listener na porta 443 que transmite pacotes TCP como estão para a porta 443 nas instâncias de back-end:

`.ebextensions/https-lb-passthrough.config`

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: TCP
    InstancePort: 443
    InstanceProtocol: TCP
```

Em uma Amazon Virtual Private Cloud (Amazon VPC) padrão, também é necessário adicionar uma regra ao grupo de segurança das instâncias para permitir o tráfego de entrada na porta 443 do balanceador de carga:

`.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"] }
```

Em uma VPC personalizada, o Elastic Beanstalk atualiza a configuração do grupo de segurança para você.

Configurar o redirecionamento de HTTP para HTTPS

Este tópico descreve como lidar com o tráfego HTTP para a sua aplicação se os usuários finais ainda a iniciarem. Isso é feito ao configurar o redirecionamento de HTTP para HTTPS, às vezes conhecido como forçando HTTPS.

Para configurar o redirecionamento, configure primeiro seu ambiente para lidar com o tráfego HTTPS. Em seguida, redirecione o tráfego HTTP para HTTPS. Essas duas etapas são discutidas nas subseções a seguir.

Configurar o ambiente para lidar com o tráfego HTTPS

Dependendo da configuração do balanceamento de carga do ambiente, siga um destes procedimentos:

- Ambiente com balanceamento de carga: configure o balanceador de carga para encerrar o HTTPS.
- Ambiente de única instância: configure a aplicação para encerrar conexões HTTPS na instância. Esta configuração depende da Plataforma do seu ambiente.

Redirecionar o tráfego HTTP para HTTPS

Para redirecionar o tráfego HTTP para HTTPS para sua aplicação, é possível configurar os servidores Web nas instâncias do seu ambiente ou configurar o Application Load Balancer do ambiente.

Configurar os servidores Web da instância

Esse método funciona em qualquer ambiente de servidor Web. Configure servidores web em suas EC2 instâncias da Amazon para responder ao tráfego HTTP com um status de resposta de redirecionamento HTTP.

Esta configuração depende da Plataforma do seu ambiente. Encontre a pasta da sua plataforma na `https-redirect`coleção em GitHub e use o arquivo de configuração de exemplo nessa pasta.

Se o seu ambiente usar as verificações de integridade do Elastic Load Balancing, o balanceador de carga esperará uma instância de integridade para responder a mensagens de verificação de integridade HTTP com respostas HTTP 200 (OK). Portanto, o servidor da web não deve redirecionar essas mensagens para HTTPS. Os arquivos de configuração demonstrativos em `https-redirect` lidam com esse requisito corretamente.

Configurar o balanceador de carga

Esse método funciona se você tiver um ambiente com balanceamento de carga que use um Application Load Balancer. Um Application Load Balancer pode enviar respostas de

redirecionamento à medida que o tráfego HTTP chega. Nesse caso, não é preciso configurar o redirecionamento nas instâncias do ambiente.

Temos dois exemplos de arquivos de configuração GitHub que mostram como configurar um Application Load Balancer para redirecionamento.

- O arquivo de configuração `alb-http-to-https-redirection-full.config` cria um listener HTTPS na porta 443 e modifica o listener padrão da porta 80 para redirecionar o tráfego HTTP de entrada para HTTPS.
- O arquivo de configuração `alb-http-to-https-redirection.config` espera que o receptor 443 seja definido. Para defini-lo, você pode usar namespaces de configuração padrão do Elastic Beanstalk ou o console do Elastic Beanstalk. Em seguida, ele cuida de modificar o listener da porta 80 para redirecionamento.

Plataformas do Elastic Beanstalk

AWS Elastic Beanstalk fornece uma variedade de plataformas nas quais você pode criar seus aplicativos. Projete a aplicação Web para uma dessas plataformas, e o Elastic Beanstalk implanta o código na versão de plataforma selecionada para criar um ambiente de aplicação ativo.

O Elastic Beanstalk oferece plataformas para diferentes linguagens de programação, servidores de aplicações e contêineres do Docker. Algumas plataformas possuem várias versões simultaneamente compatíveis.

Tópicos

- Glossário de plataformas do Elastic Beanstalk
- Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk
- Política de suporte às plataformas do Elastic Beanstalk
- Cronograma de lançamentos da plataforma Elastic Beanstalk
- Plataformas compatíveis com Elastic Beanstalk
- Plataformas Linux do Elastic Beanstalk
- Estender as plataformas Linux do Elastic Beanstalk

Glossário de plataformas do Elastic Beanstalk

A seguir estão os principais termos relacionados às AWS Elastic Beanstalk plataformas e seu ciclo de vida.

Runtime

> Os programas de software de runtime específicos de cada linguagem de programação (estrutura, bibliotecas, intérprete, vm, etc.) que são necessários para executar o código do aplicativo.

Componentes do Elastic Beanstalk

> Componentes de software que o Elastic Beanstalk adiciona a uma plataforma para habilitar a funcionalidade Elastic Beanstalk. Por exemplo, o agente de integridade aprimorada é necessário para reunir e relatar informações de integridade.

Plataforma

Uma combinação de um sistema operacional (SO), runtime, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. As plataformas fornecem componentes que estão disponíveis para executar seu aplicativo.

Versão da plataforma

Uma combinação de versões específicas de um sistema operacional (SO), runtime, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. Crie um ambiente do Elastic Beanstalk com base em uma versão da plataforma e implanta a aplicação nele.

Uma plataforma tem um número de versão semântica do formulário X.Y.Z, em que X é a versão principal, Y é a versão secundária, e Z é a versão do patch.

Uma versão de plataforma pode estar em um dos seguintes estados:

- Recomendado — A versão mais recente da plataforma em uma ramificação de plataforma compatível. Essa versão contém a maioria dos up-to-date componentes e é recomendada para uso em ambientes de produção.
- Não recomendado — Qualquer versão da plataforma que não seja a versão mais recente em sua ramificação de plataforma. Embora essas versões possam permanecer funcionais, é altamente recomendável atualizar para a versão mais recente da plataforma. Você pode usar as atualizações gerenciadas da plataforma para ajudar a permanecer up-to-date automaticamente.

Você pode verificar se uma versão da plataforma é recomendada usando o comando AWS CLI describe-platform-version e verificando o `PlatformLifecycleState` campo.

Ramificação da plataforma

Uma linha de versões de plataforma que compartilha versões específicas (geralmente principais) de alguns de seus componentes, como sistema operacional (SO), runtime ou componentes do Elastic Beanstalk. Por exemplo: Python 3.13 em execução no Amazon Linux 2023 de 64 bits; IIS 10.0 em execução no Windows Server 2025 de 64 bits. As filiais da plataforma recebem atualizações na forma de novas versões da plataforma. Cada versão sucessiva da plataforma em uma ramificação é uma atualização da anterior.

A versão recomendada em cada ramificação de plataforma compatível está disponível incondicionalmente para a criação do ambiente. Uma versão anterior da plataforma está disponível para você se você estava usando um ambiente com ela no momento em que a versão

da plataforma foi substituída por uma nova versão da plataforma. As versões anteriores da plataforma não têm a maioria dos up-to-date componentes e não são recomendadas para uso.

Uma ramificação de plataforma pode estar em um dos seguintes estados:

- Compatível: uma ramificação de plataforma atual. Ela consiste inteiramente em componentes compatíveis. Os componentes suportados não atingiram o fim da vida útil (EOL), conforme designado por seus fornecedores. Ela recebe atualizações contínuas da plataforma e é recomendada para uso em ambientes de produção. Para obter uma lista de ramificações de plataforma suportadas, consulte Plataformas suportadas com o Elastic Beanstalk no Guia Plataformas do AWS Elastic Beanstalk .
- Beta: uma ramificação da plataforma de pré-visualização e de pré-lançamento. Ela é de natureza experimental. Ela pode receber atualizações contínuas da plataforma por um tempo, mas não tem suporte a longo prazo. Uma ramificação de plataforma beta não é recomendada para uso em ambientes de produção. Use-a apenas para avaliação. Para obter uma lista de ramificações de plataforma beta, consulte Versões de plataforma do Elastic Beanstalk em beta público no Guia Plataformas do AWS Elastic Beanstalk .
- Obsoleto — Uma ramificação da plataforma em que um ou mais componentes (como o tempo de execução ou o sistema operacional) estão chegando ao fim da vida útil (EOL) ou atingiram o EOL, conforme designado por seus fornecedores. Embora uma ramificação de plataforma obsoleta continue recebendo novas versões da plataforma até a data de desativação, os componentes que atingiram o EOL não recebem atualizações. Por exemplo, se uma versão em tempo de execução atingir o EOL, a ramificação da plataforma será marcada como obsoleta, mas continuará recebendo atualizações do sistema operacional até a data de desativação da ramificação da plataforma. A ramificação da plataforma não continuará recebendo atualizações para a versão de tempo de execução do EOL. O uso de uma ramificação de plataforma obsoleta não é recomendado.
- Aposentado — Uma ramificação da plataforma que não recebe mais nenhuma atualização. As filiais descontinuadas da plataforma não estão disponíveis para criar novos ambientes do Elastic Beanstalk usando o console do Elastic Beanstalk. Se seu ambiente usa uma ramificação de plataforma descontinuada, você deve atualizar para uma ramificação de plataforma compatível para continuar recebendo atualizações. O uso de uma ramificação de plataforma descontinuada não é recomendado. Para obter mais detalhes sobre filiais de plataformas desativadas, consultethe section called “Política de suporte de plataforma”. Para obter uma lista das filiais da plataforma programadas para serem desativadas, consulte Cronograma de desativação das filiais da plataforma. Para ver as filiais anteriores da plataforma desativada, consulte Histórico de filiais desativadas da plataforma.

Se seu ambiente usar uma ramificação de plataforma defasada ou desativada, recomendamos atualizá-la para uma versão de plataforma em uma ramificação de plataforma compatível. Para obter detalhes, consulte the section called “Atualizações da plataforma”.

Você pode verificar o estado de uma ramificação da plataforma usando o comando AWS CLI describe-platform-version e verificando o `PlatformBranchLifecycleState` campo.

Atualização da plataforma

Um lançamento de novas versões da plataforma que contêm as atualizações para alguns componentes da plataforma: sistema operacional, runtime, servidor Web, servidor de aplicações e componentes do Elastic Beanstalk. As atualizações da plataforma seguem a taxonomia da versão semântica e podem ter três níveis:

- Atualização principal: uma atualização que tenha alterações incompatíveis com as versões existentes da plataforma. Talvez seja necessário modificar seu aplicativo para ser executado corretamente em uma nova versão principal. Uma atualização principal tem um novo número de versão de plataforma principal.
- Atualização secundária — Uma atualização que tem alterações que são retrocompatíveis com as versões existentes da plataforma na maioria dos casos. Dependendo do seu aplicativo, talvez seja necessário modificá-lo para ser executado corretamente em uma nova versão secundária. Uma atualização secundária tem um novo número de versão de plataforma secundária.
- Atualização de patch: uma atualização que consiste em versões de manutenção (correções de erros, atualizações de segurança e melhorias de performance) que são compatíveis com uma versão anterior da plataforma. Uma atualização de patch tem um novo número de versão de plataforma de patch.

Managed Updates

Um recurso do Elastic Beanstalk que aplica automaticamente patches e atualizações secundárias aos componentes do sistema operacional (SO), runtime, servidor Web, servidor de aplicações e Elastic Beanstalk para uma versão de plataforma compatível com o Elastic Beanstalk. Uma atualização gerenciada aplica uma versão mais recente da plataforma na mesma ramificação da plataforma ao seu ambiente. Você pode configurar as atualizações gerenciadas para aplicar somente atualizações de patch ou atualizações de versão secundária e de patch. Você também pode desabilitar atualizações gerenciadas completamente.

Para obter mais informações, consulte Atualizações gerenciadas de plataforma.

Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk

AWS e nossos clientes compartilham a responsabilidade de alcançar um alto nível de segurança e conformidade de componentes de software. Esse modelo compartilhado reduz os encargos operacionais.

Para obter detalhes, consulte o Modelo de Responsabilidade AWS Compartilhada.

AWS Elastic Beanstalk ajuda você a cumprir sua parte do modelo de responsabilidade compartilhada, fornecendo um recurso de atualizações gerenciadas. Esse recurso aplica automaticamente atualizações secundárias e patches para uma versão da plataforma compatível do Elastic Beanstalk. Se uma atualização gerenciada falhar, o Elastic Beanstalk notificará você sobre a falha a fim de garantir que você esteja ciente disso e possa tomar medidas imediatas.

Para obter mais informações, consulte Atualizações gerenciadas de plataforma.

Além disso, o Elastic Beanstalk faz o seguinte:

- Publica a política de suporte da plataforma e a programação de desativação para os próximos 12 meses.
- Libera atualizações de patch, secundárias e principais dos componentes do sistema operacional (SO), runtime, servidor de aplicações e servidor Web normalmente em até 30 dias após sua disponibilidade. O Elastic Beanstalk é responsável por criar atualizações para seus respectivos componentes que estão presentes em suas versões de plataforma compatíveis. Todas as outras atualizações vêm diretamente de seus fornecedores (proprietários ou da comunidade).

Anunciamos todas as atualizações para nossas plataformas suportadas em nossas notas de versão no guia de notas de versão AWS Elastic Beanstalk . Também fornecemos uma lista de todas as plataformas suportadas e seus componentes, juntamente com um histórico da plataforma, no guia Plataformas AWS Elastic Beanstalk . Para ter mais informações, consulte Plataformas compatíveis e histórico de componentes.

Você é responsável por fazer o seguinte:

- Atualize todos os componentes que você controla (identificados como Cliente no Modelo de Responsabilidade AWS Compartilhada). Isso inclui garantir a segurança do aplicativo, seus dados e qualquer componente que o aplicativo exigir e que tenha sido obtido por download.

- Certifique-se de que os ambientes do Elastic Beanstalk estejam em execução em uma versão compatível da plataforma e migre qualquer ambiente em execução em uma versão retirada da plataforma para uma versão compatível.
- Se estiver usando uma imagem de máquina da Amazon (AMI) personalizada para o ambiente Elastic Beanstalk, corrija, mantenha e teste essa AMI para que ela permaneça atualizada e compatível com uma versão compatível da plataforma Elastic Beanstalk. Para obter mais informações sobre o gerenciamento de ambientes com uma AMI personalizada, consulte Usar uma imagem de máquina Amazon personalizada (AMI) no ambiente Elastic Beanstalk.
- Resolva todos os problemas que surgirem em tentativas de atualização gerenciadas com falha e tente fazer a atualização novamente.
- Aplique patch ao sistema operacional, runtime, servidor de aplicações e servidor Web por conta própria se optou por não utilizar as atualizações gerenciadas do Elastic Beanstalk. Você pode fazer isso aplicando as atualizações de plataforma manualmente ou aplicar patch diretamente nos componentes em todos os recursos de ambiente relevantes.
- Gerencie a segurança e a conformidade de todos AWS os serviços que você usa fora do Elastic Beanstalk de acordo com AWS o Modelo de Responsabilidade Compartilhada.

Política de suporte às plataformas do Elastic Beanstalk

O Elastic Beanstalk é compatível com ramificações de plataformas que ainda recebem atualizações secundárias e de patches contínuas de seus fornecedores (proprietários ou comunidade). Para obter uma definição de termos relacionados, consulte Glossário de plataformas do Elastic Beanstalk.

Ramificações desativadas da plataforma

Quando um componente de uma ramificação de plataforma com suporte é marcado como Fim da Vida Útil (EOL) por seu fornecedor, o Elastic Beanstalk marca a ramificação da plataforma como retirada. Os componentes de uma ramificação de plataforma incluem o seguinte: sistema operacional (SO), versão da linguagem de runtime, servidor de aplicações ou servidor Web.

Quando uma ramificação da plataforma é marcada como desativada, as seguintes políticas são aplicáveis:

- O Elastic Beanstalk deixa de fornecer atualizações de manutenção, incluindo atualizações de segurança.
- O Elastic Beanstalk não fornece mais suporte técnico para ramificações de plataforma retiradadas.

- O Elastic Beanstalk não disponibiliza mais a ramificação da plataforma para novos clientes do Elastic Beanstalk para implantações em novos ambientes. Há um período de carência de 90 dias a partir da data de retirada publicada para clientes existentes com ambientes ativos sendo executados em ramificações de plataforma desativadas.

> **Note**
>
> Uma ramificação de plataforma retirada não estará disponível no console do Elastic Beanstalk. No entanto, ele estará disponível por meio do AWS CLI EB CLI e da EB API para clientes que tenham ambientes existentes com base na ramificação descontinuada da plataforma. Os clientes existentes também podem usar os consoles Clonar ambiente e Reconstruir ambiente.

Para obter uma lista das ramificações da plataforma programadas para retirada, consulte Programação de ramificações desativadas da plataforma no tópico de programação da plataforma Elastic Beanstalk a seguir.

Para obter mais informações sobre o que esperar quando a ramificação da plataforma de seu ambiente for retiradada, consulte Perguntas frequentes sobre descontinuação de plataformas.

Após o período de carência de 90 dias

Nossa política para ramificações de plataforma descontinuadas não remove o acesso a ambientes nem exclui recursos. No entanto, os clientes existentes que executam um ambiente Elastic Beanstalk em uma filial de plataforma retiradada devem estar cientes dos riscos de fazer isso. Esses ambientes podem acabar em uma situação imprevisível, porque o Elastic Beanstalk não é capaz de fornecer atualizações de segurança, suporte técnico ou hotfixes para ramificações de plataforma retiradadas devido ao fato de o fornecedor marcar seu componente como EOL.

Por exemplo, uma vulnerabilidade de segurança prejudicial e crítica pode surgir em um ambiente sendo executado em uma ramificação de plataforma retirada. Ou uma ação da API do EB pode parar de funcionar no ambiente se, com o tempo, tornar-se incompatível com o serviço Elastic Beanstalk. Quanto mais tempo um ambiente baseado em uma ramificação de plataforma retirada permanecer ativo, maior a chance de que esses tipos de riscos ocorram. Para continuar a obter os benefícios de importantes melhorias de funcionalidade, performance e segurança oferecidas por fornecedores de componentes nas versões mais recentes, é altamente recomendável atualizar todos os ambientes do Elastic Beanstalk para uma versão compatível da plataforma.

Se a aplicação encontrar problemas ao ser executada em uma ramificação de plataforma descontinuada e você não conseguir migrá-la para uma plataforma compatível, será necessário considerar outras alternativas. Soluções alternativas incluem o encapsulamento da aplicação em uma imagem do Docker para executá-la como um contêiner do Docker. Isso permitiria que um cliente usasse qualquer uma de nossas soluções Docker, como nossas plataformas AL2 Elastic AL2 Beanstalk 023/Docker ou outros serviços baseados em Docker, como Amazon ECS ou Amazon EKS. As alternativas que não são do Docker incluem nosso AWS CodeDeploy serviço, que permite a personalização completa dos tempos de execução que você deseja.

Cronograma de lançamentos da plataforma Elastic Beanstalk

Além da cadência mensal de lançamento de novas versões de ramificação da plataforma, nossa manutenção de lançamentos também inclui os seguintes processos:

- Lançamento de novas ramificações de plataforma: normalmente, elas introduzem uma nova versão principal de uma linguagem de runtime, sistema operacional ou servidor de aplicações.
- Retirada de ramificações da plataforma: retiraremos uma ramificação da plataforma quando um de seus componentes chegar ao fim da vida útil (EOL). Para obter mais informações sobre nossa política para filiais retiradadas, consulte Política de suporte às plataformas do Elastic Beanstalk

Tópicos

- Recursos de planejamento
- Próximos lançamentos de ramificações de plataforma
- Programação de ramificações desativadas da plataforma
- Histórico de ramificações de plataforma retiradas
- Histórico de servidores e sistemas operacionais retirados

Recursos de planejamento

Os recursos a seguir podem ajudar você a planejar a manutenção e o suporte para sua aplicação em execução em uma plataforma Elastic Beanstalk.

- AWS Elastic Beanstalk Guia de plataformas — Este guia fornece uma lista detalhada de componentes para cada uma de nossas ramificações de plataforma. Ele também fornece um histórico de plataformas por data de lançamento com os mesmos detalhes. Este guia pode informar quando componentes específicos da ramificação da sua plataforma forem alterados. Se

a aplicação começar a se comportar de forma diferente, você também poderá cruzar a data da ocorrência no guia de plataformas para ver se houve alguma alteração na plataforma que possa ter afetado a aplicação.

- AWS Elastic Beanstalk Notas de lançamento — Nossas notas de lançamento anunciam todos os lançamentos da nossa plataforma, tanto menores quanto principais. Isso inclui nossas atualizações mensais da plataforma, versões de segurança, hotfixes e anúncios de retiradadoria. É possível assinar nossos feeds de RSS na documentação das Notas de versão.

Próximos lançamentos de ramificações de plataforma

A tabela a seguir lista as próximas ramificações da plataforma Elastic Beanstalk e sua data de lançamento prevista. Essas datas são provisórias e estão sujeitas a alterações.

Versão do runtime/ramificação da plataforma	Sistema operacional	Data de lançamento prevista
Corretto 25	Amazon Linux 2023	Outubro de 2025
Corretto 25 with Tomcat 11	Amazon Linux 2023	Outubro de 2025
Python 3.14	Amazon Linux 2023	Novembro de 2025
Node.js 24	Amazon Linux 2023	Novembro de 2025
.NET 10	Amazon Linux 2023	Dezembro de 2025
PHP 8.5	Amazon Linux 2023	Janeiro de 2026
Ruby 3.5	Amazon Linux 2023	Fevereiro de 2026

Programação de ramificações desativadas da plataforma

A tabela a seguir lista as ramificações da plataforma Elastic Beanstalk que estão programadas para serem retiradas, pois alguns de seus componentes estão chegando ao fim da vida útil (EOL).

Para obter uma lista mais detalhada das ramificações de plataformas que estão sendo retiradas e que incluem seus componentes específicos, consulte Versões de plataformas que estão sendo retiradas no guia de Plataformas do AWS Elastic Beanstalk .

Versão do runtime/ramificação da plataforma	Sistema operacional	Data prevista para a retirada
Node.js 18 AL2023	Amazon Linux 2023	31 de julho de 2025
Node.js 18 AL2	Amazon Linux 2	31 de julho de 2025

Histórico de ramificações de plataforma retiradas

As tabelas a seguir listam as ramificações da plataforma Elastic Beanstalk que já estão em status de retirada. É possível ver um histórico detalhado dessas ramificações de plataforma e seus componentes em Histórico da plataforma do Guia de plataformas do AWS Elastic Beanstalk .

Amazon Linux 2023 (AL2023)

Versão do runtime/ramificação da plataforma	Data de retirada		
.NET 6 AL2023	8 de abril de 2025		

Amazon Linux (2AL2)

Versão do runtime/ramificação da plataforma	Data de retirada		
Corretto 11 with Tomcat 8.5 AL2	10 de outubro de 2024		
Corretto 8 with Tomcat 8.5 AL2	10 de outubro de 2024		
Corretto 11 with Tomcat 7 AL2	29 de junho de 2022		

Versão do runtime/ramificação da plataforma	Data de retirada
Corretto 8 with Tomcat 7 AL2	29 de junho de 2022
Node.js 16 AL2	10 de outubro de 2024
Node.js 14 AL2	10 de outubro de 2024
Node.js 12 AL2	23 de dezembro de 2022
Node.js 10 AL2	29 de junho de 2022
PHP 8.0 AL2	10 de outubro de 2024
PHP 7.4 AL2	9 de junho de 2023
PHP 7.3 AL2	29 de junho de 2022
PHP 7.2 AL2	29 de junho de 2022
Python 3.8 AL2	8 de abril de 2025
Python 3.7 AL2	10 de outubro de 2024
Ruby 3.0 AL2	10 de outubro de 2024
Ruby 2.7 AL2	10 de outubro de 2024
Ruby 2.6 AL2	23 de dezembro de 2022
Ruby 2.5 AL2	29 de junho de 2022

Amazon Linux AMI (AL1)

Versão do runtime/ramificação da plataforma	Data de retirada		
Single Container Docker	18 de julho de 2022		
Multicontainer Docker	18 de julho de 2022		
Preconfigured Docker - GlassFish 5.0 with Java 8	18 de julho de 2022		
Go 1	18 de julho de 2022		
Java 8	18 de julho de 2022		
Java 7	18 de julho de 2022		
Java 8 with Tomcat 8.5	18 de julho de 2022		
Java 7 with Tomcat 7	18 de julho de 2022		
Node.js	18 de julho de 2022		
PHP 7.2 - 7.3	18 de julho de 2022		
Python 3.6	18 de julho de 2022		
Ruby 2,4, 2.5, 2.6 with Passenger	18 de julho de 2022		

Versão do runtime/ramificação da plataforma	Data de retirada
Ruby 2.4, 2.5, 2.6 with Puma	18 de julho de 2022
Go 1.3–1.10	31 de outubro de 2020
Java 6	31 de outubro de 2020
Node.js 4.x–8.x	31 de outubro de 2020
PHP 5.4–5.6	31 de outubro de 2020
PHP 7.0–7.1	31 de outubro de 2020
Python 2.6, 2.7, 3.4	31 de outubro de 2020
Ruby 1.9.3	31 de outubro de 2020
Ruby 2.0–2.3	31 de outubro de 2020

> **Note**
>
> Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações, consulte Perguntas frequentes sobre descontinuação de plataformas.

Windows Server

Versão do runtime/ramificação da plataforma	Data de retirada		
IIS 8.5 em execução no Windows Server (e Core) de 64 bits 2012 R2 versão 0.1.0	29 de junho de 2022		
IIS 8.5 em execução no Windows Server (e Core) de 64 bits 2012 R2 versão 1.2.0	29 de junho de 2022		
IIS 10.0 em execução no Windows Server de 64 bits 2016 (e Core) versão 1.2.0	29 de junho de 2022		
ISS 8 em execução no Windows Server de 64 bits 2012 de versão de plataforma R1	22 de junho de 2022		
IIS 8 em execução no Windows Server	22 de junho de 2022		

Versão do runtime/ramificação da plataforma	Data de retirada		
de 64 bits 2012 R1 versão 0.1.0			
IIS 8 em execução no Windows Server de 64 bits 2012 R1 versão 1.2.0	22 de junho de 2022		

> **Note**
>
> Para obter mais informações sobre a descontinuação das ramificações da plataforma Windows 2012 R2, consulte Windows Server 2012 R2 platform branches retired nas AWS Elastic Beanstalk Release Notes.

Histórico de servidores e sistemas operacionais retirados

As tabelas a seguir fornecem um histórico dos sistemas operacionais, servidores de aplicações e servidores Web que não são mais compatíveis com as plataformas do Elastic Beanstalk. Todas as ramificações da plataforma que utilizavam esses componentes foram retiradas. As datas refletem a data de retiradadoria da última ramificação da plataforma do Elastic Beanstalk que incluía o componente.

Sistema operacional

Versão do SO	Data de desativação da plataforma		
Windows Server 2012 R2 running IIS 8.5	4 de dezembro de 2023		

Versão do SO	Data de desativação da plataforma		
Windows Server Core 2012 R2 running IIS 8.5	4 de dezembro de 2023		
Amazon Linux AMI (AL1)	18 de julho de 2022		
Windows Server 2012 R1	22 de junho de 2022		
Windows Server 2008 R2	28 de outubro de 2019		

Servidores de aplicações

Versão do servidor do aplicativo	Data de desativação da plataforma		
Tomcat 7	29 de junho de 2022 para plataformas Amazon Linux 2 (AL2) 18 de julho de 2022 para plataformas Amazon Linux AMI (AL1)		
Tomcat 8	31 de outubro de 2020		
Tomcat 6	31 de outubro de 2020		

Servidores da web

Versão do servidor web	Data de desativação da plataforma		
ISS 8 em execução no	22 de junho de 2022		

Versão do servidor web	Data de desativação da plataforma		
Windows Server de 64 bits			
Servidor HTTP Apache 2.2	31 de outubro de 2020		
Nginx 1.12.2	31 de outubro de 2020		

Plataformas compatíveis com Elastic Beanstalk

AWS Elastic Beanstalk fornece uma variedade de plataformas nas quais você pode criar seus aplicativos. Projete a aplicação Web para uma dessas plataformas, e o Elastic Beanstalk implanta o código na versão de plataforma selecionada para criar um ambiente de aplicação ativo.

O Elastic Beanstalk provisiona os recursos necessários para executar seu aplicativo, incluindo uma ou mais instâncias da Amazon. EC2 A pilha de software em execução nas EC2 instâncias da Amazon depende da versão específica da plataforma que você selecionou para o seu ambiente.

O nome da pilha de soluções para uma ramificação da plataforma

É possível usar o nome da pilha de soluções para uma determinada versão de ramificação da plataforma para iniciar um ambiente com a EB CLI, a API do Elastic Beanstalk, ou a AWS CLI. O guia Plataformas do AWS Elastic Beanstalk lista o nome da pilha de soluções sob a versão da ramificação da plataforma nas seções Plataformas compatíveis do Elastic Beanstalk e Histórico de plataformas.

Para recuperar todos os nomes da pilha de soluções que você pode usar para criar um ambiente, use a ListAvailableSolutionStacksAPI ou a `aws elasticbeanstalk list-available-solution-stacks`na CLI AWS .

É possível personalizar e configurar o software do qual seu aplicativo depende na plataforma Linux. Saiba mais em Personalizar software em servidores Linux e Personalizar software em Windows Servers. Há notas de release detalhadas disponíveis para releases recentes nas Notas de release do AWS Elastic Beanstalk.

Plataformas compatíveis e histórico de componentes

O guia de Plataformas do AWS Elastic Beanstalk lista todas as versões atuais do branch da plataforma na seção Plataformas compatíveis do Elastic Beanstalk. O guia de Plataformas também lista um histórico de plataformas para cada plataforma, que inclui uma lista de versões anteriores da plataforma de ramificação. Para visualizar o histórico da plataforma de cada plataforma, selecione um dos seguintes links.

- Docker
- Go
- Java SE
- Tomcat (executando Java SE)
- .NET Core no Linux
- .NET no Windows Server
- Node.js
- PHP
- Python
- Ruby

Plataformas Linux do Elastic Beanstalk

As plataformas Elastic Beanstalk Linux oferecem uma grande quantidade de funcionalidades prontas para uso. Você pode estender as plataformas de várias maneiras para oferecer suporte ao seu aplicativo. Para obter detalhes, consulte the section called “Estender plataformas Linux”.

A maioria das plataformas compatíveis com o Elastic Beanstalk são baseadas no sistema operacional Linux. Especificamente, essas plataformas são baseadas no Amazon Linux, uma distribuição Linux fornecida pela AWS. As plataformas Linux do Elastic Beanstalk usam instâncias do Amazon Elastic Compute Cloud EC2 (Amazon), e essas instâncias executam o Amazon Linux.

Tópicos

- Versões compatíveis do Amazon Linux
- Lista de plataformas Linux do Elastic Beanstalk
- Fluxo de trabalho de implantação da instância

- Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores
- Ferramentas de script de plataforma para ambientes Elastic Beanstalk

Versões compatíveis do Amazon Linux

AWS Elastic Beanstalk oferece suporte a plataformas baseadas no Amazon Linux 2 e no Amazon Linux 2023.

Para obter mais informações sobre Amazon Linux 2 e Amazon Linux 2023, consulte o seguinte:

- Amazon Linux 2 — Amazon Linux no Guia do EC2 usuário da Amazon.
- Amazon Linux 2023 – O que é o Amazon Linux 2023? no Guia do usuário Amazon Linux 2023.

Para obter detalhes sobre as versões de plataforma compatíveis, consulte Plataformas compatíveis com Elastic Beanstalk.

> **Note**
>
> Você pode migrar seu aplicativo de uma ramificação da plataforma ou do AL1 Elastic AL2 Beanstalk para a ramificação equivalente da plataforma 023. AL2 Para obter mais informações, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Amazon Linux 2023

AWS anunciou a disponibilidade geral do Amazon Linux 2023 em março de 2023. O guia do usuário Amazon Linux 2023 resume as principais diferenças entre o Amazon Linux 2 e o Amazon Linux 2023. Para obter mais informações, consulte Comparando Amazon Linux 2 e Amazon Linux 2023 no guia do usuário.

Há um alto grau de compatibilidade entre as plataformas Elastic Beanstalk Amazon Linux 2 e Amazon Linux 2023. Embora haja algumas diferenças a serem observadas:

- Instance Metadata Service versão 1 (IMDSv1) — A configuração da IMDSv1 opção Desativar é padronizada `true` em AL2 plataformas 023. O padrão está `false` nas AL2 plataformas.

- ferramenta de instância pkg-repo — A pkg-repo ferramenta não está disponível para ambientes executados em AL2 plataformas 023. No entanto, você pode aplicar manualmente as atualizações do pacote e do sistema operacional a uma instância AL2 023. Para obter mais informações, consulte Gerenciamento de pacotes e atualizações do sistema operacional no Guia do usuário do Amazon Linux 2023
- HTTPd Configuração do Apache — O `httpd.conf` arquivo Apache para plataformas AL2 023 tem algumas definições de configuração que são diferentes daquelas para: AL2
  - Negar acesso a todo o sistema de arquivos do servidor por padrão. Essas configurações são descritas em Proteger arquivos do servidor por padrão na página Dicas de segurança do site Apache.
  - Impeça que os usuários substituam os recursos de segurança que você configurou. A configuração nega o acesso à configuração `.htaccess` em todos os diretórios, exceto aqueles especificamente habilitados. Essa configuração é descrita em Protegendo as configurações do sistema na página Dicas de segurança do site Apache. A página Tutorial do Apache HTTP Server: arquivos.htaccess afirma que essa configuração pode ajudar a melhorar o desempenho.
  - Negue acesso a arquivos com padrão de nome `.ht*`. Essa configuração impede que os clientes da Web exibam arquivos `.htaccess` e `.htpasswd`.

Você pode alterar qualquer uma das definições de configuração acima para seu ambiente. Para obter mais informações, consulte Configurar o Apache HTTPD.

Lista de plataformas Linux do Elastic Beanstalk

A lista a seguir fornece as plataformas Linux compatíveis com o Elastic Beanstalk para diferentes linguagens de programação, bem como para contêineres Docker. O Elastic Beanstalk oferece plataformas baseadas no Amazon Linux 2 e Amazon Linux 2023 para todos eles. Para saber mais sobre uma plataforma, selecione o link correspondente.

- Docker (e ECS Docker)
- Go
- Tomcat (executando Java SE)
- Java SE
- .NET Core no Linux
- Node.js
- PHP

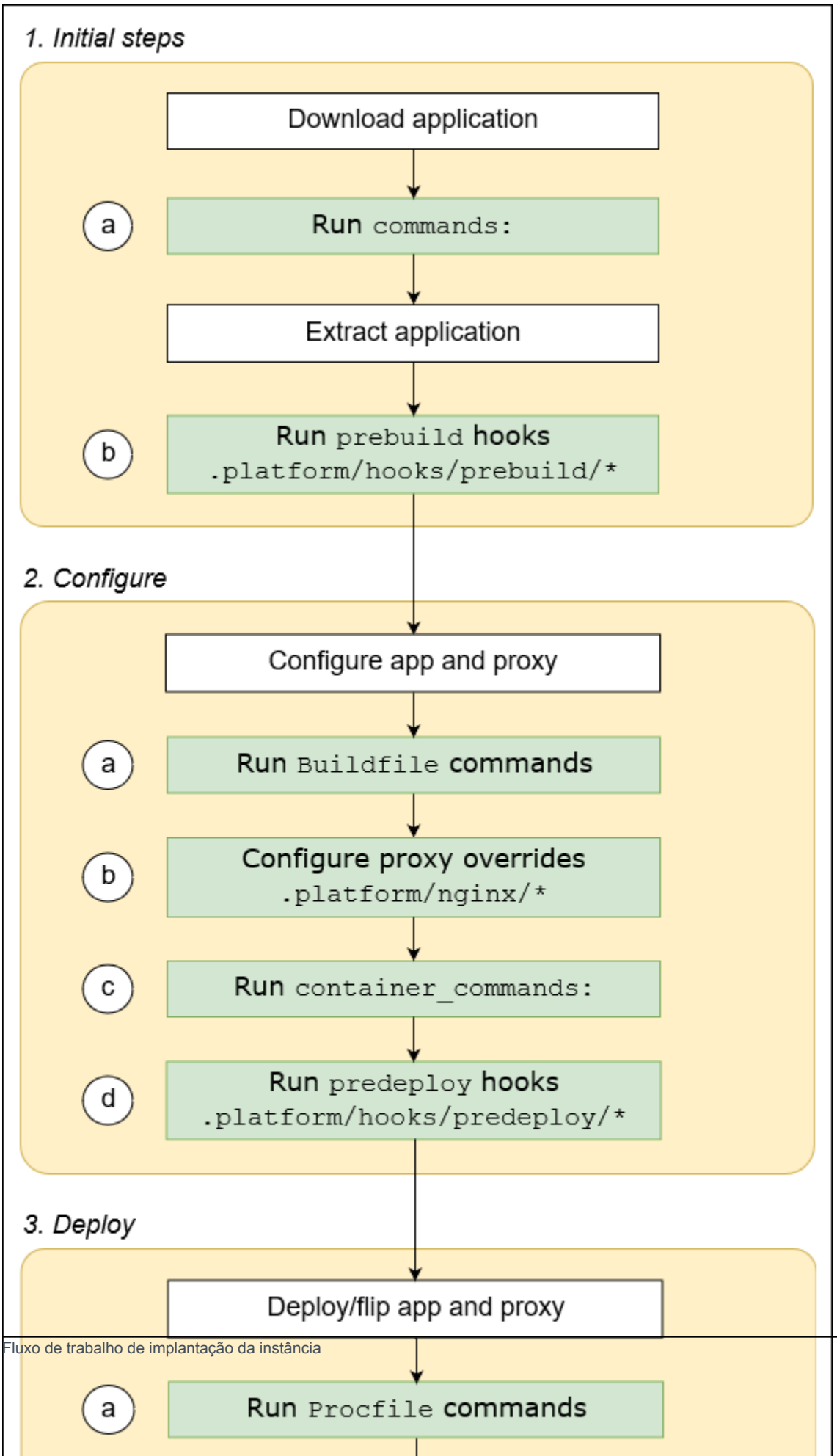
- Python
- Ruby

Fluxo de trabalho de implantação da instância

> **Note**
>
> As informações nesta seção não se aplicam à ramificação da plataforma ECS em execução no Amazon Linux 2 e no Amazon Linux 2023. Para obter mais informações, consulte a próxima seção, Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores.

Como existem muitas maneiras de estender a plataforma do ambiente, é útil saber o que acontece sempre que o Elastic Beanstalk provisiona ou implementa uma instância. O diagrama a seguir mostra todo esse fluxo de trabalho de implantação. Ele representa as diferentes fases de uma implantação e as etapas que o Elastic Beanstalk realiza em cada fase.

> **Observações**
>
> - O diagrama não representa o conjunto completo de etapas que o Elastic Beanstalk realiza em instâncias do ambiente durante a implantação. Nós fornecemos este diagrama como ilustração, para fornecer a você a ordem e o contexto para a execução de suas personalizações.
> - Para simplificar, o diagrama menciona apenas os subdiretórios de hook `.platform/hooks/*` (para implantações de aplicações), e não os subdiretórios de hook `.platform/confighooks/*` (para implantações de configurações). Os hooks nos últimos subdiretórios são executados durante exatamente as mesmas etapas que os hooks nos subdiretórios correspondentes mostrados no diagrama.


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

A lista a seguir detalha as fases e etapas de implantação.

1. Passos iniciais

   O Elastic Beanstalk baixa e extrai a aplicação. Após cada uma dessas etapas, o Elastic Beanstalk executa uma das etapas de extensibilidade.

   a. Executa comandos encontrados na seção commands: de qualquer arquivo de configuração.

   b. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/prebuild` do pacote de origem (`.platform/confighooks/prebuild` para uma implantação de configuração).

2. Configurar

   O Elastic Beanstalk configura a aplicação e o servidor de proxy.

   a. Executa os comandos encontrados no `Buildfile` no pacote de origem.

   b. Copia seus arquivos de configuração de proxy personalizados, se você tiver algum no diretório `.platform/nginx` do pacote de origem, para seu local de runtime.

   c. Executa comandos encontrados na seção container_commands: de qualquer arquivo de configuração.

   d. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/predeploy` do pacote de origem (`.platform/confighooks/predeploy` para uma implantação de configuração).

3. Implante

   O Elastic Beanstalk implanta e executa a aplicação e o servidor de proxy.

   a. Executa o comando encontrado no arquivo `Procfile` em seu pacote de origem.

   b. Executa ou executa novamente o servidor de proxy com seus arquivos de configuração de proxy personalizados, se você tiver algum.

   c. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/postdeploy` do pacote de origem (`.platform/confighooks/postdeploy` para uma implantação de configuração).

Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores

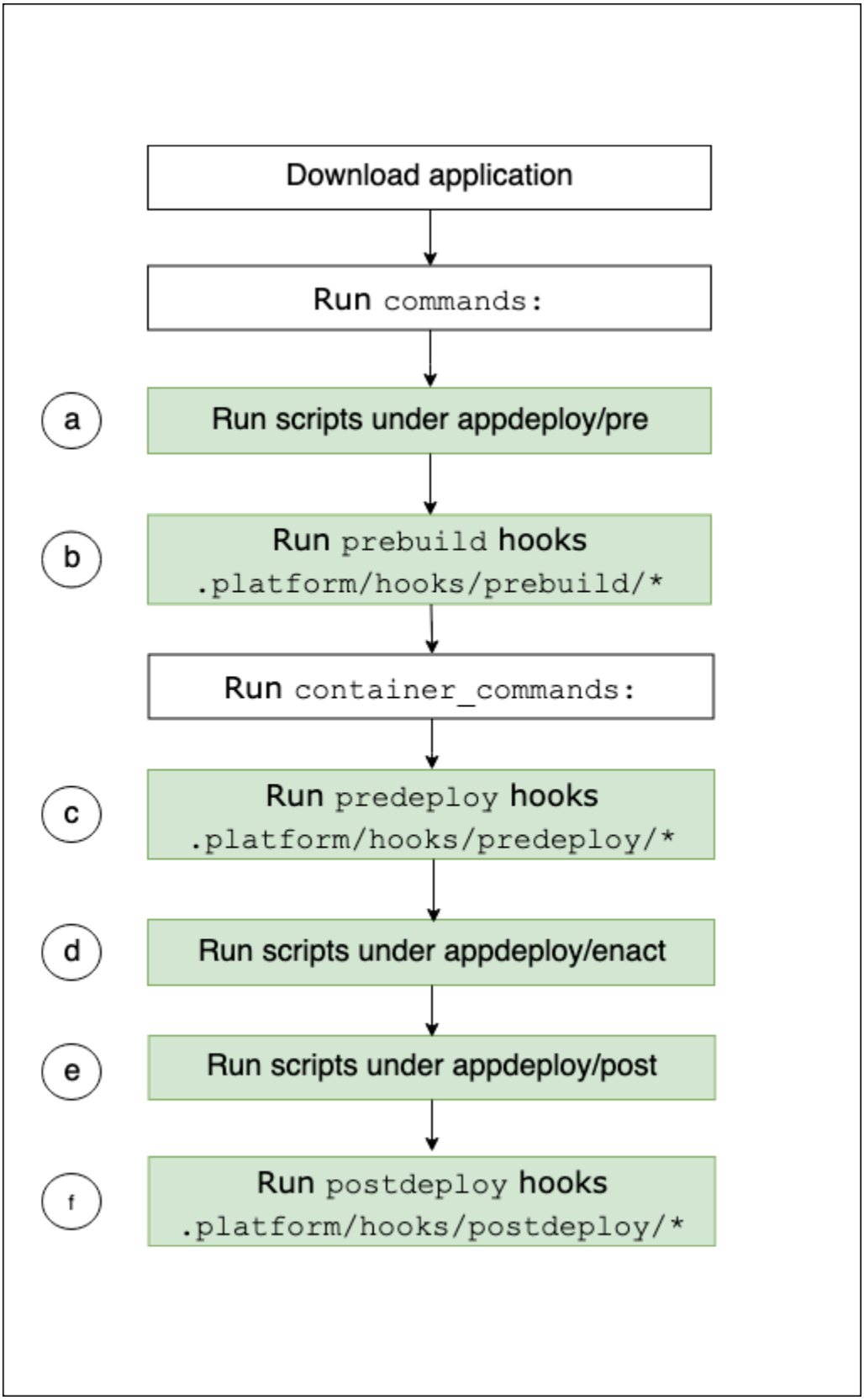
A seção anterior descreve os recursos de extensibilidade compatíveis ao longo das fases do fluxo de trabalho de implantação da aplicação. Há algumas diferenças para ramificação da plataforma Docker no ECS em execução no Amazon Linux 2 e versões posteriores. Esta seção explica como esses conceitos se aplicam a essa ramificação específica da plataforma.

Como existem muitas maneiras de estender a plataforma do ambiente, é útil saber o que acontece sempre que o Elastic Beanstalk provisiona ou implementa uma instância. O diagrama a seguir mostra todo esse fluxo de trabalho de implantação para um ambiente baseado nas ramificações de plataforma ECS em execução no Amazon Linux 2 e ECS em execução no Amazon Linux 2023. Ele representa as diferentes fases de uma implantação e as etapas que o Elastic Beanstalk realiza em cada fase.

Ao contrário do fluxo de trabalho descrito na seção anterior, a fase de configuração da implantação não é compatível com os seguintes recursos de extensibilidade: comandos `Buildfile`, comandos `Procfile`, configuração de proxy reverso.

Observações

- O diagrama não representa o conjunto completo de etapas que o Elastic Beanstalk realiza em instâncias do ambiente durante a implantação. Nós fornecemos este diagrama como ilustração, para fornecer a você a ordem e o contexto para a execução de suas personalizações.
- Para simplificar, o diagrama menciona apenas os subdiretórios de hook `.platform/hooks/*` (para implantações de aplicações), e não os subdiretórios de hook `.platform/confighooks/*` (para implantações de configurações). Os hooks nos últimos subdiretórios são executados durante exatamente as mesmas etapas que os hooks nos subdiretórios correspondentes mostrados no diagrama.



A lista a seguir detalha as etapas do fluxo de trabalho de implantação.

a. Executa todos os arquivos executáveis encontrados no diretório `appdeploy/pre` em `EBhooksDir`.

b. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/prebuild` do pacote de origem (`.platform/confighooks/prebuild` para uma implantação de configuração).

c. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/predeploy` do pacote de origem (`.platform/confighooks/predeploy` para uma implantação de configuração).

d. Executa todos os arquivos executáveis encontrados no diretório `appdeploy/enact` em `EBhooksDir`.

e. Executa todos os arquivos executáveis encontrados no diretório `appdeploy/post` em `EBhooksDir`.

f. Executa todos os arquivos executáveis encontrados no diretório `.platform/hooks/postdeploy` do pacote de origem (`.platform/confighooks/postdeploy` para uma implantação de configuração).

A referência a `EBhooksDir` representa o caminho do diretório de hooks da plataforma. Para recuperar o nome do caminho do diretório, use a ferramenta de script get-config na linha de comando da instância do ambiente, como mostrado:

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k EBhooksDir
```

Ferramentas de script de plataforma para ambientes Elastic Beanstalk

Este tópico descreve as ferramentas que AWS Elastic Beanstalk fornecem para ambientes que usam plataformas Amazon Linux. As ferramentas estão localizadas nas EC2 instâncias Amazon dos ambientes do Elastic Beanstalk.

get-config

Use a `get-config` ferramenta para recuperar valores de variáveis de ambiente em texto simples e outras informações de plataforma e instância. A ferramenta está disponível em `/opt/elasticbeanstalk/bin/get-config`.

comandos get-config

Cada comando da ferramenta `get-config` retorna um tipo específico de informação. Use a sintaxe a seguir para executar os comandos de qualquer uma das ferramentas.

```
$ /opt/elasticbeanstalk/bin/get-config command [ options ]
```

O exemplo a seguir executa o comando `environment`:

```
$ /opt/elasticbeanstalk/bin/get-config environment -k PORT
```

Dependendo do comando e das opções que você escolher, a ferramenta retorna um objeto (JSON ou YAML) com pares de chave-valor ou um único valor.

Você pode testar `get-config` usando SSH para se conectar a uma EC2 instância em seu ambiente do Elastic Beanstalk.

> **Note**
>
> Quando você executa `get-config` para teste, alguns comandos podem exigir privilégios de usuário raiz para acessar as informações subjacentes. Se você receber um erro de permissão de acesso, execute o comando novamente em `sudo`.
> Você não precisa adicionar `sudo` ao usar a ferramenta nos scripts implantados no ambiente. O Elastic Beanstalk executa todos os seus scripts como o usuário raiz.

As seções a seguir descrevem os comandos da ferramenta.

optionsettings: opções de configuração

O comando `get-config optionsettings` retorna um objeto listando as opções de configuração definidas no ambiente e usadas pela plataforma em instâncias do ambiente. Elas são organizadas por namespace.

```
$ /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"}}
```

Para retornar o valor de uma opção de configuração específica, use a opção `--namespace` (`-n`) para especificar um namespace, e a opção `--option-name` (`-o`) para especificar o nome de uma opção.

```
$ /opt/elasticbeanstalk/bin/get-config optionsettings -
n aws:elasticbeanstalk:container:php:phpini -o memory_limit
256M
```

environment : propriedades do ambiente

O `get-config environment` comando retorna um objeto contendo uma lista de propriedades do ambiente, incluindo as configuradas pelo usuário e fornecidas pelo Elastic Beanstalk. As propriedades configuradas pelo usuário são definidas no console como texto simples ou com o aws:elasticbeanstalk:application:environment namespace da opção de configuração.

```
$ /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
```

Por exemplo, o Elastic Beanstalk fornece propriedades de ambiente para conexão a uma instância de banco de dados integrada do Amazon RDS (por exemplo, `RDS_HOSTNAME`). Essas propriedades de conexão do RDS aparecem na saída do `get-config environment`. No entanto, elas não aparecem na saída do `get-config optionsettings`. Isso ocorre porque elas não foram definidas nas opções de configuração.

Para retornar uma propriedade específica do ambiente, use a opção `--key` (`-k`) para especificar uma chave de propriedade.

```
$ /opt/elasticbeanstalk/bin/get-config environment -k TESTPROPERTY
testvalue
```

> **Note**
>
> A `get-config` ferramenta não pode recuperar variáveis de ambiente que armazenam segredos. Para obter mais informações sobre como recuperar programaticamente valores

de armazenamentos secretos ou de parâmetros, consulte ou. Usando o Secrets Manager Usando Systems Manager Parameter Store

contêiner: valores de configuração na instância

O comando `get-config container` retorna um objeto que lista valores de configuração de ambiente e plataforma para instâncias de ambiente.

O exemplo a seguir mostra a saída do comando em um ambiente Tomcat do 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"}
```

Para retornar o valor de uma chave específica, use a opção `--key` (`-k`) para especificar a chave.

```
$ /opt/elasticbeanstalk/bin/get-config container -k environment_name
myenv-1da84946
```

addons: valores de configuração de complementos

O comando `get-config addons` retorna um objeto que contém informações de configuração adicionadas ao complemento do ambiente. Use-o para recuperar a configuração de um banco de dados do Amazon RDS associado ao 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"}}}
```

É possível restringir o resultado de duas maneiras. Para recuperar valores de um complemento específico, use a opção `--add-on` (`-a`) para especificar o nome do complemento.

```
$ /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"}}
```

Para retornar o valor de uma chave específica dentro de um complemento, adicione a opção `--key` (`-k`) para especificar a chave.

```
$ /opt/elasticbeanstalk/bin/get-config addons -a rds -k RDS_DB_NAME
ebdb
```

platformconfig: valores de configuração constantes

O comando `get-config platformconfig` retorna um objeto que contém informações de configuração da plataforma que são constantes para a versão da plataforma. A saída é a mesma em todos os ambientes que executam a mesma versão da plataforma. O objeto de saída do comando tem dois objetos incorporados:

- `GeneralConfig`: contém informações constantes nas versões mais recentes de todas as ramificações de plataformas Amazon Linux 2 e Amazon Linux 2023.
- `PlatformSpecificConfig`: contém informações que são constantes para a versão da plataforma e que são específicas para ela.

O exemplo a seguir mostra a saída do comando em um ambiente que usa o Tomcat 8.5 executando a ramificação da plataforma 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"}}
```

Para retornar o valor de uma chave específica, use a opção `--key` (`-k`) para especificar a chave. Essas chaves são exclusivas entre os dois objetos incorporados. Não é necessário especificar o objeto que contém a chave.

```
$ /opt/elasticbeanstalk/bin/get-config platformconfig -k AppStagingDir
/var/app/staging/
```

Opções de saída get-config

Use a opção `--output` para especificar o formato do objeto de saída. Os valores válidos são `JSON` (padrão) e `YAML`. Essa é uma opção global. Você deve especificá-la antes do nome do comando.

O exemplo a seguir retorna valores de opção de configuração no 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**
>
> A ferramenta `pkg-repo` não está disponível para ambientes baseados em plataformas Amazon Linux 2023. No entanto, você pode aplicar manualmente as atualizações do pacote e do sistema operacional a uma instância AL2 023. Para obter mais informações, consulte Gerenciamento de pacotes e atualizações do sistema operacional no Guia do usuário do Amazon Linux 2023

Em algumas circunstâncias urgentes, talvez seja necessário atualizar suas EC2 instâncias da Amazon com um patch de segurança do Amazon Linux 2 que ainda não foi lançado com as versões necessárias da plataforma Elastic Beanstalk. Você não pode executar uma atualização manual em seus ambientes do Elastic Beanstalk por padrão. Isso ocorre porque as versões da plataforma estão bloqueadas em uma versão específica do repositório do Amazon Linux 2. Esse bloqueio garante que as instâncias executem versões de software compatíveis e consistentes. Para casos urgentes, a ferramenta `pkg-repo` permite uma solução alternativa para atualizar manualmente os pacotes yum no Amazon Linux 2 se você precisar instalá-los em um ambiente antes de serem lançados em uma nova versão da plataforma do Elastic Beanstalk.

A ferramenta `pkg-repo` das plataformas Amazon Linux 2 permitem desbloquear os repositórios dos pacotes `yum`. Em seguida, você pode executar manualmente um yum update para um patch de segurança. Por outro lado, você pode seguir a atualização usando a ferramenta para bloquear os repositórios de pacotes yum para evitar atualizações adicionais. A `pkg-repo` ferramenta está disponível no `/opt/elasticbeanstalk/bin/pkg-repo` diretório de todas as EC2 instâncias em seus ambientes do Elastic Beanstalk.

As alterações usando a `pkg-repo` ferramenta são feitas somente na EC2 instância em que a ferramenta é usada. Elas não afetam outras instâncias nem impedem atualizações futuras do ambiente. Os exemplos fornecidos posteriormente neste tópico explicam como aplicar as alterações em todas as instâncias chamando os comandos `pkg-repo` de scripts e arquivos de configuração.

> **Warning**
>
> Nós não recomendamos essa ferramenta para a maioria dos usuários. Quaisquer alterações manuais aplicadas a uma versão desbloqueada da plataforma é considerada fora de banda. Essa opção só é viável para os usuários em circunstâncias urgentes que podem aceitar os seguintes riscos:
>
> - Não é possível garantir que as versões dos pacotes sejam consistentes entre todas as instâncias de seus ambientes.
> - Não há garantias de que ambientes modificados usando-se a ferramenta `pkg-repo` funcionem corretamente. Eles não foram testados e verificados em plataformas compatíveis com o Elastic Beanstalk.
>
> Recomendamos a aplicação de práticas recomendadas que incluem planos de teste e backout. Para ajudar a facilitar as melhores práticas, você pode usar o console do Elastic Beanstalk e o EB CLI para clonar um ambiente e trocar o ambiente. URLs Para obter mais informações sobre como usar essas operações, consulte Implantações azuis/verdes no capítulo Gerenciar ambientes deste guia.

Se você planeja editar manualmente os arquivos de configuração do repositório yum, execute a ferramenta `pkg-repo` primeiro. A ferramenta `pkg-repo` talvez não funcione como o esperado em um ambiente Amazon Linux 2 com arquivos de configuração do repositório yum editados manualmente. Isso ocorre porque a ferramenta pode não reconhecer as alterações de configuração.

Para obter mais informações sobre o repositório de pacotes Amazon Linux, consulte o tópico Package repository no Amazon EC2 User Guide.

comandos pkg-repo

Use a sintaxe a seguir para executar os comandos da ferramenta `pkg-repo`.

```
$ /opt/elasticbeanstalk/bin/pkg-repo command [options]
```

Os comandos de `pkg-repo` são os seguintes:

- lock: bloqueia os repositórios de pacotes `yum` em uma versão específica
- unlock: desbloqueia os repositórios de pacotes `yum` de uma versão específica
- status: lista todos os repositórios de pacotes `yum` e o status de bloqueio atual deles
- help: mostra ajuda geral ou ajuda para um comando

As opções a seguir se aplicam aos seguintes comandos.

- `lock`, `unlock` e `status` - opções: `-h`,`--help`, ou nenhum (padrão).
- `help` – opções: `lock`, `unlock`, `status` ou nenhum (padrão).

O exemplo a seguir executa o 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
```

O exemplo a seguir executa o 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
```

O exemplo a seguir executa o 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
```

O exemplo a seguir executa o comando help para o comando lock.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help lock
```

O exemplo a seguir executa o comando help para a ferramenta `pkg-repo`.

```
$ sudo /opt/elasticbeanstalk/bin/pkg-repo help
```

É possível testar `pkg-repo` usando SSH para se conectar a uma instância no ambiente do Elastic Beanstalk. Uma opção de SSH é o comando CLI do EB eb ssh.

> **Note**
>
> A ferramenta `pkg-repo` requer privilégios de usuário root para execução. Se você receber um erro de permissão de acesso, execute o comando novamente em `sudo`.
> Você não precisa adicionar `sudo` ao usar a ferramenta nos scripts ou arquivos de configuração que você implanta no seu ambiente. O Elastic Beanstalk executa todos os seus scripts como o usuário raiz.

exemplos de pkg-repo

A seção anterior fornece exemplos de linha de comando para testes em uma EC2 instância individual de um ambiente do Elastic Beanstalk. Essa abordagem pode ser útil para testes. No entanto, ela atualiza apenas uma instância por vez, portanto, não é prático aplicar alterações em todas as instâncias em um ambiente.

Uma abordagem mais pragmática é usar scripts de hook de plataforma ou um arquivo de configuração `.ebextensions` para aplicar as alterações em todas as instâncias de forma consistente.

O exemplo a seguir chama o `pkg-repo` de um arquivo de configuração da pasta `.ebextensions`. O Elastic Beanstalk executa os comandos no arquivo `update_package.config` quando você implanta seu pacote-fonte de aplicação.

```
.ebextensions
### update_package.config
```

Para receber a versão mais recente do pacote de docker, essa configuração especifica o pacote de docker no 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
```

Essa configuração não especifica nenhum pacote no comando yum update. A consequência é que todas as atualizações disponíveis são aplicadas.

```
### 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
```

O exemplo a seguir chama o `pkg-repo` de um script bash como um hook de plataforma. O Elastic Beanstalk executa o arquivo de script `update_package.sh` localizado no subdiretório `prebuild`.

```
.platform
### hooks
    ### prebuild
        ### update_package.sh
```

Para receber a versão mais recente do pacote de docker, esse script especifica o pacote do docker no comando yum update. Se o nome do pacote for omitido, todas as atualizações disponíveis serão aplicadas. O exemplo do arquivo de configuração anterior demonstra isso.

```
### 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 (Somente Amazon Linux AMI)

Nas ramificações de plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2), o Elastic Beanstalk fornece uma ferramenta adicional, que é `download-source-bundle`. Use essa ferramenta para baixar o código-fonte da sua aplicação durante a implantação da plataforma. A ferramenta está disponível em `/opt/elasticbeanstalk/bin/download-source-bundle`.

O script de exemplo `00-unzip.sh` está localizado na pasta `appdeploy/pre` em instâncias do ambiente. Ele demonstra como usar `download-source-bundle` para baixar o código-fonte da aplicação para a pasta `/opt/elasticbeanstalk/deploy/appsource` durante a implantação.

Estender as plataformas Linux do Elastic Beanstalk

Este tópico descreve como estender suas plataformas Linux com seus próprios comandos, scripts, software e configurações. Talvez seja necessário estender a plataforma para alterar o servidor proxy padrão e a configuração. Ou talvez você precise personalizar a forma como a plataforma constrói ou inicia sua aplicação.

Tópicos

- Buildfile e Procfile
- Hooks de plataforma
- Arquivos de configuração
- Configuração de proxy reverso
- Exemplo de aplicativo com extensões

Buildfile e Procfile

Algumas plataformas permitem que você personalize a maneira como cria ou prepara seu aplicativo e especifique os processos que executam seu aplicativo. Cada tópico de plataforma individual

menciona especificamente Buildfile e/ou Procfile se a plataforma oferecer suporte a eles. Procure sua plataforma específica em Plataformas.

Para todas as plataformas compatíveis, a sintaxe e a semântica são idênticas e descritas nesta página. Tópicos individuais da plataforma mencionam o uso específico desses arquivos para criar e executar aplicativos em suas respectivas linguagens.

Buildfile

Para especificar um comando personalizado de compilação e configuração para o aplicativo, coloque um arquivo chamado `Buildfile` no diretório raiz da origem do aplicativo. O nome do arquivo diferencia maiúsculas de minúsculas. Use a seguinte sintaxe para o `Buildfile`.

```
<process_name>: <command>
```

O comando no `Buildfile` deve corresponder à expressão regular: `^[A-Za-z0-9_-]+:\s*[^\s].*$`.

O Elastic Beanstalk não monitora a aplicação executada com um `Buildfile`. Use um `Buildfile` para comandos que são executados por breves períodos e são encerrados após a conclusão das tarefas. Para processos de aplicativo de longa execução que não devem ser encerrados, use o Procfile.

Todos os caminhos no `Buildfile` são relativos à raiz do pacote de origem. No seguinte exemplo de um `Buildfile`, o `build.sh` é um script de shell localizado na raiz do pacote de origem.

Example Buildfile

```
make: ./build.sh
```

Se você quiser fornecer etapas de compilação personalizadas, recomendamos que você use os hooks de plataforma `predeploy` para qualquer coisa, exceto os comandos mais simples, em vez de um `Buildfile`. Os hooks de plataforma permitem scripts mais avançados e melhor tratamento de erros. Os hooks de plataforma são descritos na próxima seção.

Procfile

Para especificar comandos personalizados para iniciar e executar o aplicativo, coloque um arquivo chamado `Procfile` no diretório raiz da origem do aplicativo. O nome do arquivo diferencia

maiúsculas de minúsculas. Use a seguinte sintaxe para o `Procfile`. Você pode especificar um ou mais comandos.

```
<process_name1>: <command1>
<process_name2>: <command2>
...
```

Cada linha no `Procfile` deve corresponder à expressão regular: `^[A-Za-z0-9_-]+:\s*[^\s].*$`.

Use um `Procfile` para processos de aplicações de longa execução que não devem ser fechadas. O Elastic Beanstalk espera que os processos sejam executados a partir do `Procfile` para serem executados continuamente. O Elastic Beanstalk monitora esses processos e reinicia qualquer processo que seja encerrado. Para processos de curta execução, use um Buildfile.

Todos os caminhos no `Procfile` são relativos à raiz do pacote de origem. O exemplo do `Procfile` a seguir define três processos. O primeiro, chamado `web` no exemplo, é o principal aplicativo web.

Example Procfile

```
web: bin/myserver
cache: bin/mycache
foo: bin/fooapp
```

O Elastic Beanstalk configura o servidor de proxy para encaminhar solicitações à sua aplicação Web principal na porta 5000, e esse número de porta pode ser configurado. Um uso comum para um `Procfile` é passar esse número de porta para sua aplicação como um argumento de comando. Para obter detalhes sobre a configuração do proxy, consulte Configuração de proxy reverso.

O Elastic Beanstalk captura streams de saída padrão e erros dos processos `Procfile` em arquivos de log. O Elastic Beanstalk fornece nomes aos arquivos de log após o processo e os armazena no `/var/log`. Por exemplo, o processo `web` do exemplo anterior gera logs chamados `web-1.log` e `web-1.error.log` para `stdout` e `stderr`, respectivamente.

Hooks de plataforma

Os hooks de plataforma são criados especificamente para estender a plataforma do ambiente. Eles são scripts personalizados e outros arquivos executáveis que são implantados como parte do código-fonte da aplicação, e são executados pelo Elastic Beanstalk durante vários estágios de provisionamento da instância.

> **Note**
>
> Os hooks de plataforma não são compatíveis com as versões da plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2).

Hooks de plataforma de implantação de aplicações

Uma implantação de aplicação ocorre quando você fornece um novo pacote de origem para implantação ou quando faz uma alteração de configuração que exige o encerramento e a recriação de todas as instâncias do ambiente.

Para fornecer hooks de plataforma executados durante a implantação de uma aplicação, coloque os arquivos no diretório `.platform/hooks` do pacote de origem, em um dos seguintes subdiretórios.

- `prebuild`: estes arquivos serão executados depois que o mecanismo da plataforma do Elastic Beanstalk fizer download e extrair o pacote de origem da aplicação, e antes de configurar a aplicação e o servidor Web.

  Os arquivos `prebuild` são executados depois da execução de comandos encontrados na seção commands, de qualquer arquivo de configuração, e antes da execução de comandos `Buildfile`.
- `predeploy`: estes arquivos são executados depois que o mecanismo da plataforma do Elastic Beanstalk configura a aplicação e o servidor Web, e antes de implantá-los em seu local final de runtime.

  Os arquivos `predeploy` são executados depois da execução dos comandos encontrados na seção container_commands de qualquer arquivo de configuração, e antes da execução de comandos `Procfile`.
- `postdeploy`: estes arquivos são executados depois que o mecanismo da plataforma do Elastic Beanstalk implanta a aplicação e o servidor de proxy.

  Esta é a última etapa do fluxo de trabalho de implantação.

Hooks de plataforma de implantação de configuração

Uma implantação de configuração ocorre quando você faz alterações de configuração que apenas atualizam instâncias do ambiente sem recriá-las. As atualizações de opção a seguir resultam em uma atualização da configuração.

- Propriedades do ambiente e configurações específicas da plataforma
- Arquivos estáticos
- AWS X-Ray daemon
- Streaming e armazenamento de logs
- Porta da aplicação (para obter detalhes, consulte Configuração de proxy reverso)

Para fornecer hooks que são executados durante a implantação de uma configuração, coloque-os no diretório `.platform/confighooks` do pacote de origem. São aplicados os mesmos três subdiretórios que aqueles aplicados para os hooks de implantação de aplicações.

Mais sobre hooks de plataforma

Os arquivos de hook podem ser arquivos binários ou arquivos de script que começam com uma linha `#!` contendo seu caminho de interpretador, como `#!/bin/bash`. Todos os arquivos devem ter permissão de execução. Use `chmod +x` para definir a permissão de execução em seus arquivos de hook. Em todas as versões de plataforma baseadas no Amazon Linux 2023 e no Amazon Linux 2 lançadas a partir de 29 de abril de 2022, o Elastic Beanstalk concede automaticamente permissões de execução para todos os scripts de hook da plataforma. Nesse caso, não é necessário conceder manualmente as permissões de execução. Para obter uma lista dessas versões da plataforma, consulte as notas de versão do Linux de 29 de abril de 2022 no Guia de notas de versão do AWS Elastic Beanstalk .

O Elastic Beanstalk executa arquivos em cada um desses diretórios em ordem lexicográfica de nomes de arquivos. Todos os arquivos são executados como o usuário `root`. O diretório de trabalho atual (cwd) para hooks de plataforma é o diretório raiz do aplicativo. Para os arquivos `prebuild` e `predeploy` é o diretório de preparação do aplicativo, e para os arquivos `postdeploy` é o diretório atual do aplicativo. Se um dos arquivos falhar (terminar com um código de saída diferente de zero), a implantação será interrompida e falhará.

Um script de texto de hooks da plataforma pode falhar se contiver caracteres de quebra de linha retorno de carro/alimentação de linha (CRLF). Se um arquivo foi salvo em um host Windows e depois transferido para um servidor Linux, ele pode conter quebras de linha CRLF do Windows. Para plataformas lançadas a partir de 29 de dezembro de 2022, o Elastic Beanstalk converte automaticamente os caracteres CRLF do Windows em caracteres de quebra de linha alimentação de linha (LF) do Linux em arquivos de texto de hooks de plataforma. Se a aplicação for executada em qualquer plataforma Amazon Linux 2 lançada antes dessa data, você precisará converter os caracteres CRLF do Windows em caracteres LF do Linux. Uma forma de fazer isso é criar e salvar

o arquivo de script em um host Linux. Ferramentas que convertem esses caracteres também estão disponíveis na internet.

Os arquivos de hook têm acesso a todas as propriedades de ambiente definidas nas opções da aplicação e às variáveis de ambiente do sistema `HOME`, `PATH` e `PORT`.

Para obter valores de variáveis de ambiente e outras opções de configuração nos scripts de hook de plataforma, é possível usar o utilitário `get-config` que o Elastic Beanstalk fornece em instâncias do ambiente. Para obter detalhes, consulte the section called “Ferramentas de script de plataforma”.

Arquivos de configuração

Você pode adicionar arquivos de configuração ao diretório `.ebextensions` do código-fonte da sua aplicação para configurar vários aspectos do ambiente do Elastic Beanstalk. Entre outras coisas, os arquivos de configuração permitem personalizar software e outros arquivos nas instâncias do ambiente e executar comandos de inicialização nas instâncias. Para obter mais informações, consulte the section called “Servidor do Linux”.

Você também pode definir opções de configuração usando arquivos de configuração. Muitas das opções controlam o comportamento da plataforma, e algumas dessas opções são específicas da plataforma.

Nas plataformas do Amazon Linux 2 e Amazon Linux 2023, recomendamos o uso de hooks de plataforma, Buildfile e Procfile para configurar e executar código personalizado em suas instâncias de ambiente durante o provisionamento de instâncias. Esses mecanismos são descritos nas seções anteriores desta página. Você ainda pode usar comandos e os comandos de contêiner em arquivos de configuração `.ebextensions`, apesar de não ser fácil trabalhar com eles. Por exemplo, escrever scripts de comando dentro de um arquivo YAML pode ser um desafio do ponto de vista da sintaxe. Você ainda precisa usar arquivos `.ebextensions` de configuração para qualquer script que precise de uma referência a um AWS CloudFormation recurso.

Configuração de proxy reverso

Todas as versões de plataformas Amazon Linux 2 e Amazon Linux 2023 usam o nginx como servidor de proxy reverso padrão. As plataformas Tomcat, Node.js, PHP e Python também oferecem suporte ao Apache HTTPD como uma alternativa. Para selecionar o Apache nessas plataformas, defina a opção `ProxyServer` no namespace `aws:elasticbeanstalk:environment:proxy` como `apache`. Todas as plataformas habilitam a configuração do servidor de proxy de forma uniforme, conforme descrito nesta seção.

> **Note**
>
> Nas versões da plataforma da AMI do Amazon Linux (anteriores ao Amazon Linux 2), talvez seja necessário configurar servidores de proxy de forma diferente. É possível encontrar esses detalhes legados nos respectivos tópicos de plataforma neste guia.

O Elastic Beanstalk configura o servidor de proxy nas instâncias do ambiente para encaminhar o tráfego da Web para a aplicação Web principal no URL raiz do ambiente; por exemplo, `http://my-env.elasticbeanstalk.com`.

Por padrão, o Elastic Beanstalk configura o proxy para encaminhar solicitações recebidas na porta 80 para a aplicação Web principal na porta 5000. Você pode configurar esse número de porta definindo a propriedade `PORT` do ambiente usando o namespace `aws:elasticbeanstalk:application:environment` em um arquivo de configuração, conforme mostrado no exemplo a seguir.

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  PORT
    value:  <main_port_number>
```

Para obter mais informações sobre como configurar variáveis de ambiente para o aplicativo, consulte the section called “Configurações de opção”.

Seu aplicativo deve ter como base a porta que está configurada para ele no proxy. Se você alterar a porta padrão usando a propriedade `PORT` do ambiente, seu código poderá acessá-la ao ler o valor da variável `PORT` do ambiente. Por exemplo, chame `os.Getenv("PORT")` em Go ou `System.getenv("PORT")` em Java. Se você configurar o proxy para enviar tráfego para vários processos de aplicativo, será possível configurar várias propriedades de ambiente e usar seus valores na configuração de proxy e no código do aplicativo. Outra opção é passar o valor da porta para o processo como um argumento de comando no `Procfile`. Para ter mais informações, consulte Buildfile e Procfile.

Configurar o nginx

O Elastic Beanstalk usa o nginx como proxy reverso padrão para mapear sua aplicação para o balanceador de carga do Elastic Load Balancing. O Elastic Beanstalk oferece uma configuração de nginx padrão que pode ser estendida ou substituída completamente por sua própria configuração.

> **Note**
>
> Ao adicionar ou editar um arquivo de configuração `.conf` do nginx, codifique-o como UTF-8.

Para estender a configuração nginx padrão do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta denominada `.platform/nginx/conf.d/` no pacote de origem da sua aplicação. A configuração nginx do Elastic Beanstalk inclui arquivos `.conf` nesta pasta automaticamente.

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- conf.d
|           `-- myconf.conf
`-- other source files
```

Para substituir completamente a configuração nginx padrão do Elastic Beanstalk, inclua uma configuração em seu pacote de origem em `.platform/nginx/nginx.conf`.

```
~/workspace/my-app/
|-- .platform
|   `-- nginx
|       `-- nginx.conf
`-- other source files
```

Se você substituir a configuração nginx do Elastic Beanstalk, adicione a linha a seguir ao seu `nginx.conf` para extrair as configurações do Elastic Beanstalk para Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk, mapeamentos automáticos de aplicações e arquivos estáticos.

```
include conf.d/elasticbeanstalk/*.conf;
```

Configurar o Apache HTTPD

As plataformas Tomcat, Node.js, PHP e Python permitem escolher o servidor de proxy Apache HTTPD como uma alternativa ao nginx. Esse não é o padrão. O exemplo a seguir configura o Elastic Beanstalk para usar o Apache HTTPD.

Example .ebextensions/httpd-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

É possível estender a configuração padrão do Apache do Elastic Beanstalk com seus arquivos de configuração adicionais. Também há opção de substituir completamente a configuração padrão do Apache do Elastic Beanstalk.

Para estender a configuração padrão do Apache do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta chamada `.platform/httpd/conf.d` no pacote de origem da aplicação. A configuração do Apache do Elastic Beanstalk inclui os arquivos `.conf` nessa pasta automaticamente.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
|-- .platform
|   -- httpd
|      -- conf.d
|         -- port5000.conf
|         -- ssl.conf
-- index.jsp
```

Por exemplo, a configuração do Apache 2.4 a seguir adiciona um ouvinte na 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>
```

Para substituir completamente a configuração padrão do Apache do Elastic Beanstalk, inclua uma configuração no pacote de origem em `.platform/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   -- httpd-proxy.config
|-- .platform
|   `-- httpd
|       `-- conf
|           `-- httpd.conf
`-- index.jsp
```

> **Note**
>
> Se você substituir a configuração do Apache do Elastic Beanstalk, adicione as linhas a seguir ao `httpd.conf` para extrair as configurações do Elastic Beanstalk para Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk, mapeamentos automáticos de aplicações e arquivos estáticos.
>
> ```
> IncludeOptional conf.d/elasticbeanstalk/*.conf
> ```

Exemplo de aplicativo com extensões

O exemplo a seguir demonstra um pacote de origem de aplicação com vários recursos de capacidade de extensão compatíveis com as plataformas Amazon Linux 2 e Amazon Linux 2023 do Elastic Beanstalk: um `Procfile` e arquivos de configuração `.ebextensions`, hooks personalizados e arquivos de configuração de 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
|-- 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**
>
> Algumas dessas extensões não são compatíveis com as versões da plataforma de AMI do Amazon Linux (anteriores ao Amazon Linux 2).

Implantar aplicações .NET Windows com o Elastic Beanstalk

> **Confira o.NET no AWS Developer Center**
>
> Você já passou por nosso Centro de desenvolvedores .Net? É nosso balcão único para tudo o que está ligado a o.NET AWS.
> Para obter mais informações, consulte o.NET no AWS Developer Center.

Este capítulo fornece instruções para configurar e implantar seus aplicativos Web ASP.NET e .NET Core do Windows em. AWS Elastic Beanstalk O Elastic Beanstalk facilita a implantação, o gerenciamento e a escalabilidade de seus aplicativos web.NET (Windows) usando a Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk, ou o. AWS CLI APIs

Este capítulo fornece estes seguintes:

- QuickStart para o.NET Core no Windows— Step-by-step instruções para criar e implantar um aplicativo Hello World .NET Core Windows usando o EB CLI.
- QuickStart para ASP.NET— Step-by-step instruções para criar e implantar um aplicativo Hello World ASP.NET usando o. AWS Toolkit for Visual Studio

Se precisar de ajuda com o desenvolvimento de aplicações .NET Core Windows, há vários lugares onde você pode ir:

- Fórum de desenvolvimento .NET: poste suas perguntas e receba feedback.
- Centro de desenvolvedores .NET: parada única para código de amostra, documentação, ferramentas e recursos adicionais.
- AWS Documentação do SDK for .NET — Leia sobre como configurar o SDK e executar amostras de código, recursos do SDK e informações detalhadas sobre as operações de API do SDK.

> **Note**
>
> Essa plataforma não oferece suporte a ambientes de trabalho. Para obter mais detalhes, consulte Ambientes de operador do Elastic Beanstalk.

Tópicos

- QuickStart: Implante um aplicativo.NET Core no Windows no Elastic Beanstalk
- QuickStart: Implante um aplicativo ASP.NET no Elastic Beanstalk
- Configurar seu ambiente de desenvolvimento .NET
- Usar o Elastic Beanstalk .NET na plataforma Windows
- Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações .NET
- O AWS Toolkit for Visual Studio
- Migrar a aplicação .NET no local para o Elastic Beanstalk
- Recomendações para componentes retirados do Windows Server no Elastic Beanstalk

QuickStart: Implante um aplicativo.NET Core no Windows no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo.NET Core no Windows e sua implantação em um AWS Elastic Beanstalk ambiente.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: Criar um aplicação .NET Core no Windows
- Etapa 2: executar a aplicação localmente

- Etapa 3: implantar sua aplicação .NET Core no Windows com a EB CLI
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços de que você precisa.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Os comandos são mostrados em lisetiquetaens precedidas por um símbolo de prompt (**>**) e o nome do diretório atual, quando apropriado.

```
C:\eb-project> this is a command
this is output
```

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

.NET Core no Windows

Se não tiver o .NET SDK instalado na máquina local, você poderá instalá-lo seguindo o link Baixar .NET no site de Documentação do .NET.

Verifique sua instalação do .NET SDK executando o comando a seguir.

```
C:\> dotnet --info
```

Etapa 1: Criar um aplicação .NET Core no Windows

Crie um diretório do projeto.

```
C:\> mkdir eb-dotnetcore
C:\> cd eb-dotnetcore
```

Em seguida, crie um exemplo de aplicativo de serviço RESTful web Hello World executando os comandos a seguir.

```
C:\eb-dotnetcore> dotnet new web --name HelloElasticBeanstalk
C:\eb-dotnetcore> cd HelloElasticBeanstalk
```

Etapa 2: executar a aplicação localmente

Execute o seguinte comando para executar sua aplicação localmente.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> dotnet run
```

A saída deve ser semelhante ao texto a seguir.

```
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**
>
> O comando `dotnet` seleciona uma porta aleatoriamente ao executar a aplicação localmente. Neste exemplo, a porta é 5228. Ao implantar a aplicação no ambiente Elastic Beanstalk, ela será executada na porta 5000.

Insira o endereço URL `http://localhost:`*`port`* no navegador. Para este exemplo específico, o comando é `http://localhost:5228`. O navegador deve exibir “Hello World!”.

Etapa 3: implantar sua aplicação .NET Core no Windows com a EB CLI

Execute os seguintes comandos para criar um ambiente Elastic Beanstalk para essa aplicação.

Para criar um ambiente e implantar sua aplicação .NET Core no Windows

1. Execute os seguintes comandos no diretório `HelloElasticBeanstalk` para publicar e compactar sua aplicação.

```
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. Crie um novo arquivo no `HelloElasticBeanstalk` chamado `aws-windows-deployment-manifest.json`com o seguinte conteúdo:

```
{
    "manifestVersion": 1,
    "deployments": {
        "aspNetCoreWeb": [
        {
            "name": "test-dotnet-core",
            "parameters": {
                "appBundle": "site.zip",
                "iisPath": "/",
                "iisWebSite": "Default Web Site"
            }
        }
        ]
    }
}
```

3. Inicialize o repositório da EB CLI com o comando eb init:

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb init -p iis dotnet-windows-server-
tutorial --region us-east-2
```

Esse comando cria um aplicação chamado `dotnet-windows-server-tutoriale` configura seu repositório local para criar ambientes com a versão mais recente da plataforma do Windows Server.

4. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb create dotnet-windows-server-env
```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb open
```

Parabéns! Você implantou uma aplicação .NET Core no Windows com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 5: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
C:\eb-dotnetcore\HelloElasticBeasntalk> eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

> **Explore mais tutoriais**
>
> Se quiser experimentar outros tutoriais com diferentes aplicações de exemplo, consulte QuickStart para ASP.NET.

Depois de implantar um ou dois aplicações de amostra e estar pronto para começar a desenvolver e executar aplicações .NET Core no Windows localmente, consulte Configurar seu ambiente de desenvolvimento .NET

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

QuickStart: Implante um aplicativo ASP.NET no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo ASP.NET e sua implantação em um ambiente. AWS Elastic Beanstalk

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação ASP.NET
- Etapa 2: executar a aplicação localmente
- Etapa 3: Implantar seu aplicativo ASP.NET com o AWS Toolkit for Visual Studio
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços de que você precisa.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Este QuickStart tutorial explica como criar um aplicativo “Hello World” e implantá-lo em um ambiente do Elastic Beanstalk com o Visual Studio e o. AWS Toolkit for Visual Studio

Visual Studio

Para fazer download e instalar o Visual Studio, siga as instruções na página de download do Visual Studio. Este exemplo usa o Visual Studio 2022. Durante a instalação do Visual Studio, selecione esses itens específicos:

- Na guia Cargas de trabalho, selecione ASP.NET e desenvolvimento da Web.
- Na guia Componentes individuais, selecione Ferramentas de desenvolvimento do .NET Framework 4.8 e Modelos de projeto e item do .NET Framework.

AWS Toolkit for Visual Studio

Para baixar e configurar, AWS Toolkit for Visual Studio siga as instruções no tópico Introdução do Guia do AWS Toolkit for Visual Studio usuário.

Etapa 1: criar uma aplicação ASP.NET

Em seguida, crie uma aplicação que você implantará em um ambiente Elastic Beanstalk. Criaremos uma aplicação Web ASP.NET "Hello World".

Para criar uma aplicação ASP.NET

1. Inicie o Visual Studio. No menu Arquivo, selecione Novo e, em seguida, Projeto.
2. A caixa de diálogo Criar um novo projeto é exibida. Selecione Aplicação Web ASP.NET (.NET Framework) e selecione Avançar.
3. Na caixa de diálogo Configurar seu novo projeto, digite `eb-aspnet`como nome do projeto. No menu suspenso Framework, selecione .NET Framework 4.8 e depois selecione Criar.

   Observe o diretório do projeto. Neste exemplo, o diretório do projeto é `C:\Users\Administrator\source\repos\eb-aspnet\eb-aspnet`.

4. A caixa de diálogo Criar um nova aplicação Web ASP.NET é exibida. Selecione o modelo Vazio. Em seguida, selecione Create (Criar).

   Neste ponto, você criou um projeto vazio de aplicação Web ASP.NET usando o Visual Studio. Em seguida, criaremos um formulário Web que servirá como ponto de entrada para a aplicação Web ASP.NET.
5. No menu Projeto, selecione Adicionar novo item. Na página Adocionar novo item, selecione Web Form e nomeie-o como `Default.aspx`. Em seguida, selecione Adicionar.
6. Adicione o seguinte 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>
```

Etapa 2: executar a aplicação localmente

No Visual Studio, no menu Depurar, selecione Iniciar depuração para executar sua aplicação localmente. O navegador deve exibir “Hello Elastic Beanstalk!. Esta é uma aplicação ASP.NET no Windows Server."

Etapa 3: Implantar seu aplicativo ASP.NET com o AWS Toolkit for Visual Studio

Siga estas etapas para criar um ambiente Elastic Beanstalk e implantar sua nova aplicação nele.

Para criar um ambiente e implantar sua aplicação ASP.NET

1. No Explorador de soluções, clique com o botão direito do mouse na aplicação e selecione Publicador no AWS Elastic Beanstalk.
2. Escolha um nome para a nova aplicação e o ambiente Elastic Beanstalk.
3. Além desse ponto, você pode continuar com os padrões fornecidos pelo Elastic Beanstalk ou modificar qualquer uma das opções e configurações a seu gosto.
4. Na página Review (Revisão), selecione Deploy (Implementar). Isso empacotará a aplicação Web ASP.NET e a implantará no Elastic Beanstalk.

   O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente. O recurso de implantação do Elastic Beanstalk irá monitorar seu ambiente até ele se tornar disponível com o código recém-implantado. Na guia Env:**<environment name>**, você verá o status do seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação do ambiente estiver concluído, o guia Env:**<environment name>** exibirá informações sobre o ambiente e a aplicação, incluindo o URL do domínio para iniciar a aplicação. Selecione este URL nesta guia ou copie e cole-o em seu navegador da Web.

Parabéns! Você implantou uma aplicação ASP.NET com o Elastic Beanstalk!

Etapa 5: limpar

Quando terminar de trabalhar com a aplicação, você pode encerrar seu ambiente no AWS Toolkit for Visual Studio.

Para encerrar seu ambiente

1. Expanda o nó Elastic Beanstalk e o nó da aplicação no AWS Explorer. Clique com o botão direito do mouse no ambiente de seu aplicativo e selecione Encerrar ambiente.
2. Quando solicitado, selecione Yes (Sim) para confirmar que você deseja encerrar o ambiente. O Elastic Beanstalk levará alguns minutos para AWS encerrar os recursos em execução no ambiente.

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do

aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

> **Explore mais tutoriais**
>
> Se quiser experimentar outros tutoriais com diferentes aplicações de exemplo, consulte QuickStart para o.NET Core no Windows.

Depois que você implantar um ou dois aplicações de amostra e estiver pronto para começar a desenvolver e executar aplicações ASP.NET localmente, consulte Configurar seu ambiente de desenvolvimento .NET

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Configurar seu ambiente de desenvolvimento .NET

Este tópico fornece instruções para configurar um ambiente de desenvolvimento .NET no Windows para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Seções

- Instalar um IDE
- Instalando o AWS Toolkit for Visual Studio

Se você precisar gerenciar AWS recursos de dentro do seu aplicativo, instale o AWS SDK para.NET. Por exemplo, é possível usar o Amazon S3 para armazenar e recuperar dados.

Com o AWS SDK para.NET, você pode começar em minutos com um único pacote disponível para download, completo com modelos de projeto do Visual Studio, AWS a biblioteca.NET, exemplos de código C# e documentação. São fornecidos exemplos práticos em C# de como usar a biblioteca para criar aplicativos. São fornecidos tutoriais de vídeo online e documentação de referência para ajudar você a aprender a usar as bibliotecas e os exemplos de códigos.

Visite a página inicial do AWS SDK for .NET para obter mais informações e instruções de instalação.

Instalar um IDE

Ambientes de desenvolvimento integrados (IDEs) fornecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você nunca usou um IDE para desenvolvimento .NET, experimente o Visual Studio Community para começar.

Visite a página do Visual Studio Community para fazer download e instalar o Visual Studio Community.

Instalando o AWS Toolkit for Visual Studio

O AWS Toolkit for Visual Studioé um plug-in de código aberto para o Visual Studio IDE que torna mais fácil para os desenvolvedores desenvolver, depurar e implantar aplicativos.NET usando o. AWS Acesse a página inicial do Toolkit for Visual Studio para obter instruções de instalação.

Usar o Elastic Beanstalk .NET na plataforma Windows

Este tópico descreve como configurar, criar e executar suas aplicações Web ASP.NET e .NET Core Windows no Elastic Beanstalk.

AWS Elastic Beanstalk oferece suporte a várias plataformas para diferentes versões da estrutura de programação.NET e do Windows Server. Consulte .NET no Windows Server com IIS no documento Plataformas do AWS Elastic Beanstalk para obter uma lista completa.

O Elastic Beanstalk fornece opções de configuração que você pode usar para personalizar o software executado EC2 nas instâncias do seu ambiente do Elastic Beanstalk. Você também pode configurar as variáveis do ambiente exigidas pela aplicação, permitir a alternância de logs para o Amazon S3 e definir configurações de framework .NET.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Configurar o ambiente .NET no console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3, configurar variáveis que a aplicação pode ler no ambiente e alterar as configurações de framework .NET.

Como configurar seu ambiente .NET no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de contêiner

- Target .NET runtime (runtime de .NET de destino): defina como `2.0` para executar CLR v2.
- Enable 32-bit applications (Habilitar aplicações 32 bits): defina como `True` para executar aplicações de 32 bits.

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Propriedades de ambiente

A seção Propriedades do ambiente permite que você especifique as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. Essas definições são passadas para o aplicativo como pares de chave-valor. Use `System.GetEnvironmentVariable` para leitura. Chaves idênticas podem existir em `web.config` e como propriedades do ambiente. Use o namespace `System.Configuration` para ler valores de `web.config`.

```
NameValueCollection appConfig = ConfigurationManager.AppSettings;
string endpoint = appConfig["API_ENDPOINT"];
```

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

O namespace aws:elasticbeanstalk:container:dotnet:apppool

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A plataforma .NET define opções no namespace `aws:elasticbeanstalk:container:dotnet:apppool` que você pode usar para configurar o runtime do .NET.

O arquivo de configuração de exemplo a seguir mostra as configurações de cada uma das opções disponíveis neste namespace:

Example .ebextensions/dotnet-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:dotnet:apppool:
    Target Runtime: 2.0
    Enable 32-bit Applications: True
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Migrar entre versões principais da plataforma do Windows Server do Elastic Beanstalk

AWS Elastic Beanstalk teve várias versões principais de sua plataforma Windows Server. Esta página aborda as principais melhorias para cada versão principal e o que deve ser considerado antes de migrar para uma versão mais recente.

A plataforma do Windows Server está na versão 2 (v2) no momento. Se o aplicativo usar qualquer versão de plataforma do Windows Server anterior à v2, recomendamos migrar para a v2.

Novidades das versões principais da plataforma do Windows Server

Plataforma do Windows Server V2

A versão 2 (v2) da plataforma Windows Server do Elastic Beanstalk foi lançada em fevereiro de 2019. A V2 aproxima o comportamento da plataforma Windows Server das plataformas Linux do Elastic Beanstalk de várias maneiras importantes. A V2 é totalmente compatível com a v1, o que facilita a migração da v1.

Agora, a plataforma do Windows Server oferece suporte a:

- Versionamento: cada versão obtém um novo número de versão, e você pode fazer referência às versões anteriores (que ainda estão disponíveis para você) ao criar e gerenciar ambientes.
- Integridade aprimorada: para obter detalhes, consulte Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk.
- Implantações imutáveis e contínuas com um lote adicional: para obter detalhes sobre as políticas de implantação, consulte Implantar aplicações em ambientes do Elastic Beanstalk.
- Atualizações imutáveis: para obter detalhes sobre os tipos de atualização, consulte Alterações de configuração.
- Atualizações gerenciadas de plataforma: para obter detalhes, consulte Atualizações gerenciadas de plataforma.

> **Note**
>
> Os novos recursos de implantação e atualização dependem da integridade aprimorada. Habilite a integridade aprimorada para usá-los. Para obter mais detalhes, consulte Habilitar relatórios de integridade aprimorada do Elastic Beanstalk.

Plataforma do Windows Server V1

A versão 1.0.0 (v1) da plataforma Windows Server do Elastic Beanstalk foi lançada em outubro de 2015. Essa versão altera a ordem na qual o Elastic Beanstalk processa comandos em arquivos de configuração durante a criação e as atualizações do ambiente.

As versões de plataformas anteriores não têm um número de versão no nome da pilha de soluções:

- Windows Server 2012 R2 de 64 bits executando IIS 8.5
- Windows Server Core 2012 R2 de 64 bits executando IIS 8.5
- Windows Server 2012 de 64 bits executando IIS 8
- Windows Server 2008 R2 de 64 bits executando IIS 7.5

Em versões anteriores, a ordem de processamento dos arquivos de configuração é inconsistente. Durante a criação do ambiente, `Container Commands` executam depois que o código-fonte do aplicativo é implantado no IIS. Durante uma implantação em um ambiente em execução, os comandos de contêiner executam antes da implantação da nova versão. Durante uma expansão, os arquivos de configuração não são processados.

Além disso, o IIS inicia antes da execução dos comandos de contêiner. Esse comportamento levou alguns clientes a implementar soluções em comandos de contêiner, pausando o servidor IIS antes da execução dos comandos e reiniciando-o após a conclusão.

A versão 1 corrige a inconsistência e aproxima o comportamento da plataforma Windows Server ao das plataformas baseadas em Linux do Elastic Beanstalk. Na plataforma v1, o Elastic Beanstalk sempre executa comandos de contêiner antes de iniciar o servidor IIS.

As pilhas de soluções da plataforma v1 têm um `v1` após a versão do Windows Server:

- Windows Server 2012 R2 v1.1.0 de 64 bits executando IIS 8.5
- Windows Server Core 2012 R2 v1.1.0 de 64 bits executando IIS 8.5
- Windows Server 2012 v1.1.0 de 64 bits executando IIS 8
- Windows Server 2008 R2 v1.1.0 de 64 bits executando IIS 7.5

Além disso, a plataforma v1 extrai o conteúdo do pacote de origem do aplicativo em `C:\staging\` antes da execução de comandos de contêiner. Após a conclusão dos comandos de contêiner, o

conteúdo dessa pasta é compactado em um arquivo .zip e implantado no IIS. Esse fluxo de trabalho permite que você modifique o conteúdo do pacote de origem do aplicativo com comandos ou com um script antes da implantação.

Migrar de versões principais anteriores da plataforma do Windows Server

Leia esta seção sobre considerações de migração antes de atualizar seu ambiente. Para atualizar a plataforma de seu ambiente para uma versão mais recente, visite Atualizar a versão de plataforma do ambiente Elastic Beanstalk.

De V1 para V2

A plataforma v2 do Windows Server não oferece suporte a .NET Core 1.x e 2.0. Se você está migrando o aplicativo do Windows Server v1 para v2 e o aplicativo usa uma dessas versões do .NET Core, atualize o aplicativo para uma versão do .NET Core que ofereça suporte a v2. Para obter uma lista de versões compatíveis, consulte .NET no Windows Server com IIS nas Plataformas do AWS Elastic Beanstalk .

Se o aplicativo usar uma imagem de máquina da Amazon (AMI), crie uma nova AMI personalizada com base em uma AMI da plataforma Windows Server v2. Para saber mais, consulte Usar uma imagem de máquina Amazon personalizada (AMI) no ambiente Elastic Beanstalk.

> **Note**
>
> Os recursos de implantação e atualização que são novos no Windows Server v2 dependem da integridade aprimorada. Ao migrar um ambiente para v2, a integridade aprimorada é desabilitada. Habilite-a para usar esses recursos. Para obter mais detalhes, consulte Habilitar relatórios de integridade aprimorada do Elastic Beanstalk.

Da pré-V1

Além das considerações sobre a migração de v1, se você estiver migrando o aplicativo de uma pilha de soluções do Windows Server anterior à v1 e estiver usando comandos de contêiner, remova todos os comandos adicionados para resolver as inconsistências de processamento ao migrar para uma versão mais recente. A partir da v1, é garantido que os comandos de contêiner sejam executados completamente antes da origem do aplicativo implantado e antes do início do IIS. Isso permite que você faça qualquer alteração na origem em `C:\staging` e modifique os arquivos de configuração do IIS durante esta etapa sem problemas.

Por exemplo, você pode usar o AWS CLI para baixar um arquivo DLL para a fonte do seu aplicativo a partir do Amazon S3:

`.ebextensions\copy-dll.config`

```
container_commands:
  copy-dll:
    command: aws s3 cp s3://amzn-s3-demo-bucket/dlls/large-dll.dll .\lib\
```

Para obter mais informações sobre o uso de arquivos de configuração, consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`).

Executar vários aplicativos e aplicativos ASP.NET Core com um manifesto de implantação

É possível usar um manifesto de implantação para informar o Elastic Beanstalk como implantar a aplicação. Ao usar esse método, você não precisa usar `MSDeploy` para gerar um pacote de origem para uma única aplicação ASP.NET que é executada no caminho raiz do seu site. Em vez disso, você pode usar um arquivo de manifesto para executar várias aplicações em caminhos diferentes. Ou, como alternativa, você pode dizer ao Elastic Beanstalk para implantar e executar a aplicação com o ASP.NET Core. Também é possível usar um manifesto de implantação para configurar um grupo de aplicações para executar suas aplicações.

Os manifestos de implantação adicionam suporte para aplicações .NET Core ao Elastic Beanstalk. Você pode implantar uma aplicação .NET Framework sem um manifesto de implantação. No entanto, as aplicações .NET Core exigem um manifesto de implantação para execução no Elastic Beanstalk. Ao usar um manifesto de implantação, você cria um arquivo do site para cada aplicação e, em seguida, empacota os arquivos do site em um segundo arquivo ZIP que contém o manifesto de implantação.

Os manifestos de implantação também adicionam a capacidade de executar vários aplicativos em diferentes caminhos. Um manifesto de implantação define um conjunto de alvos de implantação, cada um com um arquivamento do site e um caminho em que o IIS deve executá-lo. Por exemplo, você pode executar uma API da Web no caminho `/api` para atender a solicitações assíncronas e um aplicativo web no caminho raiz que consome a API.

Você também pode usar um manifesto de implantação para executar várias aplicações usando grupos de aplicações no IIS ou no Kestrel. Você pode configurar um grupo de aplicativos para

reiniciar seus aplicativos periodicamente, executar aplicativos de 32 bits ou usar uma versão específica do runtime do .NET Framework.

Para uma personalização completa, você pode escrever seus próprios scripts de implantação no Windows PowerShell e informar ao Elastic Beanstalk quais scripts executar para instalar, desinstalar e reiniciar seu aplicativo.

Os manifestos de implantação e os recursos relacionados exigem uma plataforma do Windows Server versão 1.2.0 ou posterior.

Para obter informações detalhadas sobre todas as opções de configuração, propriedades e recursos avançados disponíveis, como ignorar redefinições do IIS, consulte a referência do esquema do manifesto de implantação.

Seções

- Aplicativos .NET Core
- Executar vários aplicativos
- Configurar grupos de aplicativos
- Definir implantações personalizadas
- Referência do esquema do manifesto de implantação

Aplicativos .NET Core

É possível usar um manifesto de implantação para executar aplicações .NET Core no Elastic Beanstalk. .NET Core é uma versão multiplataforma do .NET que é fornecida com uma ferramenta de linha de comando (`dotnet`). Você pode usá-lo para gerar uma aplicação, executá-lo localmente e prepará-lo para publicação.

Para executar uma aplicação .NET Core no Elastic Beanstalk, execute `dotnet publish` e empacote a saída em um arquivo ZIP, sem incluir os diretórios. Coloque o arquivo do site em um pacote de origem com um manifesto de implantação com um destino de implantação do tipo `aspNetCoreWeb`.

O seguinte manifesto de implantação executa um aplicativo .NET Core em um arquivo de site chamado `dotnet-core-app.zip` no caminho raiz.

Example aws-windows-deployment-manifest.json - Núcleo.NET

```
{
```

```
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Inclua o arquivo de manifesto e de site em um arquivo ZIP para criar um pacote de origem.

Example dotnet-core-bundle.zip

```
.
|-- aws-windows-deployment-manifest.json
`-- dotnet-core-app.zip
```

O arquivo do site contém o código do aplicativo compilado, as dependências e o arquivo `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
```

Executar vários aplicativos

Você pode executar vários aplicativos com um manifesto de implantação definindo vários alvos de implantação.

O seguinte manifesto de implantação configura dois aplicativos .NET Core. A aplicação `WebApiSampleApp` implementa uma API simples da Web e atende a solicitações assíncronas no caminho `/api`. O aplicativo `DotNetSampleApp` é um aplicativo web que atende solicitações no caminho raiz.

Example aws-windows-deployment-manifest.json - vários aplicativos

```
{
  "manifestVersion": 1,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "WebAPISample",
        "parameters": {
          "appBundle": "WebApiSampleApp.zip",
          "iisPath": "/api"
        }
      },
```

```
      {
        "name": "DotNetSample",
        "parameters": {
          "appBundle": "DotNetSampleApp.zip",
          "iisPath": "/"
        }
      }
    ]
  }
}
```

Um aplicativo de exemplo com vários aplicativos está disponível em:

- Pacote de origem implantável - -v2.zip dotnet-multiapp-sample-bundle
- Código fonte - dotnet-multiapp-sample-source-v2.zip

Configurar grupos de aplicativos

Você pode oferecer suporte a várias aplicações no seu ambiente Windows. Duas abordagens estão disponíveis:

- Você pode usar o modelo de out-of-process hospedagem com o servidor web Kestrel. Com esse modelo, você configura várias aplicações para serem executadas em um grupo de aplicações.
- Você pode usar o modelo de hospedagem no processo. Com esse modelo, você usa vários grupos de aplicações para executar várias aplicações com apenas uma aplicação em cada grupo. Se você estiver usando o servidor IIS e precisar executar várias aplicações, deverá usar essa abordagem.

Para configurar o Kestrel para executar várias aplicações em um grupo de aplicações, adicione `hostingModel="OutofProcess"` ao arquivo `web.config`. Considere os seguintes exemplos:

Example web.config - para o modelo de hospedagem 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 - vários aplicativos

```
{
"manifestVersion": 1,
  "deployments": {"msDeploy": [
      {"name": "Web-app1",
        "parameters": {"archive": "site1.zip",
          "iisPath": "/"
        }
      },
      {"name": "Web-app2",
        "parameters": {"archive": "site2.zip",
          "iisPath": "/app2"
        }
      }
    ]
  }
}
```

O IIS não suporta várias aplicações em um grupo de aplicações porque ele usa o modelo de hospedagem no processo. Portanto, você precisa configurar várias aplicações atribuindo cada aplicação a um grupo de aplicações. Em outras palavras, atribua apenas uma aplicação a um grupo de aplicações.

Você pode configurar o IIS para usar grupos de aplicações diferentes no arquivo `aws-windows-deployment-manifest.json`. Faça as seguintes atualizações conforme você se refere ao próximo arquivo de exemplo:

- Adicione uma `iisConfig` seção que inclua uma subseção chamada `appPools`.

- No bloco `appPools`, liste os grupos de aplicações.
- Na seção `deployments`, defina uma seção `parameters` para cada aplicação.
- Para cada aplicação, a seção `parameters` especifica um arquivo, um caminho para executá-la e um `appPool` no qual executá-la.

O manifesto de implantação a seguir configura dois grupos de aplicações que reiniciam sua aplicação a cada 10 minutos. Eles também anexam suas aplicações a uma aplicação web .NET Framework que é executada no caminho especificado.

Example aws-windows-deployment-manifest.json - um aplicativo por pool de aplicativos

```
{
"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"
         }
      }
     ]
    }
}
```

Definir implantações personalizadas

Para controle ainda maior, você pode personalizar totalmente a implantação de um aplicativo definindo uma implantação personalizada.

Esse manifesto de implantação instrui o Elastic Beanstalk PowerShell a executar scripts no modo de 32 bits. Ele especifica três scripts: um `install` script (`siteInstall.ps1`) que é executado durante a inicialização e as implantações da instância, um `uninstall` script (`siteUninstall.ps1`) que é executado antes de instalar novas versões durante as implantações e um `restart` script (`siteRestart.ps1`) que é executado quando você seleciona Reiniciar o App Server no console de gerenciamento. AWS

Example aws-windows-deployment-manifest.json - implantação personalizada

```
{
  "manifestVersion": 1,
  "deployments": {
    "custom": [
      {
        "name": "Custom site",
        "architecture" : 32,
        "scripts": {
          "install": {
            "file": "siteInstall.ps1"
          },
          "restart": {
            "file": "siteRestart.ps1"
          },
          "uninstall": {
            "file": "siteUninstall.ps1"
          }
        }
      }
    ]
  }
}
```

Inclua todos os artefatos necessários para executar o aplicativo em seu pacote de origem com o manifesto e os scripts.

Example ustom-site-bundleC.zip

```
.
|-- aws-windows-deployment-manifest.json
|-- siteInstall.ps1
|-- siteRestart.ps1
|-- siteUninstall.ps1
`-- site-contents.zip
```

Referência do esquema do manifesto de implantação

O manifesto de implantação é um arquivo JSON que define como o Elastic Beanstalk deve implantar e configurar seus aplicativos Windows. Esta seção fornece uma referência abrangente para todas as propriedades e opções de configuração suportadas no esquema do manifesto.

Estrutura do manifesto

O manifesto de implantação segue um esquema JSON específico com a seguinte estrutura de nível superior:

Example Estrutura básica do manifesto

```
{
  "manifestVersion": 1,
  "skipIISReset": false,
  "iisConfig": {
    "appPools": [...]
  },
  "deployments": {
    "msDeploy": [...],
    "aspNetCoreWeb": [...],
    "custom": [...]
  }
}
```

Propriedades de nível superior

`manifestVersion`(obrigatório)

Tipo: número

Padrão: 1

Valores válidos: 1

Especifica a versão do esquema do manifesto. Atualmente, somente a versão 1 é suportada.

`skipIISReset` (opcional)

Tipo: booliano

Padrão: falso

Controla se o IIS é redefinido durante as implantações do aplicativo. Esse sinalizador afeta tanto os tipos de `aspNetCoreWeb` implantação `msDeploy` quanto os tipos de implantação.

Comportamento:

- Não especificado ou *`false`* (padrão): as redefinições do IIS são realizadas durante as operações de instalação, desinstalação e atualização. Esse é o comportamento tradicional.
- *`true`*: as redefinições do IIS são ignoradas durante as operações de implantação.

Benefícios:

- Tempo de inatividade reduzido — os aplicativos sofrem menos interrupções de serviço durante as implantações.
- Implantações mais rápidas — elimina o tempo necessário para que o IIS seja totalmente reiniciado e reinicializado.

> **Note**
>
> Ao usar`skipIISReset`, a RestartAppServeroperação executa uma redefinição do IIS independentemente dessa configuração de sinalizador.

Exemplo:

```
{
  "manifestVersion": 1,
  "skipIISReset": true,
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "my-dotnet-core-app",
        "parameters": {
          "archive": "dotnet-core-app.zip",
```

```
          "iisPath": "/"
        }
      }
    ]
  }
}
```

`deployments`(obrigatório)

Tipo: Objeto

Contém as configurações de implantação para seus aplicativos. Esse objeto pode incluir`msDeploy`,`aspNetCoreWeb`, e tipos de `custom` implantação.

`iisConfig` (opcional)

Tipo: Objeto

Define as configurações do IIS a serem aplicadas antes da implantação dos aplicativos. Atualmente, oferece suporte à configuração do pool de aplicativos.

Configuração do IIS

A `iisConfig` seção permite que você defina as configurações do IIS antes de implantar seus aplicativos. Isso é particularmente útil para configurar grupos de aplicativos com configurações específicas.

Pools de aplicativos

Os pools de aplicativos fornecem isolamento entre aplicativos e permitem que você defina as configurações de tempo de execução para grupos de aplicativos.

Example Configuração do pool de aplicativos

```
{
  "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
        }
      }
    ]
  }
}
```

Propriedades do pool de aplicativos

`name`(obrigatório)

Tipo: string

O nome do pool de aplicativos. Esse nome é usado para referenciar o pool nas configurações de implantação.

`enable32Bit` (opcional)

Tipo: booliano

Permite que um aplicativo de 32 bits seja executado em uma versão de 64 bits do Windows. Defina como `true` para aplicativos legados que exigem compatibilidade de 32 bits.

`managedPipelineMode` (opcional)

Tipo: string

Valores válidos: “Integrado”, “Clássico”

Especifica o modo de processamento de solicitações para o pool de aplicativos.

`managedRuntimeVersion` (opcional)

Tipo: string

Valores válidos: “Sem código gerenciado”, “v2.0", “v4.0"

Especifica a versão do.NET Framework para o pool de aplicativos.

`queueLength` (opcional)

Tipo: inteiro

Número máximo de solicitações que o HTTP.sys enfileira para o pool de aplicativos antes de rejeitar solicitações adicionais.

Configuração da CPU

O `cpu` objeto configura os limites de uso da CPU e o monitoramento do pool de aplicativos.

`limitPercentage` (opcional)

Tipo: número

Porcentagem máxima de tempo de CPU que os processos de trabalho no pool de aplicativos podem consumir.

`limitAction` (opcional)

Tipo: string

Valores válidos: "NoAction“, “KillW3wp”, “Throttle”, "” ThrottleUnderLoad

Ação a ser tomada quando o limite de CPU for atingido.

`limitMonitoringInterval` (opcional)

Tipo: número

Período de redefinição (em minutos) para monitoramento da CPU e limites de aceleração.

Configuração de reciclagem

O `recycling` objeto configura quando e como os processos de trabalho do pool de aplicativos são reciclados.

`regularTimeInterval` (opcional)

Tipo: inteiro

Intervalo de tempo (em minutos) após o qual o pool de aplicativos é reciclado. Defina como 0 para desativar a reciclagem com base no tempo.

`requestLimit` (opcional)

Tipo: inteiro

Número máximo de solicitações que o pool de aplicativos processa antes da reciclagem.

`memory` (opcional)

Tipo: inteiro

Quantidade de memória virtual (em kilobytes) que aciona a reciclagem do processo de trabalho.

`privateMemory` (opcional)

Tipo: inteiro

Quantidade de memória privada (em kilobytes) que aciona a reciclagem do processo de trabalho.

Tipos de implantação

O `deployments` objeto contém matrizes de configurações de implantação para diferentes tipos de aplicativos. Cada tipo de implantação tem propriedades e casos de uso específicos.

MSDeploy implantações

MSDeploy as implantações são usadas para aplicativos do.NET Framework tradicionais que podem ser implantados usando o Web Deploy ()MSDeploy.

Example MSDeploy configuração de implantação

```
{
  "deployments": {
    "msDeploy": [
      {
        "name": "WebApp",
        "description": "Main web application",
        "parameters": {
          "appBundle": "webapp.zip",
          "iisPath": "/",
          "appPool": "DefaultAppPool"
        }
      }
    ]
  }
```

```
}
```

MSDeploy propriedades de implantação

`name`(obrigatório)

Tipo: string

Nome exclusivo para a implantação. Esse nome deve ser exclusivo em todas as implantações no manifesto.

`description` (opcional)

Tipo: string

Descrição legível por humanos da implantação.

`parameters`(obrigatório)

Tipo: Objeto

Parâmetros de configuração para a MSDeploy operação.

`scripts` (opcional)

Tipo: Objeto

PowerShell scripts para execução em vários estágios do ciclo de vida da implantação.

MSDeploy parâmetros

`appBundle`(obrigatório)

Tipo: string

Caminho para o pacote de aplicativos (arquivo ZIP) em relação ao arquivo de manifesto. Esse pacote contém os arquivos do aplicativo a serem implantados.

`iisPath` (opcional)

Tipo: string

Padrão: “/”

Caminho do diretório virtual no IIS em que o aplicativo será implantado. Use “/” para o caminho raiz ou “/api” para um subdiretório.

`appPool` (opcional)

Tipo: string

Nome do pool de aplicativos para executar esse aplicativo.

Implantações do ASP.NET Core

As implantações do ASP.NET Core são projetadas especificamente para aplicativos.NET Core e o.NET 5+.

Example Configuração de implantação do ASP.NET Core

```
{
  "deployments": {
    "aspNetCoreWeb": [
      {
        "name": "CoreAPI",
        "description": "ASP.NET Core Web API",
        "parameters": {
          "appBundle": "coreapi.zip",
          "iisPath": "/api",
          "appPool": "CoreAppPool"
        }
      }
    ]
  }
}
```

As implantações do ASP.NET Core usam a mesma estrutura de propriedades das MSDeploy implantações, com a principal diferença sendo o ambiente de tempo de execução e o modelo de hospedagem usados para o aplicativo.

Parâmetros de implantação do ASP.NET Core

`appBundle`(obrigatório)

Tipo: string

Caminho para o pacote de aplicativos em relação ao arquivo de manifesto. Isso pode ser um arquivo ZIP ou um caminho de diretório contendo o aplicativo ASP.NET Core publicado.

`iisPath` (opcional)

Tipo: string

Padrão: “/”

Caminho do diretório virtual no IIS para o aplicativo ASP.NET Core.

`appPool` (opcional)

Tipo: string

Pool de aplicativos para o aplicativo ASP.NET Core. O pool será configurado adequadamente para hospedagem ASP.NET Core.

Implantações personalizadas

As implantações personalizadas fornecem controle total sobre o processo de implantação por meio de PowerShell scripts. Esse tipo de implantação é útil para cenários complexos que exigem instalação, configuração ou lógica de implantação personalizadas.

Example Configuração de implantação personalizada

```
{
  "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
          }
        }
      }
```

```
    ]
  }
}
```

Propriedades de implantação personalizadas

`name`(obrigatório)

Tipo: string

Nome exclusivo para a implantação personalizada.

`description` (opcional)

Tipo: string

Descrição da implantação personalizada.

`architecture` (opcional)

Tipo: inteiro

Padrão: 32

Valores válidos: 32, 64

A especificação de arquitetura para o modo de execução de scripts PowerShell

`scripts`(obrigatório)

Tipo: Objeto

PowerShell scripts que definem o comportamento da implantação. As implantações personalizadas oferecem suporte a tipos de script adicionais em comparação com outros tipos de implantação.

Scripts de implantação

Os scripts de implantação são PowerShell scripts executados em pontos específicos durante o ciclo de vida da implantação. Diferentes tipos de implantação oferecem suporte a diferentes conjuntos de eventos de script.

Eventos de script

Os seguintes eventos de script estão disponíveis dependendo do tipo de implantação:

Scripts de implantação padrão (MSDeploy e aspNetCore Web)

`preInstall`

É executado antes que o aplicativo seja instalado ou atualizado.

`postInstall`

É executado depois que o aplicativo é instalado ou atualizado.

`preRestart`

É executado antes que o aplicativo seja reiniciado.

`postRestart`

É executado depois que o aplicativo é reiniciado.

`preUninstall`

É executado antes da desinstalação do aplicativo.

`postUninstall`

É executado após a desinstalação do aplicativo.

Scripts de implantação personalizados (somente implantações personalizadas)

`install`

Script de instalação principal para implantações personalizadas. Esse script é responsável pela instalação do aplicativo ou serviço.

`restart`

Script para reiniciar o aplicativo ou serviço. Chamado quando o ambiente é reiniciado.

`uninstall`

Script para desinstalar o aplicativo ou serviço. Chamado durante o encerramento do ambiente ou a remoção do aplicativo.

Propriedades do script

Cada script é definido como um objeto com as seguintes propriedades:

`file`(obrigatório)

Tipo: string

Caminho para o arquivo de PowerShell script em relação ao arquivo de manifesto. O script deve ter uma `.ps1` extensão.

`ignoreErrors` (opcional)

Tipo: booliano

Padrão: falso

Quando definido como`true`, a implantação continua mesmo se o script falhar. Use isso para scripts não críticos ou operações de limpeza.

Example Exemplo de configuração de script

```
{
  "scripts": {
    "preInstall": {
      "file": "backup-config.ps1",
      "ignoreErrors": true
    },
    "postInstall": {
      "file": "configure-app.ps1"
    }
  }
}
```

Usando o EC2 Fast Launch com ramificações da plataforma Windows

O recurso EC2 Fast Launch reduz os tempos de execução de instâncias do Windows em seus ambientes do Elastic Beanstalk. O objetivo deste tópico é orientá-lo sobre como usar esse recurso com seus ambientes do Elastic Beanstalk. A partir da versão 2.16.2 da plataforma Windows, lançada em 22 de janeiro de 2025, os lançamentos da plataforma Elastic Beanstalk incluem base com o Fast Launch ativado. AMIs EC2

Disponibilidade padrão de lançamento EC2 rápido

As versões mais recentes da plataforma Windows do Elastic Beanstalk AMIs incluem EC2 base com o Fast Launch ativado automaticamente, sem custos adicionais. No entanto, quando versões

mais recentes da plataforma são lançadas, o EC2 Fast Launch pode não permanecer ativado automaticamente com base nas versões mais antigas AMIs da plataforma.

Recomendamos atualizar para a versão mais recente da plataforma Windows para usar o base AMIs com o EC2 Fast Launch ativado automaticamente. No entanto, se precisar continuar usando a versão atual da plataforma, você pode habilitar manualmente o EC2 Fast Launch na AMI básica do seu ambiente. Para obter instruções, consulte Configurando manualmente o EC2 Fast Launch.

Configurando manualmente o EC2 Fast Launch

> **Note**
>
> A ativação manual do EC2 Fast Launch pode gerar custos adicionais em comparação ao uso de versões da plataforma com o EC2 Fast Launch ativado automaticamente. Para obter mais informações sobre os custos do EC2 Fast Launch, consulte a página Gerenciar custos dos recursos subjacentes do EC2 Fast Launch no Guia EC2 do usuário da Amazon.

Siga estas etapas para habilitar o EC2 Fast Launch em uma AMI baseada no Windows usada pelo seu ambiente do Elastic Beanstalk:

Para habilitar manualmente o EC2 Fast Launch para seu ambiente Elastic Beanstalk

1. Identifique a AMI básica do seu ambiente:

   Siga as etapas em Criação de uma AMI personalizada para identificar a ID de AMI base do seu ambiente. Observe que você não precisa criar uma AMI personalizada. Você só precisa seguir as etapas para localizar sua ID de AMI base atual.
2. Ative o EC2 Fast Launch na AMI:

   Use as instruções em Habilitar o EC2 Fast Launch no Guia EC2 do usuário da Amazon para configurar o EC2 Fast Launch para sua AMI.

Adicionar uma instância de banco de dados do Amazon RDS ao ambiente de aplicações .NET

Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon

RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Nestas instruções, o banco de dados é acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Seções

- Adicionar uma instância de banco de dados ao seu ambiente
- Download de um driver
- Conectar-se a um banco de dados

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.

Nome da propriedade	Descrição	Valor da propriedade
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
`RDS_PASSWORD`	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Download de um driver

Faça download e instale o pacote `EntityFramework` e um driver de banco de dados para o seu ambiente de desenvolvimento com `NuGet`.

Provedores comuns de banco de dados de framework de entidade para .NET

- SQL Server – `Microsoft.EntityFrameworkCore.SqlServer`
- MySQL – `Pomelo.EntityFrameworkCore.MySql`
- PostgreSQL – `Npgsql.EntityFrameworkCore.PostgreSQL`

Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `ConfigurationManager.AppSettings` para ler as propriedades e configurar uma conexão de banco de dados.

Example Helpers.cs - método de string de conexão

```
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 + ";";
    }
  }
}
```

Use a string de conexão para inicializar o contexto do banco de dados.

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();
    }
  }
}
```

O AWS Toolkit for Visual Studio

O Visual Studio fornece modelos para diferentes linguagens de programação e tipos de aplicativos. Você pode começar com qualquer um desses modelos. O AWS Toolkit for Visual Studio também fornece três modelos de projeto que inicializam o desenvolvimento do seu aplicativo AWS : Console Project AWS , Web Project AWS e Empty Project. Para este exemplo, você vai criar um novo aplicativo web ASP.NET.

Como criar um novo projeto de aplicativo web ASP.NET

1. No Visual Studio, no menu Arquivo, clique em Novo e em Projeto.
2. Na caixa de diálogo Novo projeto, clique em Modelos instalados, clique em Visual C# e, em seguida, clique em Web. Clique em Aplicativo web ASP.NET vazio, digite um nome de projeto e, em seguida, clique em OK.

Para executar um projeto

Execute um destes procedimentos:

1. Pressione F5.

2. Selecione Iniciar depuração no menu Depurar.

Testar localmente

O Visual Studio facilita testar seu aplicativo localmente. Para testar ou executar aplicativos web em ASP.NET, você precisa de um servidor web. O Visual Studio oferece várias opções, como o Internet Information Services (IIS), IIS Express ou o Visual Studio Development Server. Para saber mais sobre cada uma dessas opções e decidir qual é a melhor para você, consulte Servidores Web no Visual Studio para projetos Web ASP.NET .

Criar um ambiente do Elastic Beanstalk

Depois de testar a aplicação, você estará pronto para implantá-la no Elastic Beanstalk.

> **Note**
>
> O Arquivo de configuração precisa fazer parte do projeto para ser incluído no arquivo. Se preferir, em vez de incluir os arquivos de configuração no projeto, você pode usar o Visual Studio para implantar todos os arquivos na pasta do projeto. Em Solution Explorer, clique com o botão direito do mouse no nome do projeto e, em seguida, clique em Propriedades. Clique na guia Package/Publish Web. Na seção Items to deploy, selecione All Files in the Project Folder na lista suspensa.

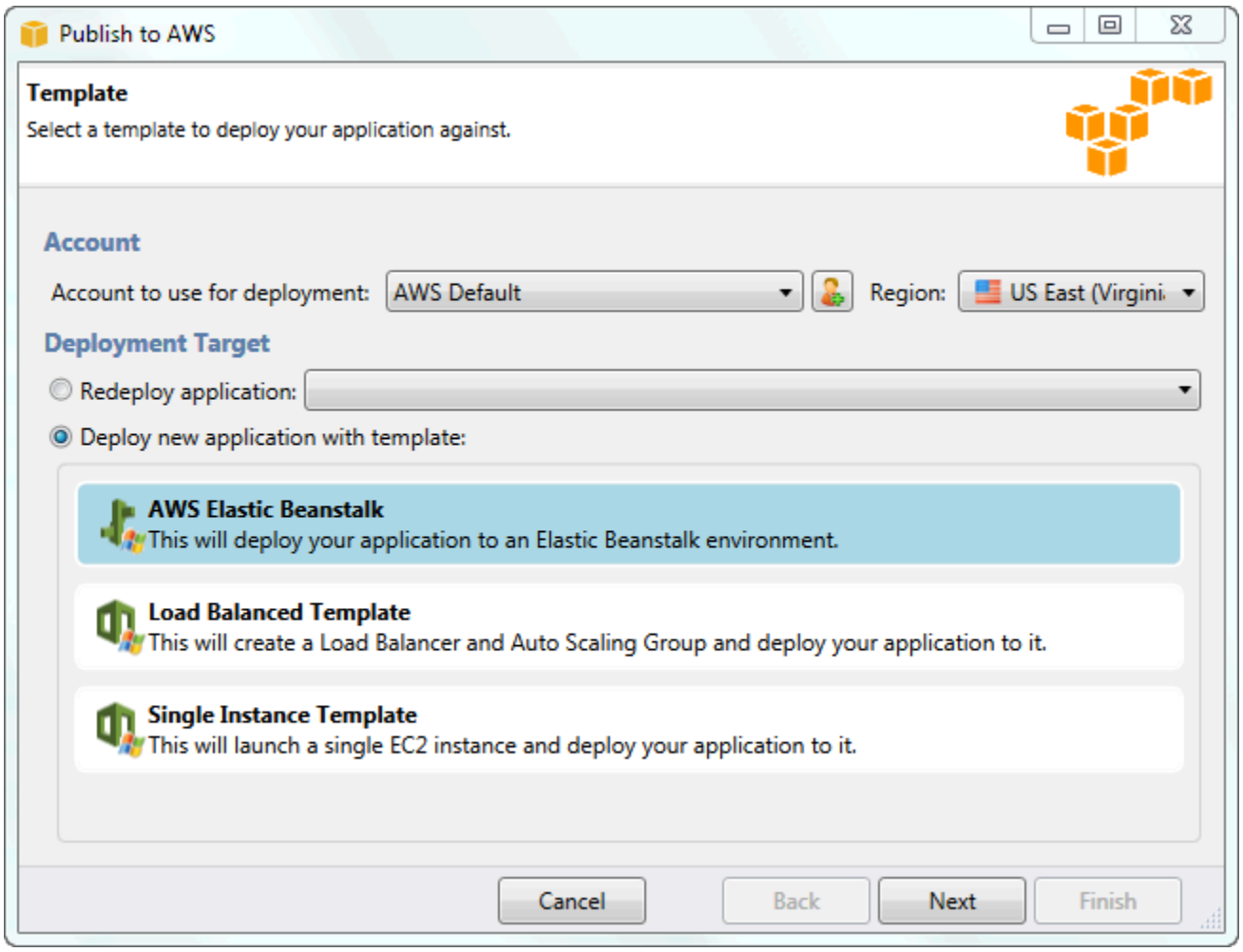
Para implantar seu aplicativo no Elastic Beanstalk AWS usando o kit de ferramentas do Visual Studio

1. No Solution Explorer, clique com o botão direito do mouse em seu aplicativo e selecione Publicar em AWS.
2. No assistente Publicar na AWS, insira as informações da sua conta.

    a. Para Conta da AWS a utilizar para implantação, selecione sua conta ou selecione Outra para inserir as informações da nova conta.

    b. Para Região, selecione a região em que você deseja implantar o aplicativo. Para obter informações sobre AWS regiões disponíveis, consulte AWS Elastic Beanstalk Endpoints e cotas no. Referência geral da AWS Se você selecionar uma região não compatível com o Elastic Beanstalk, a opção para implantar no Elastic Beanstalk se tornará indisponível.

    c. Clique em Implantar novo aplicativo com modelo e selecione Elastic Beanstalk. Em seguida, clique em Próximo.



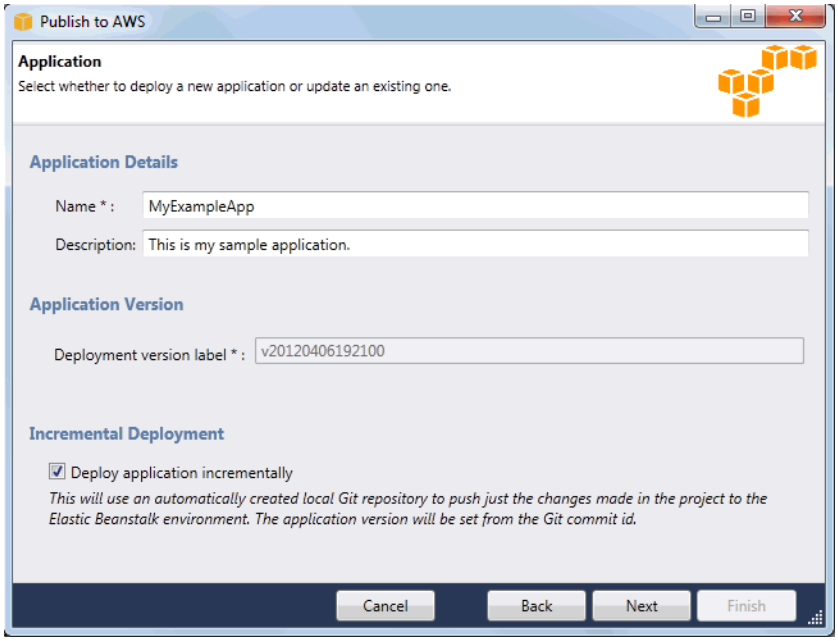
3. Na página Aplicativo, insira os detalhes de seu aplicativo.

   a. Em Nome, digite o nome do aplicativo.

   b. Em Descrição, digite uma descrição do aplicativo. Esta etapa é opcional.

   c. O rótulo da versão do aplicativo é exibido automaticamente em Deployment version label (Rótulo da versão de implantação)

   d. Selecione Implantar aplicativo incrementalmente para implantar somente os arquivos alterados. Uma implantação incremental é mais rápida porque você está atualizando somente os arquivos alterados em vez de todos os arquivos. Se você escolher essa opção, uma versão do aplicativo será definida a partir do ID de confirmação do Git. Se você optar por não implantar seu aplicativo de forma incremental, poderá atualizar o rótulo da versão na caixa Rótulo da versão de implantação.



e. Clique em Next.

4. Na página Ambiente, descreva os detalhes de seu ambiente.

   a. Selecione Criar um novo ambiente para este aplicativo.

   b. Em Nome, digite um nome para o ambiente.

   c. Em Descrição, caracterize seu ambiente. Esta etapa é opcional.

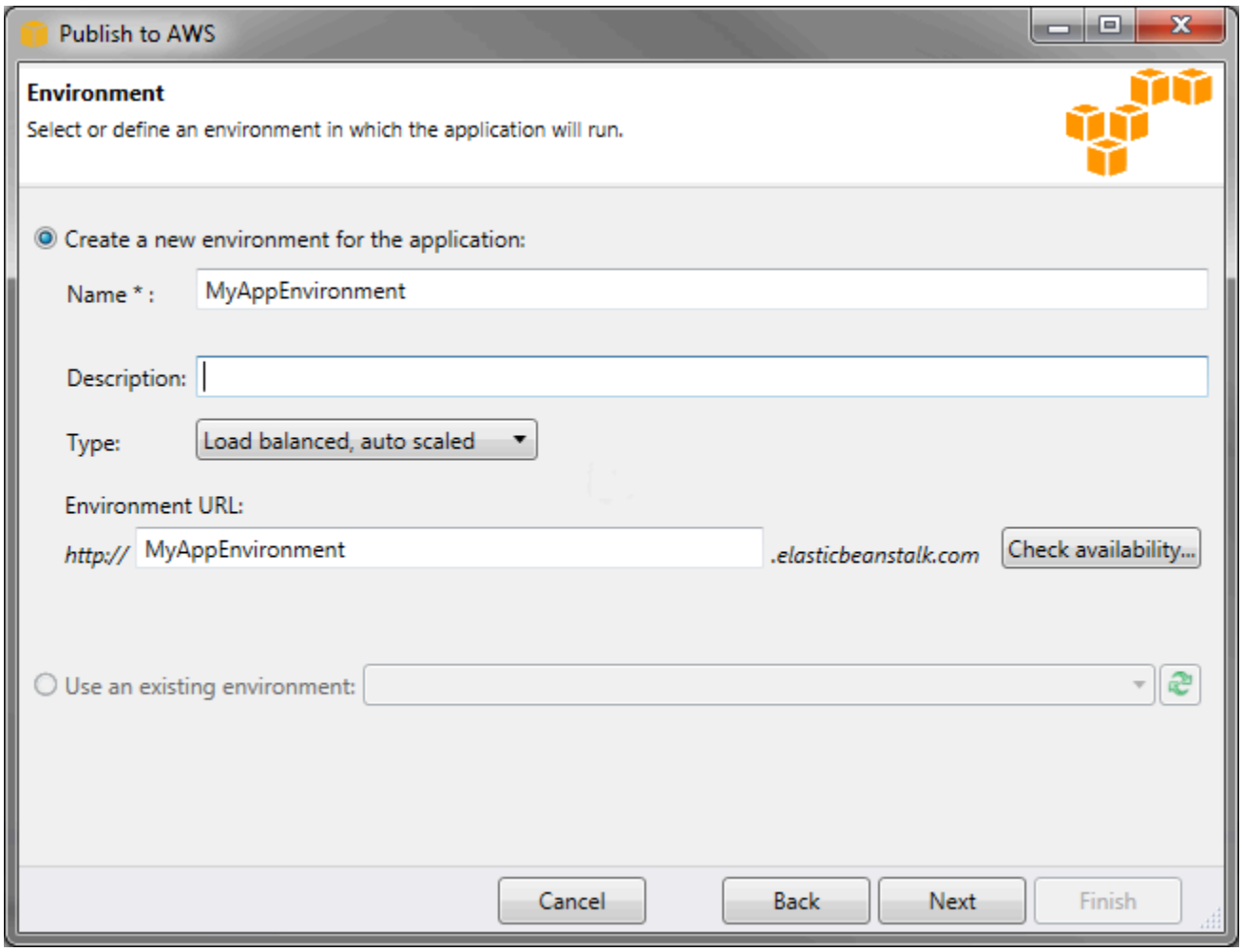
   d. Em Tipo, selecione o tipo de ambiente que deseja.

      Você pode selecionar um ambiente Carga balanceada, escalabilidade automática ou Única instância. Para obter mais informações, consulte Tipos de ambiente.

      > **Note**
      >
      > Para ambientes de uma única instância, as configurações de balanceamento de carga, autoescalabilidade e URL de verificação de integridade não se aplicam

   e. O URL do ambiente é exibido automaticamente em URL do ambiente assim que você move o cursor para essa caixa.

   f. Clique em Verificar disponibilidade para garantir que o URL do ambiente esteja disponível.



g. Clique em Next.

5. Na página Opções da AWS , configure as opções adicionais e as informações de segurança de sua implantação.

   a. Em Tipo de contêiner, selecione Windows Server 2012 de 64 bits executando IIS 8 ou Windows Server 2008 de 64 bits executando IIS 7.5.

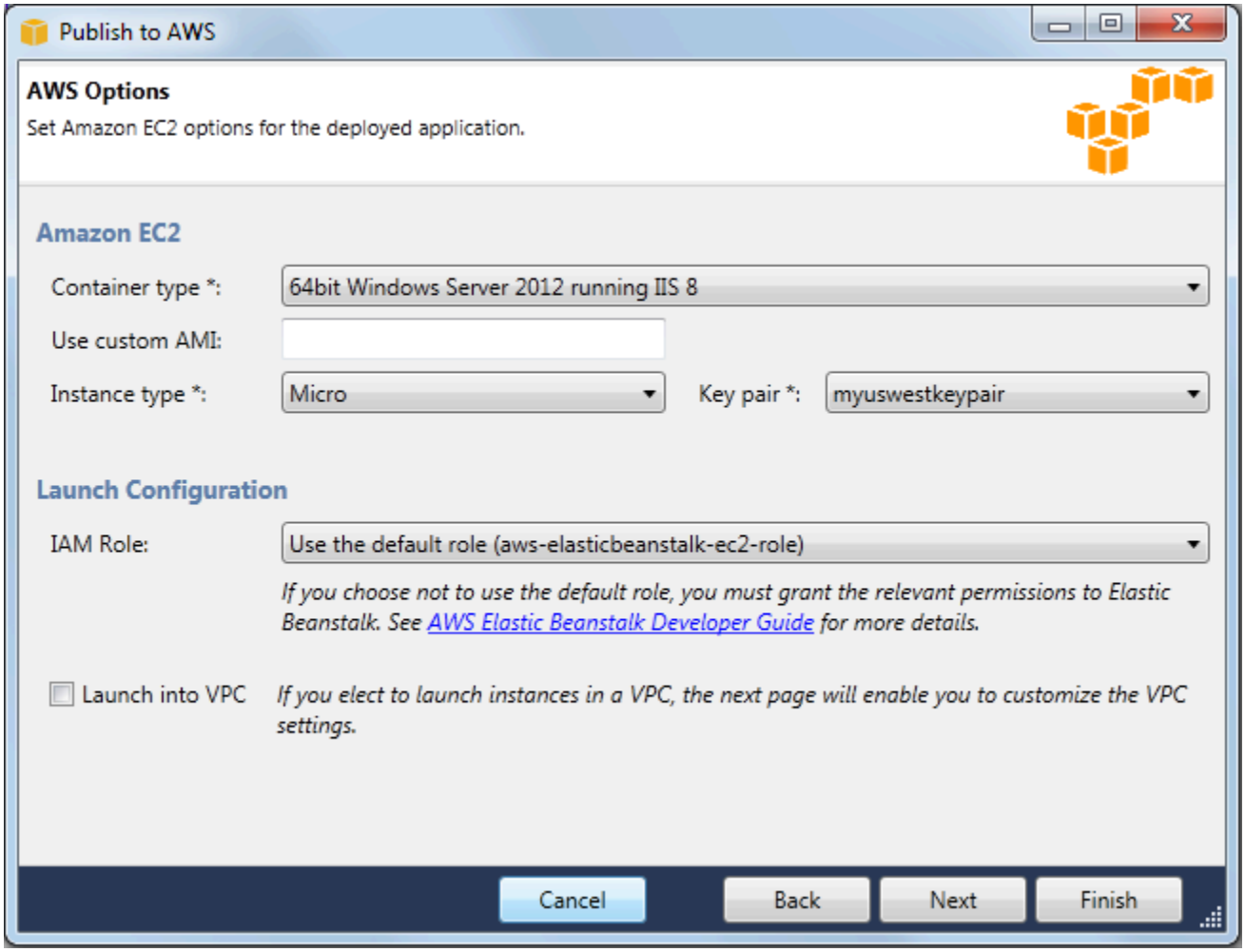
   b. Para Tipo de instância, selecione Micro.

   c. Para Par de chaves, selecione Criar novo par de chaves. Digite um nome para o novo par de chaves: neste exemplo, usamos **`myuswestkeypair`** e, depois, clique em OK. Um key pair permite o acesso por desktop remoto às suas instâncias da Amazon EC2 . Para obter mais informações sobre os pares de EC2 chaves da Amazon, consulte Usando credenciais no Guia do usuário do Amazon Elastic Compute Cloud.

   d. Selecione um perfil da instância.

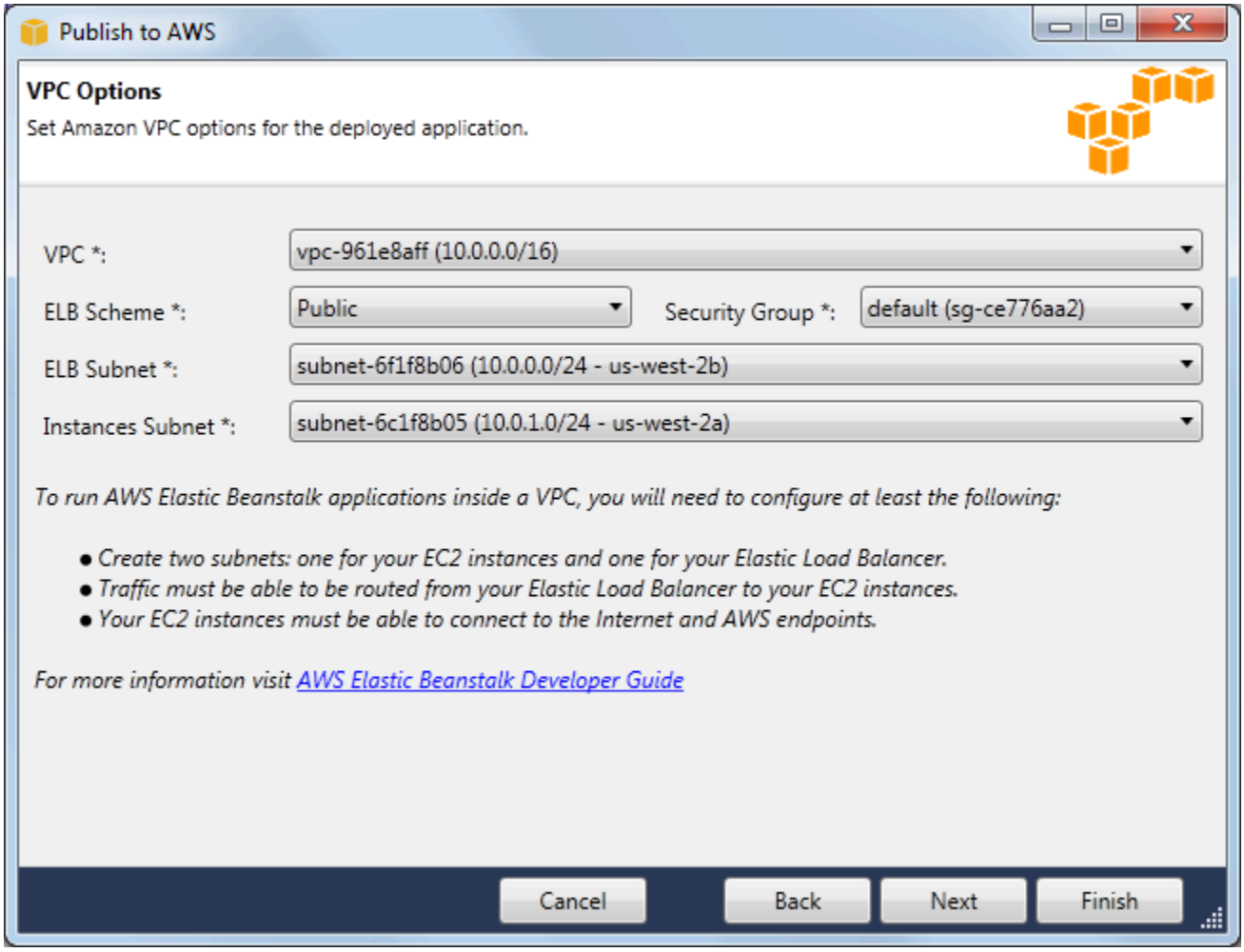
Se você não tiver um perfil da instância, selecione Criar um perfil da instância padrão. Para obter informações sobre como usar perfis de instância com o Elastic Beanstalk, consulte Gerenciar perfis de instância do Elastic Beanstalk.

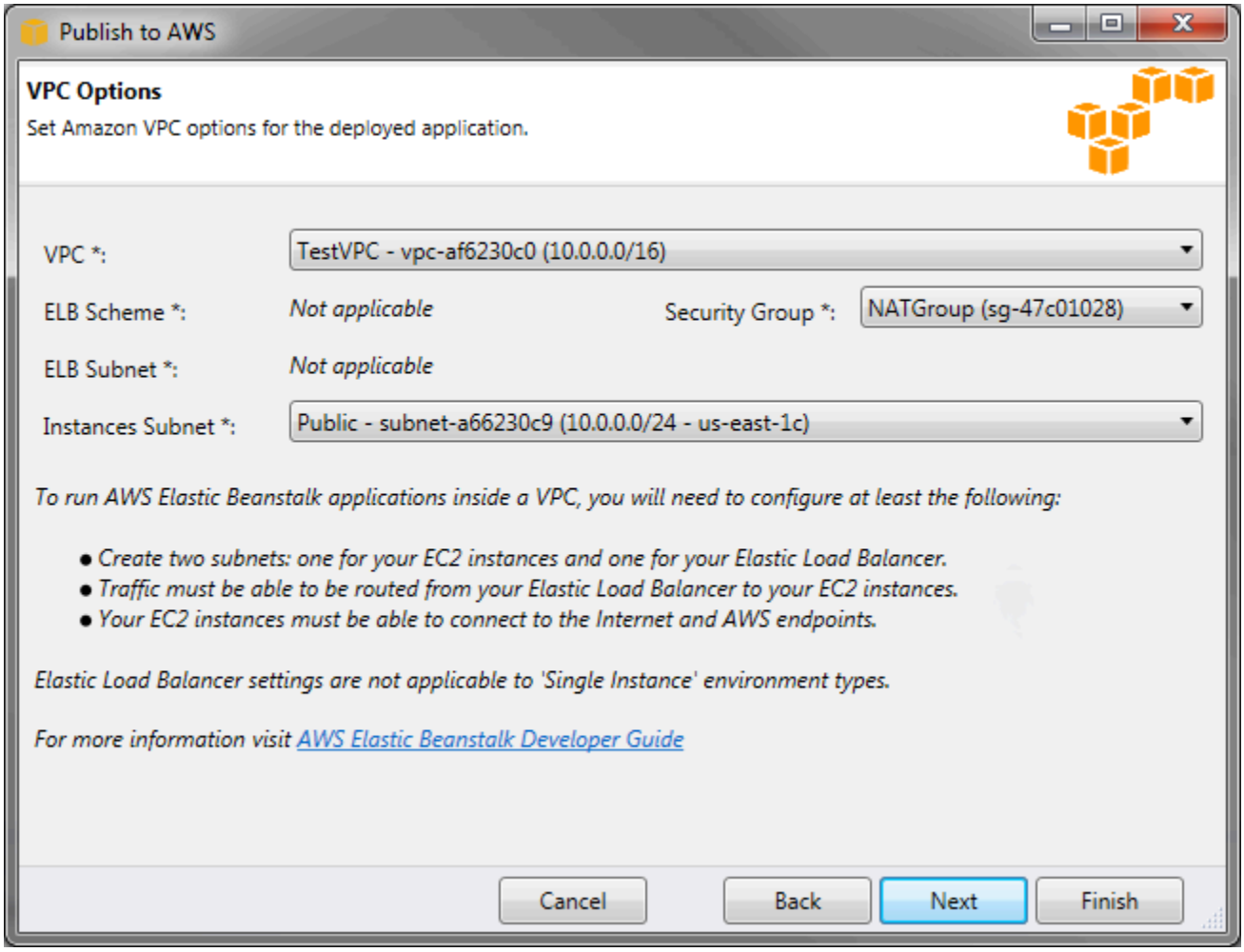
e. Se você tiver uma VPC personalizada que gostaria de usar com seu ambiente, clique em Iniciar no VPC. Você pode configurar as informações da VPC na próxima página. Para obter mais informações sobre a Amazon VPC, consulte Amazon Virtual Private Cloud (Amazon VPC). Para obter uma lista de tipos de contêiner não legados compatíveis, consulte the section called “Por que algumas versões de plataforma são marcadas como legadas?”



f. Clique em Next.

6. Se você optou por iniciar seu ambiente dentro de uma VPC, a Opções da VPC será exibida; caso contrário, a página Opções adicionais será exibida. Aqui você configurará as opções de sua 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. Selecione o ID da VPC em que gostaria de iniciar seu ambiente.

b. Para um ambiente escalável e com balanceamento de carga, selecione private (privado) em ELB Scheme (Esquema ELB) se não quiser que o Elastic Load Balancer fique disponível na Internet.

   Para um ambiente de única instância, essa opção não é aplicável porque o ambiente não tem um load balancer. Para obter mais informações, consulte Tipos de ambiente.

c. Para um ambiente escalável e com balanceamento de carga, selecione as sub-redes para o balanceador de carga elástico e as instâncias. EC2 Se você criou sub-redes públicas e privadas, certifique-se de que o balanceador de carga elástico e as EC2 instâncias estejam associados à sub-rede correta. Por padrão, a Amazon VPC cria uma sub-rede pública padrão usando 10.0.0.0/24 e uma sub-rede privada usando 10.0.1.0/24. Você pode visualizar suas sub-redes existentes no console da Amazon VPC em. https://console.aws.amazon.com/vpc/

Para um ambiente de única instância, a VPC precisa apenas de uma sub-rede pública para a instância. A seleção de uma sub-rede para o load balancer não é aplicável porque o ambiente não tem um load balancer. Para obter mais informações, consulte Tipos de ambiente.

d. Para um ambiente escalável e com balanceamento de carga, selecione o grupo de segurança criado para as instâncias, se aplicável.

   Para um ambiente de única instância, você não precisa de um dispositivo NAT. Selecione o security group padrão. O Elastic Beanstalk atribui um endereço IP elástico à instância, permitindo que ela acesse a Internet.

e. Clique em Next.

7. Na página Opções do aplicativo, configure as opções de seu aplicativo.

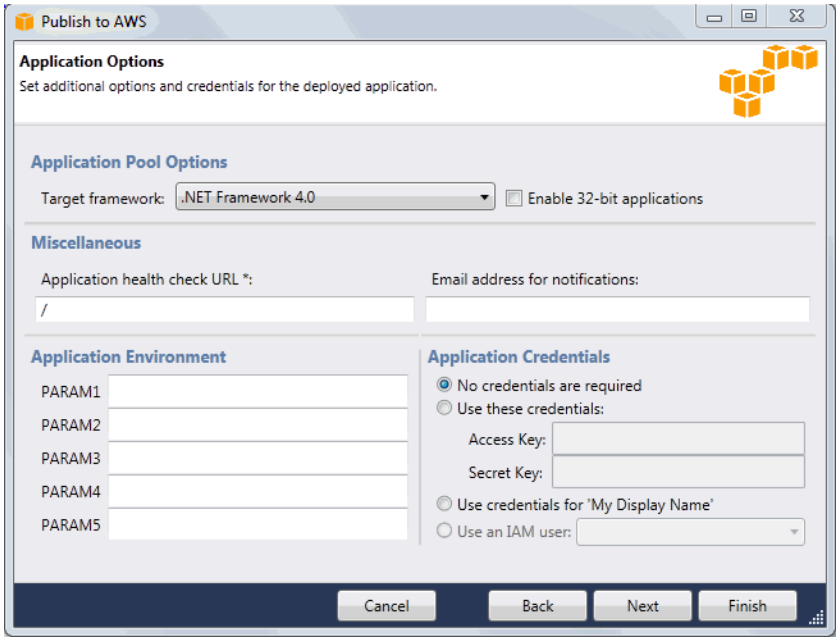
   a. Para o framework de destino, selecione .NET Framework 4.0.

   b. O Elastic Load Balancing usa uma verificação de saúde para determinar se as EC2 instâncias da Amazon que executam seu aplicativo estão íntegras. A verificação de integridade determina um status de integridade da instância sondando um URL especificado em um intervalo definido. Você pode substituir o URL padrão para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/index.aspx`), inserindo-o na caixa URL da verificação de integridade do aplicativo. Para obter mais informações sobre como configurar verificações de integridade do aplicativo, consulte Verificação de integridade.

   c. Digite um endereço de e-mail para receber notificações do Amazon Simple Notification Service (Amazon SNS) sobre eventos importantes que afetam o seu aplicativo.

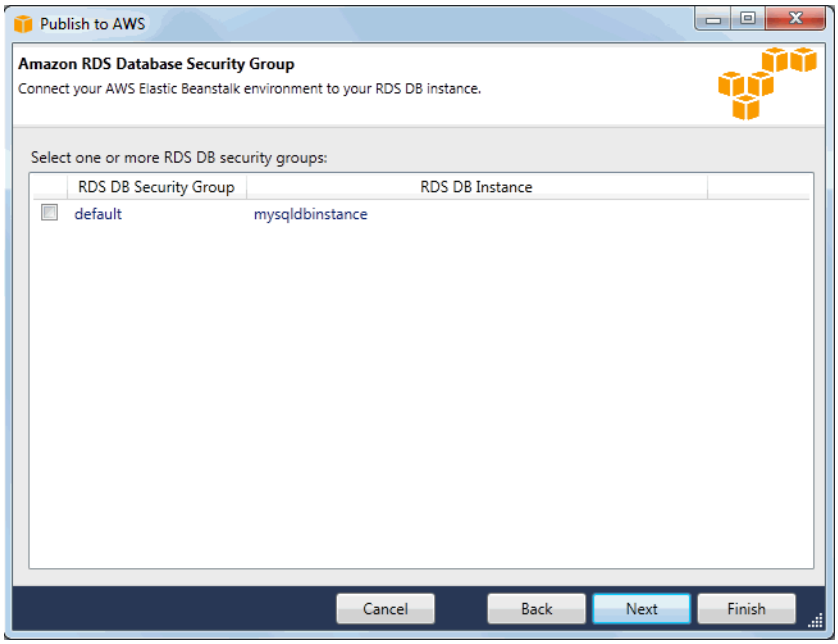
   d. A seção Ambiente do aplicativo permite que você especifique variáveis de ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. Essa configuração proporciona maior portabilidade, pois acaba com a necessidade de recompilar o código-fonte quando você move entre ambientes.

   e. Selecione a opção de credenciais do aplicativo que deseja usar para implantar seu aplicativo.

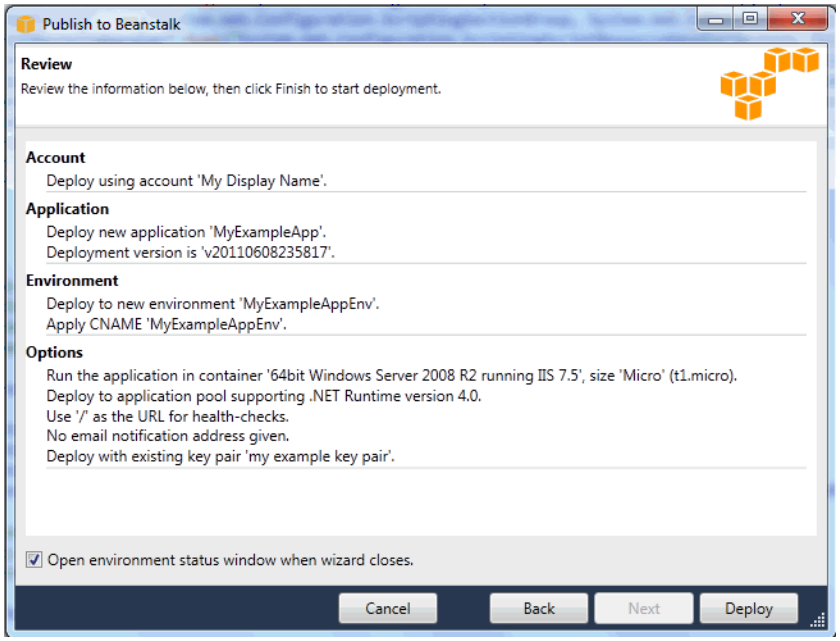


f. Clique em Next.

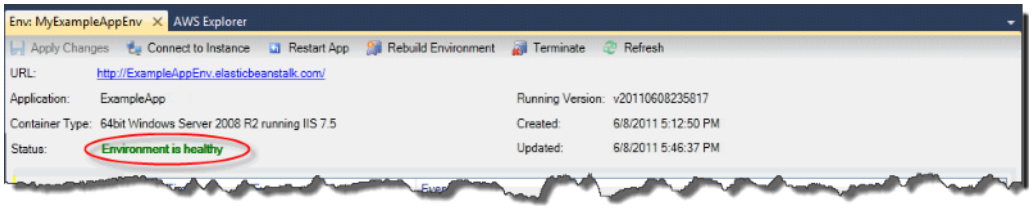
8. Se você já tiver configurado um banco de dados do Amazon RDS, a página Security group do banco de dados do Amazon RDS será exibida. Se você deseja conectar seu ambiente do Elastic Beanstalk com sua instância de banco de dados do Amazon RDS, selecione um ou mais security groups. Caso contrário, vá para a próxima etapa. Quando estiver pronto, clique em Próximo.



9. Revise as opções de implantação. Se tudo estiver como você deseja, clique em Implantar.



Seu projeto ASP.NET atualizado será exportado como um arquivo de implantação da web, enviado para o Amazon S3 e registrado como uma nova versão do aplicativo no Elastic Beanstalk. O recurso de implantação do Elastic Beanstalk irá monitorar seu ambiente até ele se tornar disponível com o código recém-implantado. Na guia env:<nome do ambiente>, você verá o status do seu ambiente.



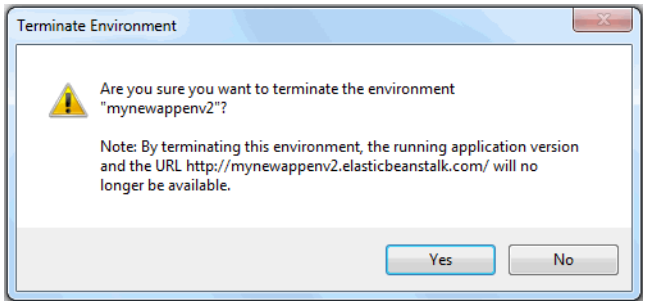
Encerrar um ambiente

Para evitar cobranças por AWS recursos não utilizados, você pode encerrar um ambiente em execução usando o AWS Toolkit for Visual Studio.

> **Note**
>
> Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Para encerrar um ambiente

1. Expanda o nó Elastic Beanstalk e o nó da aplicação no AWS Explorer. Clique com o botão direito do mouse no ambiente de seu aplicativo e selecione Encerrar ambiente.
2. Quando solicitado, clique em Sim para confirmar que você deseja encerrar o ambiente. O Elastic Beanstalk levará alguns minutos para AWS encerrar os recursos em execução no ambiente.



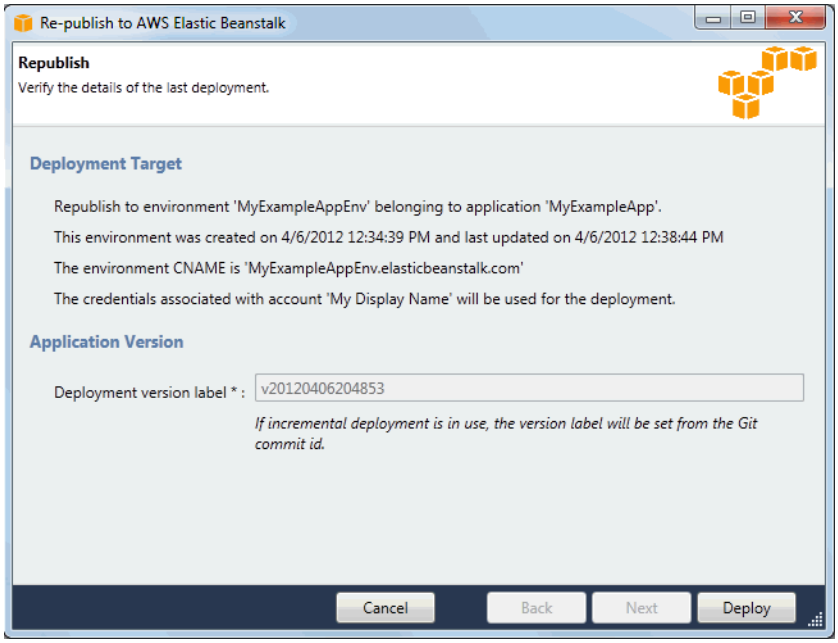
> **Note**
>
> Quando você encerra o ambiente, o CNAME associado ao ambiente encerrado torna-se disponível para qualquer pessoa utilizar.

Implantar em seu ambiente

Agora que você testou o seu aplicativo, é fácil editá-lo, reimplantá-lo e ver os resultados em instantes.

Para editar e reimplantar seu aplicativo web ASP.NET

1. No Solution Explorer, clique com o botão direito do mouse em seu aplicativo e clique em Republicar no ambiente < > ***your environment name***. O assistente Republicar no AWS Elastic Beanstalk é aberto.



2. Revise os detalhes de implantação e clique em Deploy.

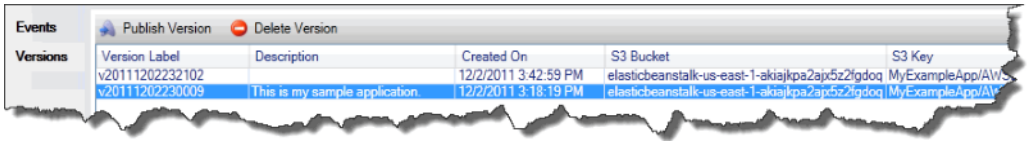
> **Note**
>
> Se você quiser alterar as configurações, clique em Cancelar e use, em vez disso, o assistente Publicar na AWS. Para obter instruções, consulte Criar um ambiente do Elastic Beanstalk.

Seu projeto web ASP.NET atualizado será exportado como um arquivo de implantação da web com o rótulo da nova versão, enviado para o Amazon S3 e registrado como uma nova versão do aplicativo no Elastic Beanstalk. O recurso de implantação do Elastic Beanstalk monitora seu ambiente existente até ele se tornar disponível com o código recém-implantado. Na guia env: < ***environment name*** >, você verá o status do seu ambiente.

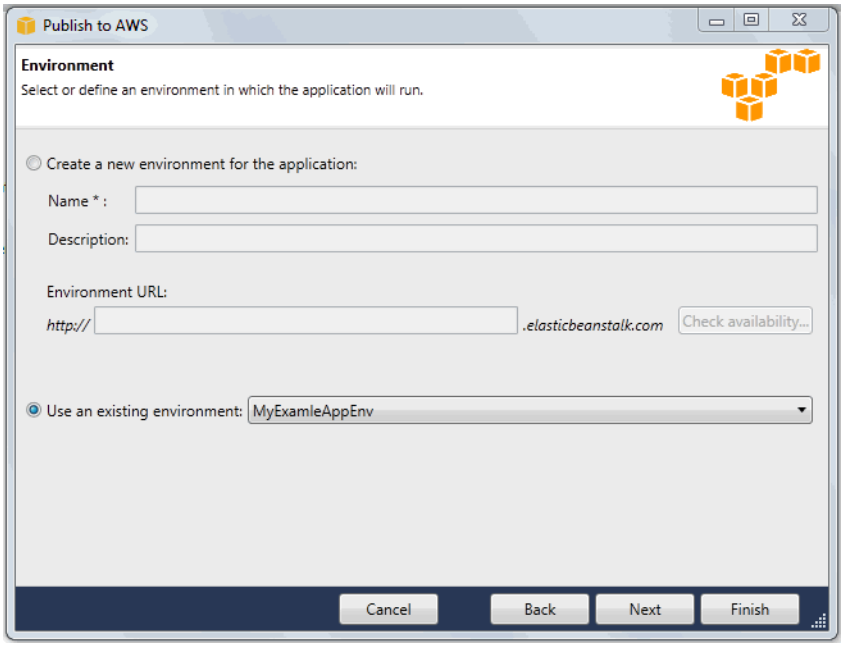
Você também pode implantar um aplicativo existente em um ambiente existente se, por exemplo, você precisar reverter para uma versão anterior do aplicativo.

Para implantar uma versão do aplicativo em um ambiente existente

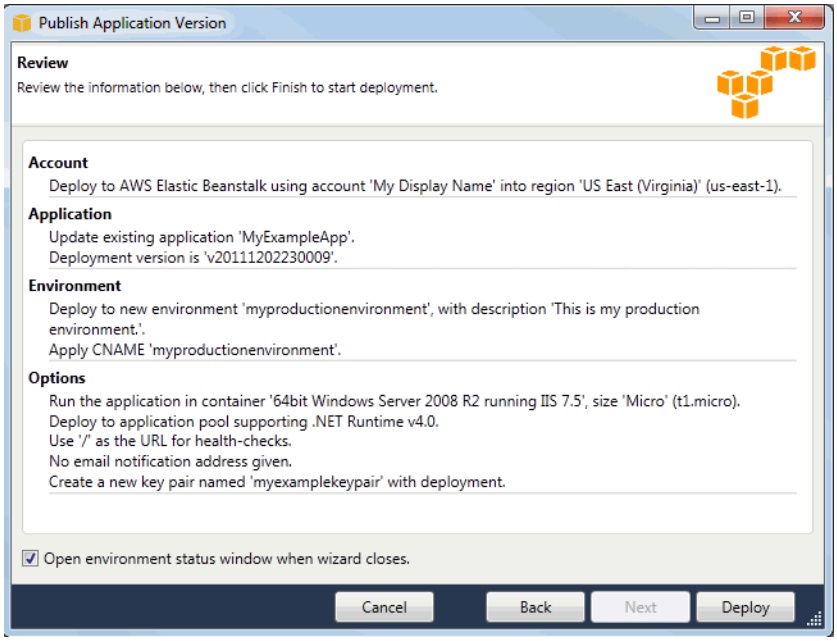
1. Clique com o botão direito do mouse na aplicação do Elastic Beanstalk expandindo o nó Elastic Beanstalk no AWS Explorer. Selecione View Status.
2. Na guia Aplicativo: < ***application name*** >, clique em Versões.



3. Clique na versão do aplicativo que você deseja implantar e clique em Publish Version.
4. No assistente Publish Application Version, clique em Next.



5. Analise as opções de implantação e clique em Deploy.



Seu projeto ASP.NET será exportado como um arquivo de implantação da web e enviado para o Amazon S3. O recurso de implantação do Elastic Beanstalk irá monitorar seu ambiente até ele se tornar disponível com o código recém-implantado. Na guia env: **< *environment name* >**, você verá o status do seu ambiente.

Gerenciar os ambientes de aplicações do Elastic Beanstalk

Com o AWS Toolkit for Visual Studio e AWS o Management Console, você pode alterar o provisionamento e a configuração dos recursos usados pelos AWS ambientes de aplicativos. Para obter informações sobre como gerenciar seus ambientes de aplicativos usando o AWS Management Console, consulteCriação de ambientes no Elastic Beanstalk. Esta seção discute as configurações de serviço específicas que você pode editar no AWS Toolkit for Visual Studio como parte da configuração de ambientes de aplicações.

Alterar os ajustes de configuração do ambiente

Quando você implanta seu aplicativo, o Elastic Beanstalk configura vários serviços de computação em nuvem. AWS Você pode controlar como esses serviços individuais são configurados usando o AWS Toolkit for Visual Studio.

Para editar as configurações de ambiente de um aplicativo

- Expanda o nó do Elastic Beanstalk e o nó da aplicação. Depois, clique com o botão direito no ambiente do Elastic Beanstalk no AWS Explorer. Selecione View Status.

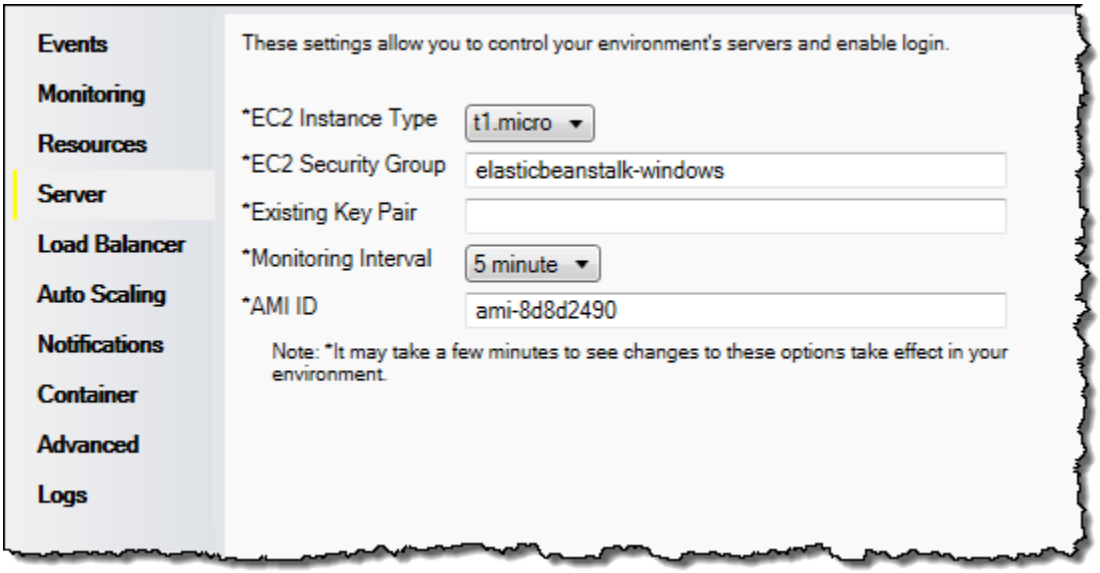
  Agora, você pode definir as seguintes configurações:

  - de aplicativos
  - Balanceamento de carga
  - Auto Scaling
  - Notificações
  - Propriedades de ambiente

Configurando instâncias EC2 do servidor usando o AWS kit de ferramentas do Visual Studio

O Amazon Elastic Compute Cloud (Amazon EC2) é um serviço web que você usa para iniciar e gerenciar instâncias de servidor nos datacenters da Amazon. Você pode usar as instâncias EC2 do servidor Amazon a qualquer momento, pelo tempo que precisar e para qualquer finalidade legal. As instâncias estão disponíveis em diferentes tamanhos e configurações. Para obter mais informações, acesse a Amazon EC2.

Você pode editar a configuração da instância Amazon do ambiente EC2 do Elastic Beanstalk com a guia Servidor dentro da guia do ambiente do aplicativo no Toolkit for AWS Visual Studio.



Tipos de EC2 instância da Amazon

Instance type (Tipo de instância) exibe os tipos de instância disponíveis para a aplicação do Elastic Beanstalk. Altere o tipo de instância para selecionar um servidor com as características (inclusive o tamanho da memória e a capacidade de CPU) mais adequadas à aplicação. Por exemplo, os aplicativos com operações muito intensas e demoradas podem exigir mais CPU ou memória.

Para obter mais informações sobre os tipos de EC2 instância da Amazon disponíveis para seu aplicativo Elastic Beanstalk, consulte Tipos de instância no Guia do usuário do Amazon Elastic Compute Cloud.

Grupos EC2 de segurança da Amazon

Você pode controlar o acesso ao seu aplicativo Elastic Beanstalk usando EC2 um grupo de segurança da Amazon. Um grupo de segurança define regras de firewall para suas instâncias.

Essas regras especificam o tráfego de rede de entrada que será entregue à sua instância. Qualquer outro tráfego de entrada será descartado. Você pode modificar as regras para um grupo a qualquer momento. As novas regras são aplicadas automaticamente para todas as instâncias em execução e as iniciadas no futuro.

Você pode configurar seus grupos de EC2 segurança da Amazon usando o AWS Management Console ou usando o AWS Toolkit for Visual Studio. Você pode especificar quais grupos de EC2 segurança da Amazon controlam o acesso ao seu aplicativo Elastic Beanstalk inserindo os nomes de um ou mais nomes de grupos de segurança da EC2 Amazon (delimitados por vírgulas) na caixa de texto Security Groups. EC2

> **Note**
>
> Certifique-se de que a porta 80 (HTTP) esteja acessível a partir de 0.0.0.0/0 como o intervalo CIDR de origem, caso deseje permitir verificações de integridade do seu aplicativo. Para obter mais informações sobre as verificações de integridade, consulte Verificações de integridade.

Para criar um grupo de segurança usando o AWS kit de ferramentas do Visual Studio

1. No Visual Studio, no AWS Explorer, expanda o EC2 nó da Amazon e clique duas vezes em Security Groups.
2. Clique em Create Security Group e insira um nome e uma descrição para o seu security group.
3. Clique em OK.

Para obter mais informações sobre os grupos EC2 de segurança da Amazon, consulte Como usar grupos de segurança no Guia do usuário do Amazon Elastic Compute Cloud.

Pares de EC2 chaves da Amazon

Você pode fazer login com segurança nas EC2 instâncias da Amazon provisionadas para seu aplicativo Elastic Beanstalk com um par de chaves da Amazon. EC2

> **Important**
>
> Você deve criar um par de EC2 chaves da Amazon e configurar suas instâncias da Amazon provisionadas pelo Elastic Beanstalk para usar o EC2 par de chaves da Amazon antes de

poder acessar suas EC2 instâncias da Amazon provisionadas pelo Elastic Beanstalk. EC2 Você pode criar seu par de chaves usando o AWS assistente Publish to dentro do AWS Toolkit for Visual Studio ao implantar seu aplicativo no Elastic Beanstalk. Se você quiser criar pares de chaves adicionais usando o Toolkit, siga as etapas abaixo. Como alternativa, você pode configurar seus pares de EC2 chaves da Amazon usando o AWS Management Console. Para obter instruções sobre como criar um par de chaves para a Amazon EC2, consulte o Guia de introdução do Amazon Elastic Compute Cloud.

A caixa de texto Existing Key Pair permite que você especifique o nome de um par de EC2 chaves da Amazon que você pode usar para fazer login com segurança nas EC2 instâncias da Amazon que executam seu aplicativo Elastic Beanstalk.

Para especificar o nome de um par de EC2 chaves da Amazon

1. Expanda o EC2 nó da Amazon e clique duas vezes em Key Pairs.
2. Clique em Create Key Pair e insira o nome do par de chaves.
3. Clique em OK.

Para obter mais informações sobre os pares de EC2 chaves da Amazon, acesse Como usar EC2 as credenciais da Amazon no Guia do usuário do Amazon Elastic Compute Cloud. Para obter mais informações sobre como se conectar às EC2 instâncias da Amazon, consulteListar e conectar com instâncias de servidor.

Intervalo de monitoramento

Por padrão, somente as CloudWatch métricas básicas da Amazon estão habilitadas. Elas retornam dados em intervalos de cinco minutos. Você pode habilitar CloudWatch métricas mais granulares de um minuto selecionando 1 minuto para o Intervalo de monitoramento na seção Servidor da guia Configuração do seu ambiente no. AWS Toolkit for Eclipse

> **Note**
>
> As taxas CloudWatch de serviço da Amazon podem ser aplicadas a métricas de intervalo de um minuto. Consulte a Amazon CloudWatch para obter mais informações.

ID da AMI personalizado

Você pode substituir a AMI padrão usada para suas EC2 instâncias da Amazon por sua própria AMI personalizada inserindo o identificador da sua AMI personalizada na caixa ID de AMI personalizada na seção Servidor da guia Configuração do seu ambiente no AWS Toolkit for Eclipse.

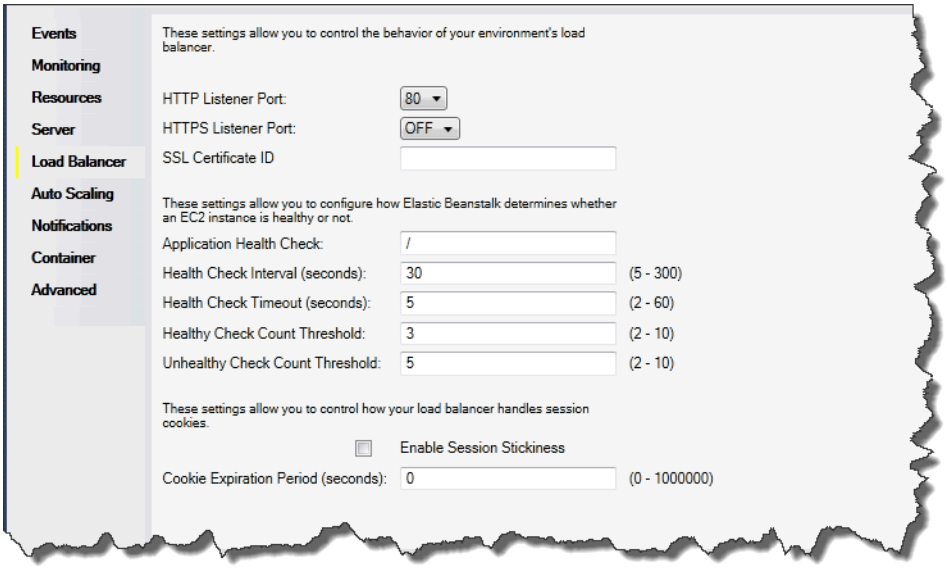
> **Important**
>
> O uso da AMI própria é uma tarefa avançada que deve ser feita com cuidado. Se você precisar de uma AMI personalizada, recomendamos começar com a AMI do Elastic Beanstalk padrão e depois modificá-la. Para serem consideradas saudáveis, o Elastic Beanstalk EC2 espera que as instâncias da Amazon atendam a um conjunto de requisitos, incluindo ter um gerenciador de host em execução. Se esses requisitos não forem atendidos, talvez o ambiente não funcione corretamente.

Configurando o Elastic Load Balancing usando AWS o kit de ferramentas para Visual Studio

O Elastic Load Balancing é um serviço web da Amazon que ajuda você a aprimorar a disponibilidade e a escalabilidade de seu aplicativo. Esse serviço facilita a distribuição de cargas de aplicativos entre duas ou mais EC2 instâncias da Amazon. O Elastic Load Balancing permite disponibilidade através de redundância e é compatível com aumento de tráfego do seu aplicativo.

O Elastic Load Balancing permite que você distribua e balanceie automaticamente o tráfego de entrada dos aplicativos em todas as instâncias que você está executando. O serviço também facilita a adição de novas instâncias, caso você precise aumentar a capacidade do aplicativo.

O Elastic Beanstalk provisiona automaticamente o Elastic Load Balancing ao implantar uma aplicação. Você pode editar a configuração da instância Amazon do ambiente EC2 do Elastic Beanstalk com a guia Load Balancer dentro da guia do ambiente do aplicativo no Toolkit AWS for Visual Studio.



As seções a seguir descrevem os parâmetros do Elastic Load Balancing que você pode configurar para o seu aplicativo.

Portas

O balanceador de carga provisionado para lidar com solicitações para seu aplicativo Elastic Beanstalk envia solicitações para as instâncias da Amazon que estão executando seu aplicativo. EC2 O balanceador de carga provisionado pode ouvir solicitações nas portas HTTP e HTTPS e rotear solicitações para as EC2 instâncias da Amazon em seu aplicativo. AWS Elastic Beanstalk Por padrão, o load balancer lida com solicitações na porta HTTP. Pelo menos uma das portas (HTTP ou HTTPS) deve estar ativada.



> **Important**
>
> Certifique-se de que a porta especificada não esteja bloqueada, caso contrário, os usuários não poderão se conectar à aplicação do Elastic Beanstalk.

Controle da porta HTTP

Para desativar a porta HTTP, selecione OFF em HTTP Listener Port. Para ativar a porta HTTP, você deve selecionar uma porta HTTP (por exemplo, a 80) na lista.

> **Note**
>
> Para acessar o ambiente usando uma porta diferente da porta 80 padrão, como a porta 8080, adicione uma escuta ao load balancer existente e configure a nova escuta nessa porta. Por exemplo, usando o AWS CLI for Classic load balancers, digite o comando a seguir, *LOAD_BALANCER_NAME* substituindo-o pelo nome do seu load balancer para o Elastic Beanstalk.
>
> ```
> aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
>  --listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
>  InstancePort=80"
> ```
>
> Por exemplo, usando o AWS CLI for Application Load Balancers, digite o comando a seguir, *LOAD_BALANCER_ARN* substituindo-o pelo ARN do seu load balancer para o Elastic Beanstalk.
>
> ```
> aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
>  --port 8080
> ```
>
> Se você quiser que o Elastic Beanstalk monitore o ambiente, não remova a escuta na porta 80.

Controle da porta HTTPS

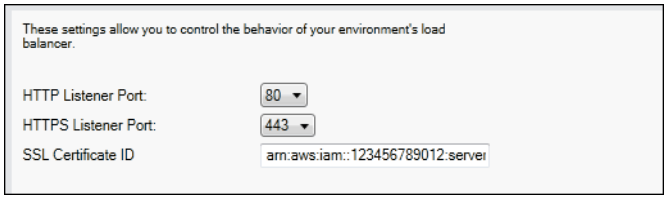
O Elastic Load Balancing oferece suporte ao protocolo HTTPS/TLS para permitir a criptografia do tráfego para conexões de clientes com o load balancer. As conexões do balanceador de carga com as EC2 instâncias usam criptografia de texto simples. Por padrão, a porta HTTPS é desativada.

Para ativar a porta HTTPS

1. Crie um novo certificado usando AWS Certificate Manager (ACM) ou carregue um certificado e uma chave para AWS Identity and Access Management (IAM). Para obter mais informações sobre como solicitar um certificado do ACM, consulte Solicitação de um certificado no Manual

do usuário do AWS Certificate Manager . Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte Importação de certificados no Manual do usuário do AWS Certificate Manager . Se o ACM não estiver disponível na sua região, use AWS Identity and Access Management (IAM) para carregar um certificado de terceiros. Os serviços ACM e IAM armazenarão o certificado e fornecerão um nome de recurso da Amazon (ARN) para o certificado SSL. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte Trabalhar com certificados de servidor no Trabalhar com certificados de servidor no Guia do usuário do IAM.

2. Especifique a porta HTTPS selecionando uma porta em HTTPS Listener Port.



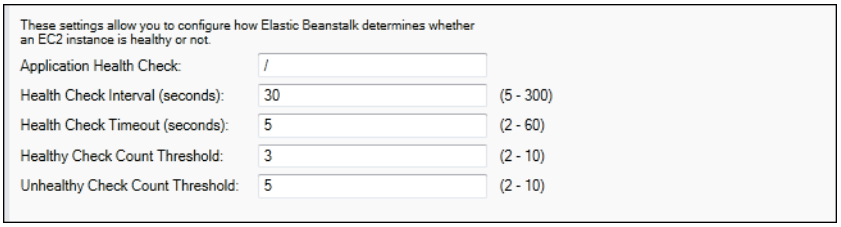
3. Para SSL Certificate ID (ID de certificado SSL), insira o Nome de recurso da Amazon (ARN) do seu certificado SSL. Por exemplo, o **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** ou o **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**. Use o certificado SSL que você criou ou do qual fez upload na etapa 1.

Para desativar a porta HTTPS, selecione OFF em HTTPS Listener Port.

Verificações de integridade

A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Por padrão, o Elastic Beanstalk usa TCP: 80 para contêineres não legados e HTTP: 80 para contêineres legados. Você pode substituir o URL padrão para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/default.aspx`), inserindo-o na caixa Application Health Check URL. Se você substituir o URL padrão, o Elastic Beanstalk usará HTTP para consultar o recurso. Para verificar se você está usando um tipo de contêiner legado, consulte the section called “Por que algumas versões de plataforma são marcadas como legadas?”

Você pode controlar as configurações da verificação de saúde usando a seção EC2 Instance Health Check do painel Load Balancing.



A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Substitua o URL padrão para corresponder a um recurso existente no seu aplicativo (por exemplo, `/myapp/index.jsp`), inserindo-o na caixa Application Health Check URL.

A lista a seguir descreve os parâmetros de verificação de integridade que você pode definir para o aplicativo.

- Em Health Check Interval (segundos), insira o número de segundos que o Elastic Load Balancing espera entre as verificações de saúde das instâncias da Amazon do seu aplicativo. EC2
- Em Health Check Timeout (seconds), especifique o número de segundos que o Elastic Load Balancing aguarda por uma resposta antes que ele considere que a instância não está respondendo.
- Em Healthy Check Count Threshold e Unhealthy Check Count Threshold, especifique o número de testes consecutivos de URL bem-sucedidos ou sem êxito antes que o Elastic Load Balancing altere o status de integridade da instância. Por exemplo, especificar **5** em Unhealthy Check Count Threshold significa que o URL deve retornar uma mensagem de erro ou tempo esgotado cinco vezes consecutivas antes que o Elastic Load Balancing considere que a verificação de integridade falhou.

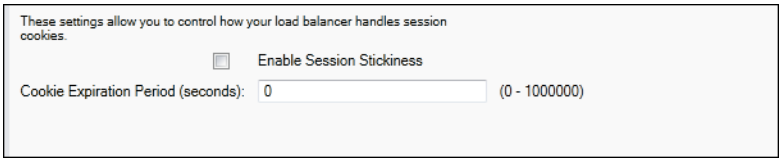
Sessões

Por padrão, um load balancer roteia cada solicitação de forma independente para a instância de servidor com a menor carga. Por comparação, uma sticky session vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.

O Elastic Beanstalk usa cookies HTTP gerados pelo balanceador de carga quando as sticky sessions são permitidas em uma aplicação. O load balancer usa um cookie especial gerado pelo load balancer para rastrear cada solicitação na instância do aplicativo. Quando o load balancer receber uma solicitação, ele primeiro verificará se esse cookie está presente na solicitação. Se estiver, a solicitação será enviada para a instância do aplicativo especificada no cookie. Se não houver um cookie, o load balancer selecionará uma instância de aplicativo com base no algoritmo de

balanceamento de carga existente. Um cookie será inserido na resposta para vincular as solicitações subsequentes do mesmo usuário para essa instância do aplicativo. A configuração da política define a expiração do cookie, que estabelece a validade de cada cookie.

Você pode usar a seção Sessions na guia Load Balancer para especificar se deseja ou não que o load balancer do seu aplicativo permita a perdurabilidade da sessão.



Para obter mais informações sobre o Elastic Load Balancing, consulte o Guia do desenvolvedor do Elastic Load Balancing.

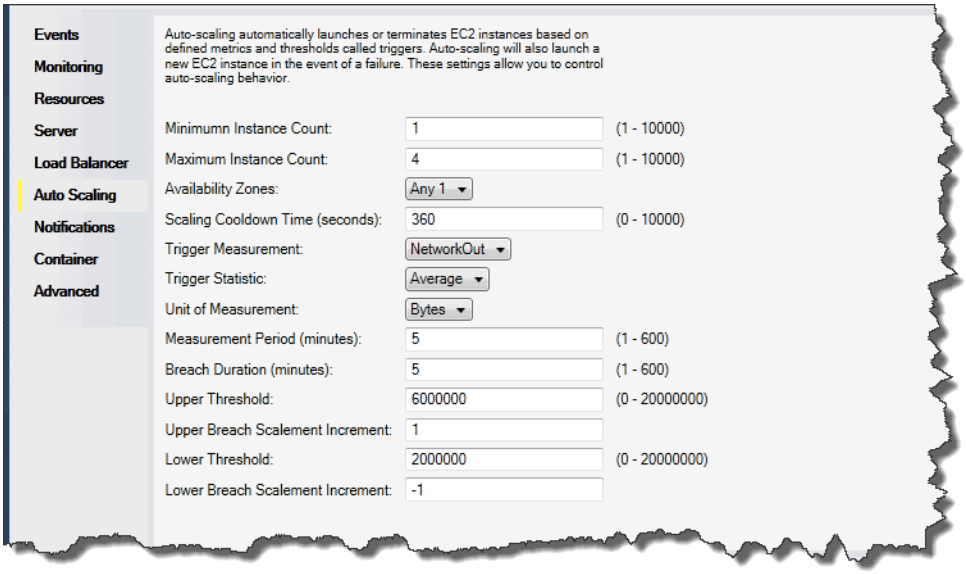
Configurando o Auto Scaling usando o kit de ferramentas AWS para Visual Studio

O Amazon EC2 Auto Scaling é um serviço web da Amazon projetado para iniciar ou encerrar automaticamente instâncias da EC2 Amazon com base em acionadores definidos pelo usuário. Os usuários podem configurar grupos de Auto Scaling e associar triggers a esses grupos para escalar automaticamente recursos de computação com base em métricas como uso de largura de banda ou utilização da CPU. O Amazon EC2 Auto Scaling trabalha com CloudWatch a Amazon para recuperar métricas para as instâncias do servidor que executam seu aplicativo.

O Amazon EC2 Auto Scaling permite que você pegue um grupo de EC2 instâncias da Amazon e defina vários parâmetros para que esse grupo aumente ou diminua automaticamente em número. O Amazon EC2 Auto Scaling pode adicionar ou remover EC2 instâncias da Amazon desse grupo para ajudá-lo a lidar perfeitamente com as mudanças de tráfego em seu aplicativo.

O Amazon EC2 Auto Scaling também monitora a integridade de cada EC2 instância da Amazon que ele executa. Se alguma instância for encerrada inesperadamente, o Amazon EC2 Auto Scaling detecta o encerramento e executa uma instância substituta. Esse recurso permite que você mantenha automaticamente um número fixo e desejado de EC2 instâncias da Amazon.

O Elastic Beanstalk provisiona o EC2 Amazon Auto Scaling para sua aplicação. Você pode editar a configuração da instância Amazon do ambiente EC2 do Elastic Beanstalk com a guia Auto Scaling dentro da guia do ambiente do aplicativo no Toolkit for AWS Visual Studio.

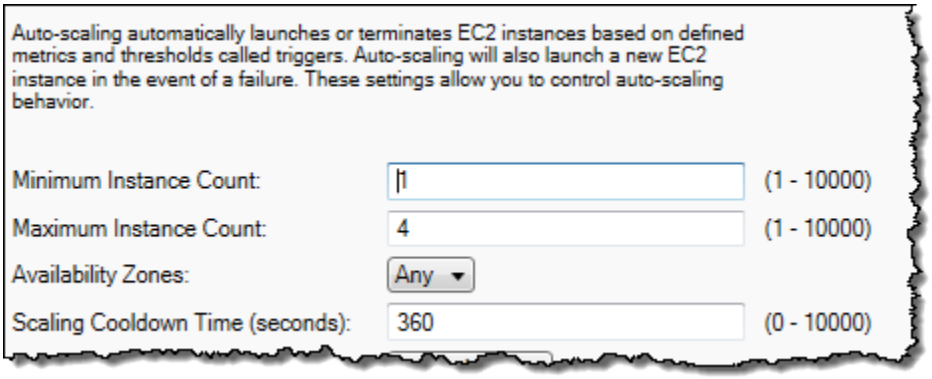


A seção a seguir discute como configurar os parâmetros de Auto Scaling para seu aplicativo.

Iniciar a configuração

Você pode editar a configuração de lançamento para controlar como seu aplicativo Elastic Beanstalk provisiona os recursos do EC2 Amazon Auto Scaling.

As caixas Minimum Instance Count (Contagem de instância mínima) e Maximum Instance Count (Contagem de instância máxima) permitem que você especifique os tamanhos mínimo e máximo do grupo de Auto Scaling que a aplicação do Elastic Beanstalk usa.



> **Note**
>
> Para manter um número fixo de EC2 instâncias da Amazon, defina a Contagem mínima de instâncias e a Contagem máxima de instâncias com o mesmo valor.

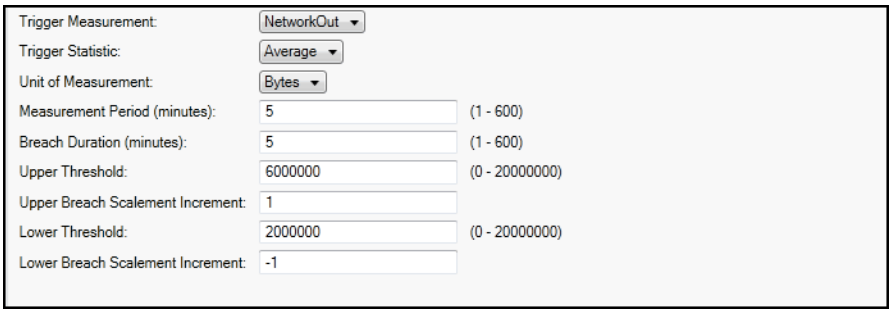
A caixa Zonas de disponibilidade permite que você especifique o número de zonas de disponibilidade nas quais você deseja que suas EC2 instâncias da Amazon estejam. É importante definir esse número se quiser criar aplicativos tolerantes a falhas. Se uma Zona de disponibilidade ficar inativa, suas instâncias continuarão sendo executadas em suas outras Zonas de disponibilidade.

> **Note**
>
> No momento, não é possível especificar a zona de disponibilidade na qual sua instância estará.

Acionadores

Um gatilho é um mecanismo do Amazon EC2 Auto Scaling que você configura para informar ao sistema quando você quer aumentar (escalar) o número de instâncias e quando você quer diminuir (escalar) o número de instâncias. Você pode configurar gatilhos para serem acionados em qualquer métrica publicada na Amazon CloudWatch, como a utilização da CPU, e determinar se as condições especificadas foram atendidas. Quando o limite superior ou inferior das condições que você especificou para a métrica forem violados para o período especificado, o trigger iniciará um processo de longa duração chamado de ação de escalabilidade.

Você pode definir um gatilho de escalabilidade para seu aplicativo Elastic Beanstalk usando o Toolkit AWS for Visual Studio.



Os gatilhos do Amazon EC2 Auto Scaling funcionam observando uma CloudWatch métrica específica da Amazon para uma instância. Os triggers incluem utilização de CPU, tráfego de rede e a atividade do disco. Use a configuração Trigger Measurement para selecionar uma métrica para o trigger.

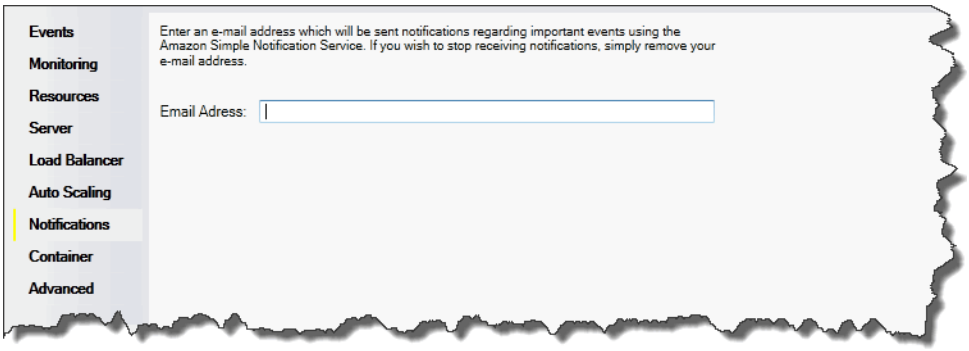
A lista a seguir descreve os parâmetros de acionamento que você pode configurar usando o AWS Management Console.

- Você pode especificar quais estatísticas o trigger deve usar. Você pode selecionar Minimum, Maximum, Sum ou Average em Trigger Statistic.
- Em Unit of Measurement, especifique a unidade de medida do trigger.
- O valor na caixa Período de medição especifica com que frequência a Amazon CloudWatch mede as métricas do seu gatilho. A Breach Duration é a quantidade de tempo que uma métrica pode ficar além do limite definido (conforme especificado em Upper Threshold e Lower Threshold) antes que o trigger seja acionado.
- Para incremento superior da escala de violação e incremento inferior da escala de violação, especifique quantas EC2 instâncias da Amazon adicionar ou remover ao realizar uma atividade de escalabilidade.

Para obter mais informações sobre o Amazon EC2 Auto Scaling, consulte a seção Amazon Auto EC2 Scaling na documentação do Amazon Elastic Compute Cloud.

Configurando notificações usando o AWS kit de ferramentas para Visual Studio

O Elastic Beanstalk usa o Amazon Simple Notification Service (Amazon SNS) para notificar você de eventos importantes que afetam sua aplicação. Para permitir notificações do Amazon SNS, basta inserir seu endereço de e-mail na caixa Email Address. Para desativar essas notificações, remova seu endereço de e-mail da caixa.



Configurando contêineres.NET usando o AWS kit de ferramentas para Visual Studio

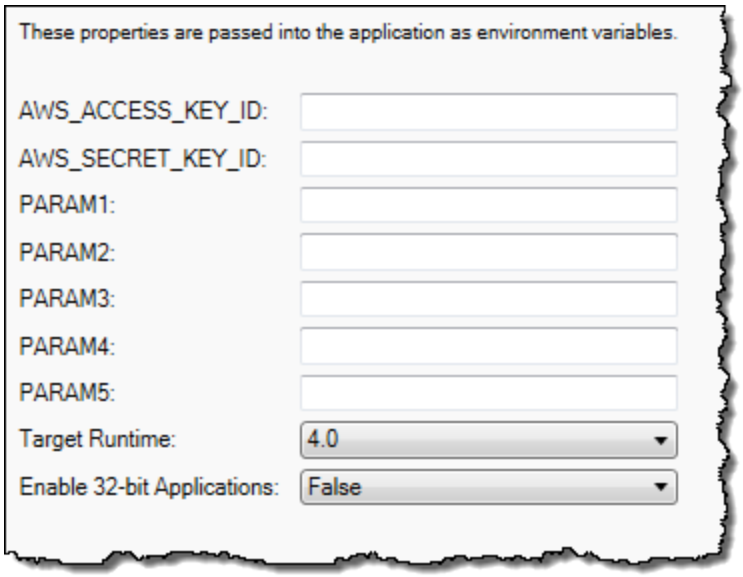
O painel Container/.net Options permite que você ajuste o comportamento de suas EC2 instâncias da Amazon e habilite ou desabilite a rotação de logs do Amazon S3. Você pode usar o AWS Toolkit for Visual Studio para configurar as informações do contêiner.

> **Note**
>
> É possível modificar as definições de configuração com tempo de inatividade zero, basta trocar o CNAME dos ambientes. Para obter mais informações, consulte Implantações azuis/verdes com o Elastic Beanstalk.

Se desejar, você pode estender o número de parâmetros. Para obter mais informações sobre como estender os parâmetros, consulte Configurações de opção.

Como acessar o painel de opções de contêiner/.NET para a aplicação do Elastic Beanstalk

1. No AWS Toolkit for Visual Studio, expanda o nó do Elastic Beanstalk e o nó do seu aplicativo.
2. No AWS Explorer, clique duas vezes no ambiente do Elastic Beanstalk.
3. Na parte inferior do painel Overview, clique na guia Configuration.
4. Em Container, configure as opções de contêiner.

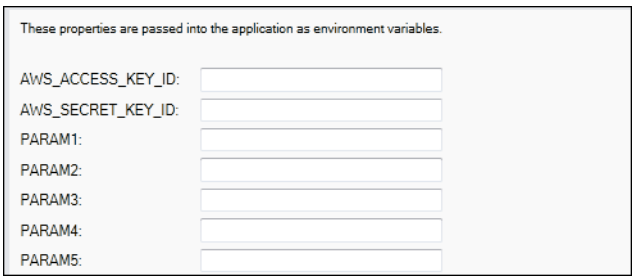


Opções de contêiner .NET

Você pode escolher a versão do. NET Framework para o seu aplicativo. Escolha 2.0 ou 4.0 para o Target runtime. Selecione Enable 32-bit Applications se quiser permitir aplicativos de 32 bits.

Configurações do aplicativo

A seção Application Settings permite que você defina as variáveis de ambiente que você pode ler do código do seu aplicativo.



Gerenciar contas

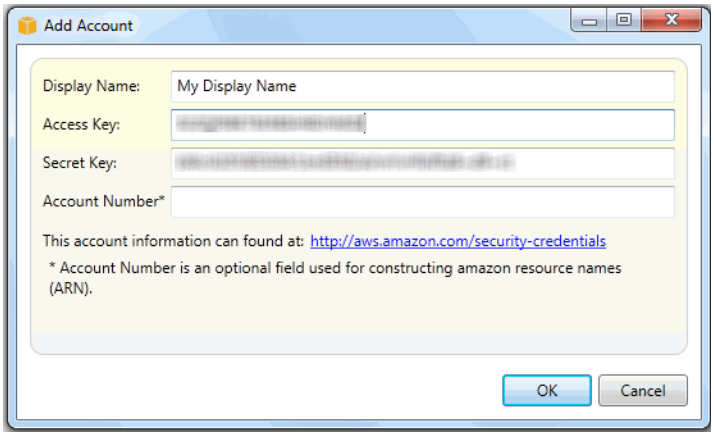
Se você quiser configurar AWS contas diferentes para realizar tarefas diferentes, como teste, preparação e produção, você pode adicionar, editar e excluir contas usando o AWS Toolkit for Visual Studio.

Para gerenciar várias contas

1. No Visual Studio, no menu View (Exibir), clique em AWS Explorer.
2. Ao lado da lista Conta, clique no botão Adicionar Conta.



A caixa de diálogo Adicionar Conta é exibida.



3. Preencha as informações solicitadas.

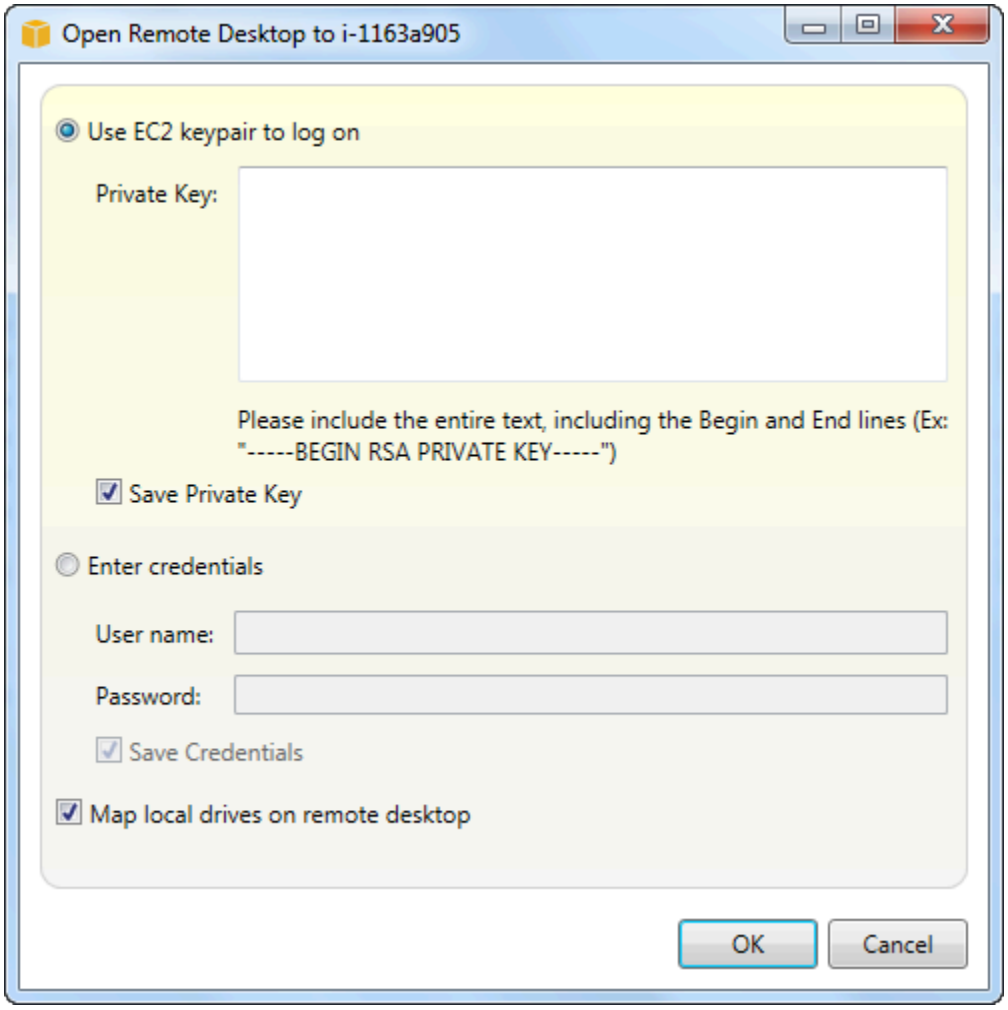
4. As informações da sua conta aparecem na guia AWS Explorer. Quando você publicar no Elastic Beanstalk, poderá selecionar a conta que deseja usar.

Listar e conectar com instâncias de servidor

Você pode ver uma lista de EC2 instâncias da Amazon executando seu ambiente de aplicativos do Elastic Beanstalk por meio do Toolkit AWS for Visual Studio ou do Management Console. AWS Você pode se conectar a essas instâncias usando a conexão de desktop remoto. Para obter informações sobre como listar e se conectar às instâncias do seu servidor usando o AWS Management Console, consulteListar e conectar com instâncias de servidor. A seção a seguir mostra e conecta você às instâncias do servidor usando o AWS Toolkit for Visual Studio.

Para visualizar e se conectar às EC2 instâncias da Amazon para um ambiente

1. No Visual Studio, no AWS Explorer, expanda o EC2 nó da Amazon e clique duas vezes em Instâncias.
2. Clique com o botão direito do mouse no ID da EC2 instância da Amazon em execução no balanceador de carga do seu aplicativo na coluna Instância e selecione Abrir área de trabalho remota no menu de contexto.



3. Selecione Usar EC2 par de chaves para fazer login e colar o conteúdo do arquivo de chave privada que você usou para implantar seu aplicativo na caixa Chave privada. Como alternativa, insira seu nome de usuário e senha nas caixas de texto Nome de usuário e Senha.

> **Note**
>
> Se o par de chaves estiver armazenado no Toolkit, a caixa de texto não será exibida.

4. Clique em OK.

Monitorar a integridade do aplicativo

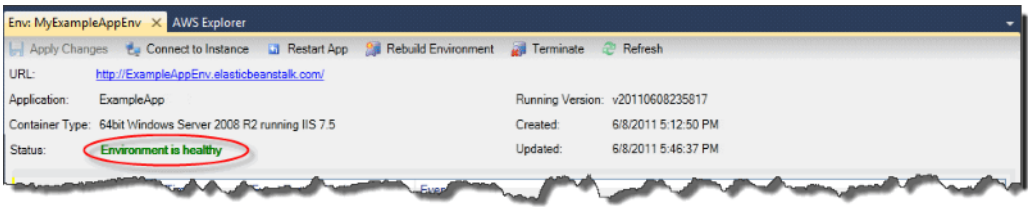
Quando você executa um site de produção, é importante saber que seu aplicativo está disponível e respondendo às solicitações. Para ajudar a monitorar a capacidade de resposta da aplicação, o

Elastic Beanstalk dispõe de recursos que monitoram as estatísticas sobre a aplicação e criam alertas que são acionados quando os limites são excedidos.

Para obter informações sobre o monitoramento de integridade fornecido pelo Elastic Beanstalk, consulte Relatórios de integridade básica.

Você pode acessar informações operacionais sobre seu aplicativo usando o AWS Toolkit for Visual Studio ou AWS o Management Console.

O toolkit exibe o status do ambiente e a integridade do aplicativo no campo Status.



Para monitorar a integridade do aplicativo

1. No AWS Toolkit for Visual Studio, AWS no Explorer, expanda o nó do Elastic Beanstalk e, em seguida, expanda o nó do seu aplicativo.
2. Clique com o botão direito do mouse no ambiente do Elastic Beanstalk e, depois, clique em View Status (Visualizar status).
3. Na guia do seu ambiente de aplicativos, clique em Monitoramento.

   O painel Monitoramento inclui um conjunto de gráficos que mostra o uso de recursos do seu ambiente de aplicativos específico.

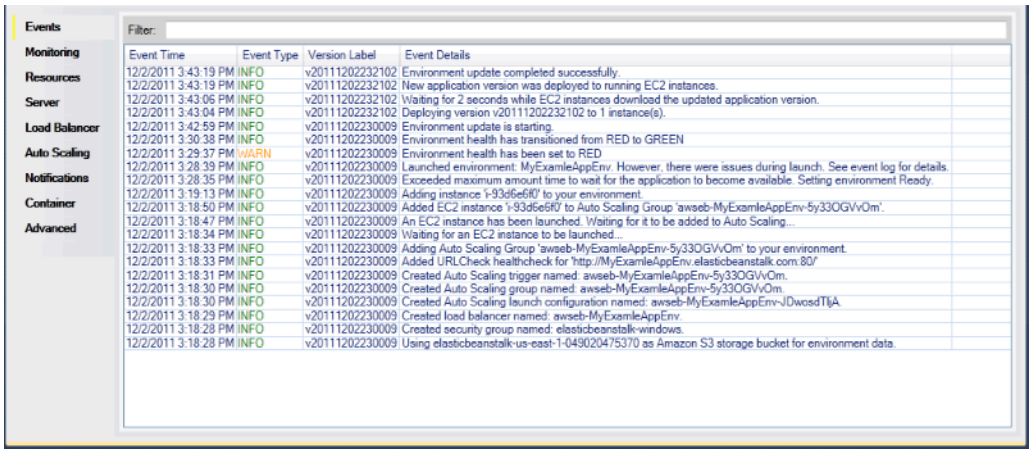


> **Note**
>
> Por padrão, o intervalo de tempo é definido como a última hora. Para modificar essa configuração, na lista Período, clique em um período diferente.

Você pode usar o AWS Toolkit for Visual Studio ou AWS o Management Console para visualizar eventos associados ao seu aplicativo.

Para visualizar eventos do aplicativo

1. No AWS Toolkit for Visual Studio, AWS no Explorer, expanda o nó do Elastic Beanstalk e o nó do seu aplicativo.
2. Clique com o botão direito do mouse no ambiente do Elastic Beanstalk em AWS Explorer e, depois, clique em View Status (Visualizar status).
3. Na guia do seu ambiente de aplicativos, clique em Eventos.



Implantar aplicações do Elastic Beanstalk em .NET usando a ferramenta de implantação

O AWS Toolkit for Visual Studio inclui uma ferramenta de implantação, uma ferramenta de linha de comando que fornece a mesma funcionalidade do assistente de implantação AWS no Toolkit. Você pode usar a ferramenta de implantação em seu pipeline da compilações ou em outros scripts para automatizar as implantações no Elastic Beanstalk.

A ferramenta de implantação oferece suporte a implantações iniciais e reimplantações. Se você já implantou seu aplicativo usando a ferramenta de implantação, pode reimplantá-lo usando o assistente de implantação no Visual Studio. Da mesma forma, se você implantou usando o assistente, pode reimplantar usando a ferramenta de implantação.

> **Note**
>
> A ferramenta de implantação não aplica os valores recomendados às opções de configuração, como o console ou a EB CLI. Use os arquivos de configuração para garantir que as configurações necessárias sejam configuradas quando você iniciar seu ambiente.

Este capítulo orienta você pelo processo de implantação de uma aplicação .NET demonstrativa no Elastic Beanstalk usando a ferramenta de implantação e, depois, pelo processo de reimplantação dela usando uma implantação incremental. Para ver mais detalhes sobre a ferramenta de implantação, incluindo as opções de parâmetros, consulte Ferramenta de implantação.

Pré-requisitos

Para usar a ferramenta de implantação, você precisa instalar o AWS Toolkit for Visual Studio. Para obter informações sobre os pré-requisitos e as instruções de instalação, consulte AWS Toolkit for Microsoft Visual Studio.

A ferramenta de implantação costuma ser instalada em um dos seguintes diretórios no Windows:

32 bits	64 bits
`C:\Program Files\AWS Tools\Deployment Tool\awsdeploy.exe`	`C:\Program Files (x86)\AWS Tools\Deployment Tool\awsdeploy.exe`

Implantar no Elastic Beanstalk

Para implantar a aplicação demonstrativa no Elastic Beanstalk usando a ferramenta de implantação, você precisa primeiro modificar o arquivo de configuração `ElasticBeanstalkDeploymentSample.txt`, que é fornecido no diretório `Samples`. Esse arquivo de configuração contém as informações necessárias para implantar seu aplicativo, incluindo

o nome do aplicativo, a versão do aplicativo, o nome do ambiente e suas credenciais de AWS acesso. Após modificar o arquivo de configuração, você poderá usar a linha de comando para implantar o aplicativo de exemplo. O arquivo de implantação Web é carregado para o Amazon S3 e registrado como uma nova versão da aplicação com o Elastic Beanstalk. A implantação do aplicativo leva alguns minutos. Quando o ambiente estiver íntegro, a ferramenta de implantação emitirá um URL para o aplicativo em execução.

Como implantar uma aplicação .NET no Elastic Beanstalk

1. No `Samples` subdiretório em que a ferramenta de implantação está instalada, abra `ElasticBeanstalkDeploymentSample.txt` e insira sua chave de AWS acesso e chave AWS secreta, como no exemplo a seguir.

   ```
   ### AWS Access Key and Secret Key used to create and deploy the application
    instance
   AWSAccessKey = AKIAIOSFODNN7EXAMPLE
   AWSSecretKey = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   ```

   > **Note**
   >
   > Para acesso à API, é necessário ter um ID de chave de acesso e uma chave de acesso secreta. Use as chaves de acesso do usuário do IAM em vez das chaves de acesso da Usuário raiz da conta da AWS . Para obter mais informações sobre a criação de chaves de acesso, consulte Gerenciar chaves de acesso para usuários do IAM no Guia do usuário do IAM.

2. No prompt de linha de comando, digite o seguinte:

   ```
   C:\Program Files (x86)\AWS Tools\Deployment Tool>awsdeploy.exe /w Samples
   \ElasticBeanstalkDeploymentSample.txt
   ```

   A implantação do aplicativo leva alguns minutos. Se a implantação for bem-sucedida, você verá a mensage, `Application deployment completed; environment health is Green`.

   > **Note**
   >
   > Se você receber o seguinte erro, o CNAME já existe.

```
[Error]: Deployment to AWS Elastic Beanstalk failed with exception: DNS name
 (MyAppEnv.elasticbeanstalk.com) is not available.
```

Como o CNAME deve ser exclusivo, você precisa alterar `Environment.CNAME` em `ElasticBeanstalkDeploymentSample.txt`.

3. No navegador da Web, acesse o URL do seu aplicativo em execução. O formato do URL é <CNAME.elasticbeanstalk.com> (e., **`MyAppEnv.elasticbeanstalk.com`**).

Migrar a aplicação .NET no local para o Elastic Beanstalk

AWS Elastic Beanstalk fornece um caminho de migração simplificado para seus aplicativos Windows executados nos Serviços de Informações da Internet (IIS) por meio da Interface de Linha de Comando (EB CLI) do Elastic Beanstalk. O eb migrate comando descobre automaticamente seus sites, aplicativos e diretórios virtuais do IIS, preserva suas configurações e os implanta na nuvem. AWS

Esse recurso de migração integrado oferece uma abordagem mais simples que reduz a complexidade e o tempo normalmente associados às migrações para a nuvem. O processo de migração ajuda a manter a funcionalidade do aplicativo e a integridade da configuração durante a transição para AWS o.

Para obter instruções completas e detalhadas sobre como migrar seus aplicativos do IIS para AWS Elastic Beanstalk, consulte o Migração de aplicativos IIS capítulo deste guia.

Recomendações para componentes retirados do Windows Server no Elastic Beanstalk

Este tópico fornece recomendações se suas aplicações estiverem sendo executadas atualmente nas ramificações da plataforma retiradada do Windows Server 2012 R2. Ele também aborda o suporte obsoleto para as versões dos protocolos TLS 1.0 e 1.1 em nossos endpoints de API de AWS serviço e nas ramificações da plataforma afetada.

Descontinuação das ramificações da plataforma Windows Server 2012 R2

O Elastic Beanstalk retirou as filiais da plataforma Windows Server 2012 R2 em 4 de dezembro de 2023 e AMIs tornou privadas as associadas a essas plataformas em 10 de abril de 2024. Essa ação

impede o lançamento de instâncias em ambientes Windows Server 2012 que usam a AMI padrão do Beanstalk.

Se tiver algum ambiente em execução em filiais retiradadas da plataforma Windows, recomendamos que você o migre para uma das seguintes plataformas do Windows Server, que são atuais e totalmente compatíveis:

- Windows Server 2022 com IIS 10.0 versão 2.x
- Windows Server 2019 com IIS 10.0 versão 2.x

Veja considerações completas sobre migração em Migrar de versões principais anteriores da plataforma do Windows Server.

Para obter mais informações sobre a substituição da plataforma, consulte Política de suporte às plataformas do Elastic Beanstalk.

> **Note**
>
> Se você não conseguir migrar para essas plataformas totalmente suportadas, recomendamos usar a imagem personalizada AMIs criada com o Windows Server 2012 R2 ou o Windows Server 2012 R2 Core AMIs como imagem base, caso ainda não tenha feito isso. Para obter instruções detalhadas, consulte Preservar o acesso a uma imagem de máquina da Amazon (AMI) para uma plataforma descontinuada. Entre em contato com o Centro de suporte da AWS se precisar de acesso temporário a uma AMI enquanto executa uma dessas etapas de migração.

Compatibilidade com TLS 1.2

Em 31 de dezembro de 2023, AWS começou a aplicar totalmente o TLS 1.2 em todos os endpoints AWS da API. Essa ação removeu a capacidade de usar as versões 1.0 e 1.1 do TLS com todos AWS APIs. Essas informações foram originalmente comunicadas em 28 de junho de 2022. Para evitar o risco de impacto na disponibilidade, atualize todos os ambientes que estejam executando as versões de plataforma identificadas aqui para uma versão mais recente o mais rápido possível, caso ainda não tenha feito isso.

Impacto potencial

As versões das plataformas Elastic Beanstalk que executam o TLS v1.1 ou anterior são afetadas. Essa alteração afeta as ações do ambiente que incluem, entre outras, as seguintes: implementações de configuração, implementações de aplicações, ajuste de escala automático, lançamento de novo ambiente, rotação de logs, relatórios de integridade aprimorados e publicação de logs de aplicações no bucket do Amazon S3 associado às suas aplicações.

Versões da plataforma Windows afetadas

Os clientes com ambientes Elastic Beanstalk na seguinte versão de plataforma são aconselhados a atualizar cada um de seus ambientes correspondentes para a plataforma Windows versão 2.8.3 ou posterior, lançada em 18 de fevereiro de 2022.

- Windows Server 2019 - versão da plataforma 2.8.2 ou anteriores

Os clientes com ambientes Elastic Beanstalk nas seguintes versões de plataforma são aconselhados a atualizar cada um de seus ambientes correspondentes para a plataforma Windows versão 2.10.7 ou posterior, lançada em 28 de dezembro de 2022.

- Windows Server 2016 - versão da plataforma 2.10.6 ou versões anteriores
- Windows Server 2012: todas as versões da plataforma; essa plataforma foi descontinuada em 4 de dezembro de 2023
- Windows Server 2008 — todas as versões da plataforma; essa plataforma foi desativada em 28 de outubro de 2019

Para obter a lista das versões da plataforma Windows Server mais recentes e compatíveis, consulte Plataformas compatíveis no guia Plataformas do AWS Elastic Beanstalk .

Para obter detalhes e melhores práticas sobre como atualizar seu ambiente, consulte Atualizar a versão de plataforma do ambiente Elastic Beanstalk.

Implantação de aplicativos.NET core (Linux) com o Elastic Beanstalk

> **Confira o.NET no AWS Developer Center**
>
> Você já passou por nosso Centro de desenvolvedores .Net? É nosso balcão único para tudo o que está ligado a o.NET AWS.
> Para obter mais informações, consulte o.NET no AWS Developer Center.

Este capítulo fornece instruções para configurar e implantar seu aplicativo web.NET core (Linux) no. AWS Elastic Beanstalk O Elastic Beanstalk facilita a implantação, o gerenciamento e a escalabilidade de seus aplicativos web.NET core (Linux) usando a Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk, ou o. AWS CLI APIs

Siga as etapas em QuickStart para o.NET Core no Linux para step-by-step obter instruções para criar e implantar um aplicativo web ASP.NET Core Hello World com a CLI do EB.

Tópicos

- QuickStart: Implante um aplicativo.NET Core no Linux no Elastic Beanstalk
- Configuração do ambiente de desenvolvimento .NET Core no Linux para o Elastic Beanstalk
- Usar o Elastic Beanstalk .NET Core na plataforma Linux
- O AWS Toolkit for Visual Studio - Trabalhando com o.NET Core no Elastic Beanstalk
- Migração da plataforma .NET no Windows Server para a plataforma .NET Core no Linux no Elastic Beanstalk

QuickStart: Implante um aplicativo.NET Core no Linux no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo.NET Core no Linux e sua implantação em um AWS Elastic Beanstalk ambiente.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação .NET Core no Linux
- Etapa 2: executar a aplicação localmente
- Etapa 3: implantar sua aplicação .NET Core no Linux com a EB CLI
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços necessários.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

.NET Core no Linux

Se não tiver o .NET SDK instalado na máquina local, você poderá instalá-lo seguindo o link Baixar .NET no site de Documentação do .NET.

Verifique sua instalação do .NET SDK executando o comando a seguir.

```
~$ dotnet --info
```

Etapa 1: criar uma aplicação .NET Core no Linux

Crie um diretório do projeto.

```
~$ mkdir eb-dotnetcore
~$ cd eb-dotnetcore
```

Em seguida, crie uma aplicação Hello World de exemplo executando os seguintes comandos.

```
~/eb-dotnetcore$ dotnet new web --name HelloElasticBeanstalk
~/eb-dotnetcore$ cd HelloElasticBeanstalk
```

Etapa 2: executar a aplicação localmente

Execute o seguinte comando para executar sua aplicação localmente.

```
~/eb-dotnetcore/HelloElasticBeasntalk$ dotnet run
```

A saída deve ser semelhante ao texto a seguir.

```
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**
>
> O comando `dotnet` seleciona uma porta aleatoriamente ao executar a aplicação localmente. Neste exemplo, a porta é 5052. Ao implantar a aplicação no ambiente Elastic Beanstalk, ela será executada na porta 5000.

Insira o endereço URL `http://localhost:port` no navegador. Para este exemplo específico, o comando é `http://localhost:5052`. O navegador deve exibir “Hello World!”.

Etapa 3: implantar sua aplicação .NET Core no Linux com a EB CLI

Execute os seguintes comandos para criar um ambiente Elastic Beanstalk para essa aplicação.

Para criar um ambiente e implantar sua aplicação .NET Core no Linux

1. Compile e publique sua aplicação em uma pasta para implantação no ambiente do Elastic Beanstalk que você está prestes a criar.

   ```
   ~$ cd eb-dotnetcore/HelloElasticBeanstalk
   ~/eb-dotnetcore/HelloElasticBeanstalk$ dotnet publish -o site
   ```

2. Navegue até o diretório no qual você acabou de publicar a aplicação.

   ```
   ~/eb-dotnetcore/HelloElasticBeanstalk$ cd site
   ```

3. Inicialize o repositório da EB CLI com o comando eb init:

   Observe os seguintes detalhes relacionados à versão da ramificação da plataforma que você especifica no comando:

   - *x.y.z*Substitua o comando a seguir pela versão mais recente da ramificação da plataforma .NET 6 em AL2 023.
   - Para localizar a versão mais recente da ramificação da plataforma, consulte .NET Core no Linux Plataformas compatíveis no Guia de plataformas do AWS Elastic Beanstalk .

- Um exemplo de um nome de pilha de soluções que inclui o número da versão é `64bit-amazon-linux-2023-v`**`3.1.1`**`-running-.net-6`. Neste exemplo, a versão da ramificação é 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.
```

Esse comando cria uma aplicação `dotnetcore-tutorial` e configura seu repositório local para criar ambientes com a versão da plataforma .NET Core on Linux especificada no comando.

4. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

```
~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 ]
```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

5. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

   com

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb create dotnet-tutorial
```

   O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb open
```

Parabéns! Você implantou uma aplicação .NET Core no Linux com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 5: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~eb-dotnetcore/HelloElasticBeanstalk/site$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.

- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

Depois que você implantar um ou dois aplicações de amostra e estiver pronto para começar a desenvolver e executar aplicações .NET Core no Linux localmente, consulte Configuração do ambiente de desenvolvimento .NET Core no Linux para o Elastic Beanstalk.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Configuração do ambiente de desenvolvimento .NET Core no Linux para o Elastic Beanstalk

Este tópico fornece instruções para configurar um ambiente de desenvolvimento .NET Core no Linux para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Seções

- Instalar o .NET Core SDK
- Instalar um IDE

- Instalando o AWS Toolkit for Visual Studio

Instalar o .NET Core SDK

É possível usar o .NET Core SDK para desenvolver aplicativos executados no Linux.

Consulte a página de downloads do .NET para fazer download e instalar o .NET Core SDK.

Instalar um IDE

Ambientes de desenvolvimento integrados (IDEs) fornecem uma variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você nunca usou um IDE para desenvolvimento .NET, experimente o Visual Studio Community para começar.

Visite a página Visual Studio Community para fazer download e instalar o Visual Studio Community.

Instalando o AWS Toolkit for Visual Studio

O AWS Toolkit for Visual Studioé um plug-in de código aberto para o Visual Studio IDE que torna mais fácil para os desenvolvedores desenvolver, depurar e implantar aplicativos.NET usando o. AWS Consulte a página inicial do Toolkit for Visual Studio para obter instruções de instalação.

Usar o Elastic Beanstalk .NET Core na plataforma Linux

Este tópico descreve como configurar, criar e executar suas aplicações .NET Core no Linux no Elastic Beanstalk.

AWS Elastic Beanstalk oferece suporte a várias ramificações de plataforma para diferentes versões do framework do.NET Core que são executadas no sistema operacional Linux. Consulte .NET Core no Linux em Plataformas do AWS Elastic Beanstalk para obter uma lista completa.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Considerações sobre o .NET Core na plataforma Linux

Servidor de proxy

A plataforma .NET Core no Linux do Elastic Beanstalk inclui um proxy reverso que encaminha solicitações para a aplicação. Por padrão, o Elastic Beanstalk usa NGINX como servidor de proxy. É

possível optar por não usar nenhum servidor de proxy e configurar o Kestrel como o servidor Web. O Kestrel é incluído por padrão em modelos de projeto ASP.NET Core.

Estrutura do aplicativo

É possível publicar aplicações dependentes do runtime que usam o runtime do .NET Core fornecido pelo Elastic Beanstalk. Você também pode publicar aplicativos independentes que incluem o runtime do .NET Core e as dependências do aplicativo no pacote de origem. Para saber mais, consulte the section called “Empacotar aplicativos”.

Configuração da plataforma

Para configurar os processos que são executados em instâncias do servidor em seu ambiente, inclua um Procfile opcional em seu pacote de origem. Um `Procfile` será necessário se você tiver mais de um aplicativo no pacote de origem.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos o Elastic Beanstalk executa para a aplicação.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Configurar o ambiente .NET Core no Linux

As configurações do.NET Core na plataforma Linux permitem que você ajuste o comportamento de suas instâncias da Amazon EC2 . Você pode editar a configuração da instância Amazon do ambiente do Elastic Beanstalk usando o console EC2 do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar o ambiente .NET Core no Linux usando o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log Options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Propriedades de ambiente

A seção Propriedades do ambiente permite que você especifique as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

No ambiente .NET Core no Linux em execução no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando `Environment.GetEnvironmentVariable("variable-name")`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código.

```
string endpoint = Environment.GetEnvironmentVariable("API_ENDPOINT");
```

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespace de configuração do .NET Core no Linux

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A plataforma .NET Core no Linux é compatível com opções no namespace a seguir, além das opções compatíveis com todos os ambientes do Elastic Beanstalk:

- `aws:elasticbeanstalk:environment:proxy`: escolha usar NGINX ou nenhum servidor de proxy. Os valores válidos são `nginx` ou `none`.

O arquivo de configuração demonstrativo a seguir mostra o uso das opções de configuração específicas do .NET Core no Linux.

Example .ebextensions/proxy-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Agrupamento de aplicações para o .NET Core na plataforma Linux Elastic Beanstalk

É possível executar aplicações .NET Core dependentes do runtime e independentes no AWS Elastic Beanstalk.

Uma aplicação dependente de runtime usa um runtime do .NET Core que o Elastic Beanstalk fornece para executar sua aplicação. O Elastic Beanstalk usa o arquivo `runtimeconfig.json` no pacote

de origem para determinar o runtime a ser usado para sua aplicação. O Elastic Beanstalk escolhe o runtime compatível mais recente disponível para a aplicação.

Um aplicativo independente inclui o runtime do .NET Core, o aplicativo e suas dependências. Para usar uma versão do runtime do .NET Core não incluída pelo Elastic Beanstalk em suas plataformas, forneça uma aplicação independente.

Exemplos

É possível compilar aplicativos independentes e dependentes do runtime com o comando `dotnet publish`. Para saber mais sobre a publicação de aplicativos .NET Core, consulte Visão geral da publicação de aplicativos .NET Core na documentação do .NET Core.

A estrutura de arquivos demonstrativa a seguir define uma única aplicação que usa um runtime do .NET Core fornecido pelo Elastic Beanstalk.

```
### appsettings.Development.json
### appsettings.json
### dotnetcoreapp.deps.json
### dotnetcoreapp.dll
### dotnetcoreapp.pdb
### dotnetcoreapp.runtimeconfig.json
### web.config
### Procfile
### .ebextensions
### .platform
```

É possível incluir vários aplicativos no pacote de origem. O exemplo a seguir define dois aplicativos a serem executados no mesmo servidor Web. Para executar vários aplicativos, é necessário incluir um Procfile no pacote de origem. Para ver um exemplo completo de aplicativo, consulte dotnet-core-linux-multiple-apps.zip.

```
### 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
```

Usar um Procfile para configurar seu ambiente .NET Core no Linux Elastic Beanstalk

Para executar vários aplicações no mesmo servidor Web, é necessário incluir um `Procfile` no pacote de origem que informa ao Elastic Beanstalk quais aplicações devem ser executadas.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos são executados pelo Elastic Beanstalk para o aplicativo e quais argumentos são recebidos por esses processos.

O exemplo a seguir usa um `Procfile` para especificar duas aplicações a serem executadas pelo Elastic Beanstalk no mesmo servidor Web.

Example Procfile

```
web: dotnet ./dotnet-core-app1/dotnetcoreapp1.dll
web2: dotnet ./dotnet-core-app2/dotnetcoreapp2.dll
```

Para obter detalhes sobre como gravar e usar um `Procfile`, consulte Buildfile e Procfile.

Configurar o servidor de proxy

AWS Elastic Beanstalk usa o NGINX como proxy reverso para retransmitir solicitações para seu aplicativo. O Elastic Beanstalk oferece uma configuração NGINX padrão que você pode estender ou substituir completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy NGINX para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a propriedade do ambiente `PORT` para a porta de escuta do seu aplicativo principal.

> **Note**
>
> A porta que seu aplicativo escuta não afeta a porta na qual o servidor NGINX escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as AL2 plataformas AL2 023/ suportam um recurso de configuração de proxy uniforme. Para obter mais informações sobre como configurar o servidor proxy nas versões da sua plataforma executando AL2 023/AL2, consulte. Configuração de proxy reverso

O arquivo de configuração de exemplo a seguir estende a configuração de NGINX do ambiente. A configuração direciona solicitações para `/api` para um segundo aplicativo Web que escuta na porta 5200 no servidor Web. Por padrão, o Elastic Beanstalk encaminha solicitações para uma única aplicação que escuta na 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;
}
```

O AWS Toolkit for Visual Studio - Trabalhando com o.NET Core no Elastic Beanstalk

Este tópico mostra como você pode executar as seguintes tarefas usando o AWS Toolkit for Visual Studio:

- Crie uma aplicação Web ASP.NET Core usando um modelo do Visual Studio.
- Crie um ambiente Amazon Linux do Elastic Beanstalk.
- Implante a aplicação Web ASP.NET Core no novo ambiente Amazon Linux.

Este tópico também explora como você pode usar o AWS Toolkit for Visual Studio para gerenciar seus ambientes de aplicativos do Elastic Beanstalk e monitorar a integridade do seu aplicativo.

AWS Toolkit for Visual Studio É um plug-in para o IDE do Visual Studio. Com o toolkit, é possível implantar e gerenciar aplicações no Elastic Beanstalk enquanto trabalha no ambiente do Visual Studio.

Seções

- Pré-requisitos
- Criar um projeto de aplicação
- Criar um ambiente do Elastic Beanstalk e implantar a aplicação
- Encerrar um ambiente
- Gerenciar os ambientes de aplicações do Elastic Beanstalk
- Monitorar a integridade do aplicativo

Pré-requisitos

Antes de iniciar este tutorial, é necessário instalar o AWS Toolkit for Visual Studio. Para obter instruções, consulte Configurar o AWS Toolkit for Visual Studio.

Se você nunca usou o kit de ferramentas antes, a primeira coisa que precisará fazer depois de instalar o kit de ferramentas é registrar suas AWS credenciais no kit de ferramentas. Para obter mais informações sobre isso, consulte Fornecer credenciais da AWS.

Criar um projeto de aplicação

Se você não tiver um projeto de aplicação .NET Core no Visual Studio, poderá facilmente criar um usando um dos modelos de projeto do Visual Studio.

Como criar um novo projeto de aplicação Web ASP.NET Core

1. No Visual Studio, no menu File (Arquivo), escolha New (Novo) e, depois, escolha Project (Projeto).
2. Na caixa de diálogo Create a new project (Criar um projeto), selecione C#, Linux e Cloud (Nuvem).
3. Na lista de modelos de projeto exibida, selecione ASP.NET Core Web Application (Aplicação Web ASP.NET Core) e Next (Próximo).

   > **Note**
   >
   > Se a ASP.NET Core Web Application (Aplicação Web ASP.NET Core) não estiver listada nos modelos de projeto, você poderá instalá-la no Visual Studio.
   >
   > 1. Role até a parte inferior da lista de modelos e selecione o link Install more tools and features (Instalar mais ferramentas e recursos) que está localizado na lista de modelos.
   > 2. Se você for solicitado a permitir que a aplicação do Visual Studio, faça alterações no dispositivo, selecione Yes (Sim).
   > 3. Selecione a guia Workloads (Cargas de trabalho) e a opção ASP.NET and web development (ASP.NET e desenvolvimento Web).
   > 4. Selecione o botão Modify (Modificar). O Visual Studio Installer (Instalador do Visual Studio) instala o modelo de projeto.
   > 5. Depois que o instalador for concluído, saia do painel para retornar ao ponto em que você parou no Visual Studio.

4. Na caixa de diálogo Configure your new project (Configurar o novo projeto), insira um Project name (Nome do projeto). O Solution name (Nome da solução) é padrão para o nome do projeto. Depois, selecione Create (Criar).
5. Na caixa de diálogo Create a new ASP.NET Core web application (Criar uma aplicação Web ASP.NET Core), selecione .NET Core e ASP.NET Core 3.1. Na lista de tipos de aplicação exibida, selecione Web Application (Aplicação Web) e selecione o botão Create (Criar).

O Visual Studio exibe a caixa de diálogo Creating Project (Criando projeto) ao criar a aplicação. Depois que o Visual Studio concluir a geração da aplicação, será exibido um painel com o nome da aplicação.

Criar um ambiente do Elastic Beanstalk e implantar a aplicação

Esta seção descreve como criar um ambiente do Elastic Beanstalk para a aplicação e implantá-la nesse ambiente.

Como criar um ambiente e implantar a aplicação

1. No Visual Studio, selecione View (Visualizar) e, depois, Solution Explorer.
2. Em Solution Explorer, abra o menu de contexto (clique com o botão direito do mouse) da aplicação e selecione Publish to AWS Elastic Beanstalk(Publicar no).
3. No assistente Publicar na AWS Elastic Beanstalk, insira as informações da sua conta.

    a. Em Account profile to use (Perfil de conta a ser usado), selecione a conta default (padrão) ou selecione o ícone Add another account (Adicionar outra conta) para inserir as informações da novas conta.

    b. Em Region (Região), selecione a região em que você deseja implantar a aplicação. Para obter informações sobre AWS regiões disponíveis, consulte AWS Elastic Beanstalk Endpoints and Quotas no. Referência geral da AWS Se você selecionar uma região não compatível com o Elastic Beanstalk, a opção para implantar no Elastic Beanstalk não estará disponível.

    c. Selecione Create a new application environment (Criar um ambiente de aplicação) e Next (Próximo).
4. Na caixa de diálogo Application Environment (Ambiente de aplicação), insira os detalhes do novo ambiente de aplicação.
5. Na próxima caixa de diálogo de AWSopções, defina EC2 as opções da Amazon e outras opções AWS relacionadas para seu aplicativo implantado.

    a. Para o tipo de contêiner, selecione Amazon Linux 2 v de 64 bits ***<n.n.n>*** executando o.NET Core.

> **Note**
>
> Recomendamos que você selecione a versão atual da plataforma do Linux. Essa versão contém as correções de erros e segurança mais recentes que estão incluídas na imagem de máquina da Amazon (AMI) mais recente.

b. Em Instance Type (Tipo de instância), selecione t2.micro. (Escolher um tipo de microinstância minimizará o custo associado à execução da instância.)

c. Para Par de chaves, selecione Criar novo par de chaves. Insira um nome para o novo par de chaves e selecione OK. (Neste exemplo, usamos **`myuseastkeypair`**.) Um key pair permite o acesso por desktop remoto às suas instâncias da Amazon EC2 . Para obter mais informações sobre os pares de EC2 chaves da Amazon, consulte Usando credenciais no Guia do usuário do Amazon Elastic Compute Cloud.

d. Para uma aplicação simples e de baixo tráfego, selecione Single instance environment (Ambiente de instância única). Para ter mais informações, consulte Tipos de ambiente

e. Escolha Próximo.

Para obter mais informações sobre as AWS opções que não são usadas neste exemplo, considere as seguintes páginas:

- Em Use custom AMI (Usar AMI personalizada), consulte Usar uma imagem de máquina Amazon personalizada (AMI) no ambiente Elastic Beanstalk.
- Se você não selecionar Single instance environment (Ambiente de instância única), será necessário escolher um Load balance type (Tipo de balanceamento de carga). Consulte Balanceador de carga do ambiente do Elastic Beanstalk para obter mais informações.
- O Elastic Beanstalk usará a configuração padrão da Amazon VPC (Amazon Virtual Private Cloud) se você não escolheu Use non-default VPC (Usar VPC não padrão). Para obter mais informações, consulte Usar o Elastic Beanstalk com Amazon VPC.
- Escolher a opção Enable Rolling Deployments (Habilitar implantações contínuas) divide uma implantação em lotes para evitar possíveis períodos de inatividade durante as implantações. Para obter mais informações, consulte Implantar aplicações em ambientes do Elastic Beanstalk.
- Escolher a opção Relational Database Access (Acesso ao banco de dados relacional) conecta o ambiente do Elastic Beanstalk a um banco de dados do Amazon RDS criado anteriormente

com grupos de segurança de banco de dados do Amazon RDS. Para obter mais informações, consulte Controlar o acesso com grupos de segurança no Guia do usuário do Amazon RDS.

6. Selecione Next (Próximo) na caixa de diálogo Permissions (Permissões).
7. Selecione Next (Próximo) na caixa de diálogo Applications Options (Opções de aplicações).
8. Revise as opções de implantação. Depois de verificar se as configurações estão corretas, selecione Deploy (Implantar).

A aplicação Web ASP.NET Core é exportada como um arquivo de implantação da Web. Esse arquivo é, então, carregado para o Amazon S3 e registrado como uma nova versão da aplicação com o Elastic Beanstalk. O recurso de implantação do Elastic Beanstalk monitora o ambiente existente até ele se tornar disponível com o código recém-implantado. O Status do ambiente será exibido na guia Env:<nome do ambiente>. Depois que o status for atualizado para Environment is healthy (O ambiente está íntegro), você poderá selecionar o endereço URL para executar a aplicação Web.

Encerrar um ambiente

Para evitar cobranças por AWS recursos não utilizados, você pode usar o AWS Toolkit for Visual Studio para encerrar um ambiente em execução.

> **Note**
>
> Você sempre pode iniciar um novo ambiente usando a mesma versão mais tarde.

Encerrar um ambiente

1. Expanda o nó Elastic Beanstalk e o nó da aplicação. No AWS Explorer, abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione Terminate Environment (Terminar ambiente).
2. Quando solicitado, selecione Yes (Sim) para confirmar que você deseja encerrar o ambiente. São necessários alguns minutos para que o Elastic Beanstalk AWS encerre os recursos em execução no ambiente.

O Status do ambiente na guia Env:<nome do ambiente> é alterado para Terminating (Encerrando) e, por fim, para Terminated (Encerrado).

> **Note**
>
> Quando você encerra o ambiente, o CNAME associado ao ambiente encerrado torna-se disponível para qualquer pessoa utilizar.

Gerenciar os ambientes de aplicações do Elastic Beanstalk

Esta seção descreve as configurações de serviço específicas que você pode editar no AWS Toolkit for Visual Studio como parte da configuração de ambientes de aplicações. Com o AWS Toolkit for Visual Studio e AWS o Management Console, você pode alterar o provisionamento e a configuração dos recursos usados pelos AWS ambientes de aplicativos. Para obter informações sobre como gerenciar seus ambientes de aplicativos usando o AWS Management Console, consulteCriação de ambientes no Elastic Beanstalk.

Alterar os ajustes de configuração do ambiente

Quando você implanta seu aplicativo, o Elastic Beanstalk configura AWS vários serviços de computação em nuvem conectados. Você pode controlar como esses serviços individuais são configurados usando o AWS Toolkit for Visual Studio.

Para editar as configurações de ambiente de um aplicativo

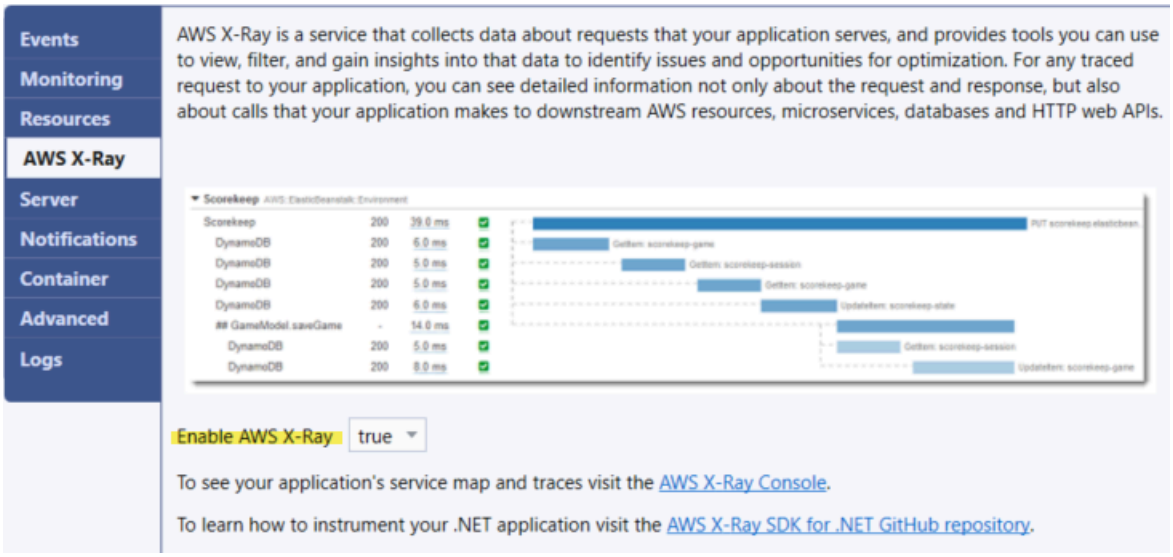
1. No Visual Studio, no menu Arquivo, escolha AWS Explorer.
2. Expanda o nó do Elastic Beanstalk e o nó da aplicação. Abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione View Status (Visualizar status).

   Agora, você pode definir as seguintes configurações:

   - AWS X-Ray
   - Servidor
   - Load balancer (aplica-se apenas a ambientes de várias instâncias)
   - Auto Scaling (aplica-se apenas a ambientes de várias instâncias)
   - Notificações
   - Contêiner
   - Opções de configuração avançadas

Configurando o AWS X-Ray usando o AWS kit de ferramentas para Visual Studio

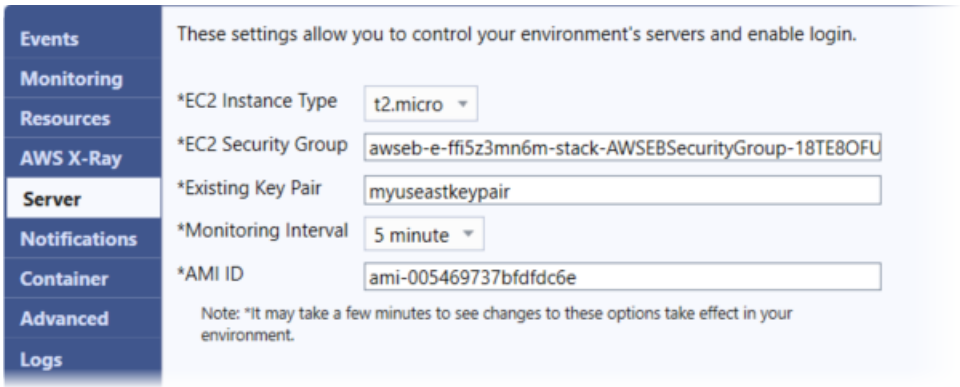
AWS O X-Ray fornece recursos de rastreamento de solicitações, coleta de exceções e criação de perfil. Com o painel AWS X-Ray, você pode ativar ou desativar o X-Ray para seu aplicativo. Para obter mais informações sobre o X-Ray, consulte o Guia do desenvolvedor do AWS X-Ray.



Configurando EC2 instâncias usando o AWS kit de ferramentas para Visual Studio

Você pode usar o Amazon Elastic Compute Cloud (Amazon EC2) para iniciar e gerenciar instâncias de servidor nos datacenters da Amazon. Você pode usar as instâncias EC2 do servidor Amazon a qualquer momento, pelo tempo que precisar e para qualquer finalidade legal. As instâncias estão disponíveis em diferentes tamanhos e configurações. Para obter mais informações, consulte Amazon EC2.

Você pode editar a configuração da sua EC2 instância Amazon com a guia Servidor dentro da guia Ambiente do aplicativo no AWS Toolkit for Visual Studio.



Tipos de EC2 instância da Amazon

Instance type (Tipo de instância) exibe os tipos de instância disponíveis para a aplicação do Elastic Beanstalk. Altere o tipo de instância para selecionar um servidor com as características (inclusive o tamanho da memória e a capacidade de CPU) mais adequadas à aplicação. Por exemplo, os aplicativos com operações muito intensas e demoradas podem exigir mais CPU ou memória.

Para obter mais informações sobre os tipos de EC2 instância da Amazon disponíveis para seu aplicativo Elastic Beanstalk, consulte Tipos de instância no Guia do usuário do Amazon Elastic Compute Cloud.

Grupos EC2 de segurança da Amazon

Você pode controlar o acesso ao seu aplicativo Elastic Beanstalk usando EC2 um grupo de segurança da Amazon. Um grupo de segurança define regras de firewall para suas instâncias. Essas regras especificam qual tráfego de rede de entrada deve ser fornecido para a instância. Todo o outro tráfego de entrada é descartado. Você pode modificar as regras para um grupo a qualquer momento. As novas regras são aplicadas automaticamente para todas as instâncias em execução e as iniciadas no futuro.

Você pode especificar quais grupos de EC2 segurança da Amazon controlam o acesso ao seu aplicativo Elastic Beanstalk. Para fazer isso, insira os nomes de grupos de EC2 segurança específicos da Amazon (separando vários grupos de segurança com vírgulas) na caixa de texto Grupos de EC2 segurança. Você pode fazer isso usando o AWS Management Console ou o AWS Toolkit for Visual Studio.

Para criar um grupo de segurança usando o AWS kit de ferramentas do Visual Studio

1. No Visual Studio, no AWS Explorer, expanda o EC2 nó da Amazon e selecione Security Groups.
2. Selecione Create Security Group (Criar grupo de segurança) e insira um nome e uma descrição para o grupo de segurança.
3. Selecione OK.

Para obter mais informações sobre os grupos EC2 de segurança da Amazon, consulte Como usar grupos de segurança no Guia do usuário do Amazon Elastic Compute Cloud.

Pares de EC2 chaves da Amazon

Você pode fazer login com segurança nas EC2 instâncias da Amazon provisionadas para seu aplicativo Elastic Beanstalk com um par de chaves da Amazon. EC2

> **Important**
>
> Você deve criar um par de EC2 chaves da Amazon e configurar suas EC2 instâncias da Amazon provisionadas pelo Elastic Beanstalk para poder acessar essas instâncias. Você pode criar seu par de chaves usando o AWS assistente Publish to dentro do AWS Toolkit for Visual Studio ao implantar seu aplicativo no Elastic Beanstalk. Se você quiser criar pares de chaves adicionais usando o Toolkit, siga as etapas descritas aqui. Como alternativa, você pode configurar seus pares de EC2 chaves da Amazon usando o AWS Management Console. Para obter instruções sobre como criar um par de chaves para a Amazon EC2, consulte o Guia de introdução do Amazon Elastic Compute Cloud.

A caixa de texto Existing Key Pair permite que você especifique o nome de um par de EC2 chaves da Amazon que você pode usar para fazer login com segurança nas EC2 instâncias da Amazon que estão executando seu aplicativo Elastic Beanstalk.

Para especificar o nome de um par de EC2 chaves da Amazon

1. Expanda o EC2 nó da Amazon e selecione pares de chaves.
2. Selecione Create Key Pair (Criar par de chaves) e insira o nome do par de chaves.
3. Selecione OK.

Para obter mais informações sobre os pares de EC2 chaves da Amazon, acesse Como usar EC2 as credenciais da Amazon no Guia do usuário do Amazon Elastic Compute Cloud. Para obter mais informações sobre como se conectar às EC2 instâncias da Amazon, consulte

Intervalo de monitoramento

Por padrão, somente as CloudWatch métricas básicas da Amazon estão habilitadas. Elas retornam dados em intervalos de cinco minutos. Você pode habilitar CloudWatch métricas mais granulares de um minuto selecionando 1 minuto para o Intervalo de monitoramento na seção Servidor da guia Configuração do seu ambiente no. AWS Toolkit for Eclipse

> **Note**
>
> As taxas CloudWatch de serviço da Amazon podem ser aplicadas a métricas de intervalo de um minuto. Consulte a Amazon CloudWatch para obter mais informações.

ID da AMI personalizado

Você pode substituir a AMI padrão usada para suas EC2 instâncias da Amazon por sua própria AMI personalizada inserindo o identificador da sua AMI personalizada na caixa ID de AMI personalizada na seção Servidor da guia Configuração do seu ambiente no AWS Toolkit for Eclipse.

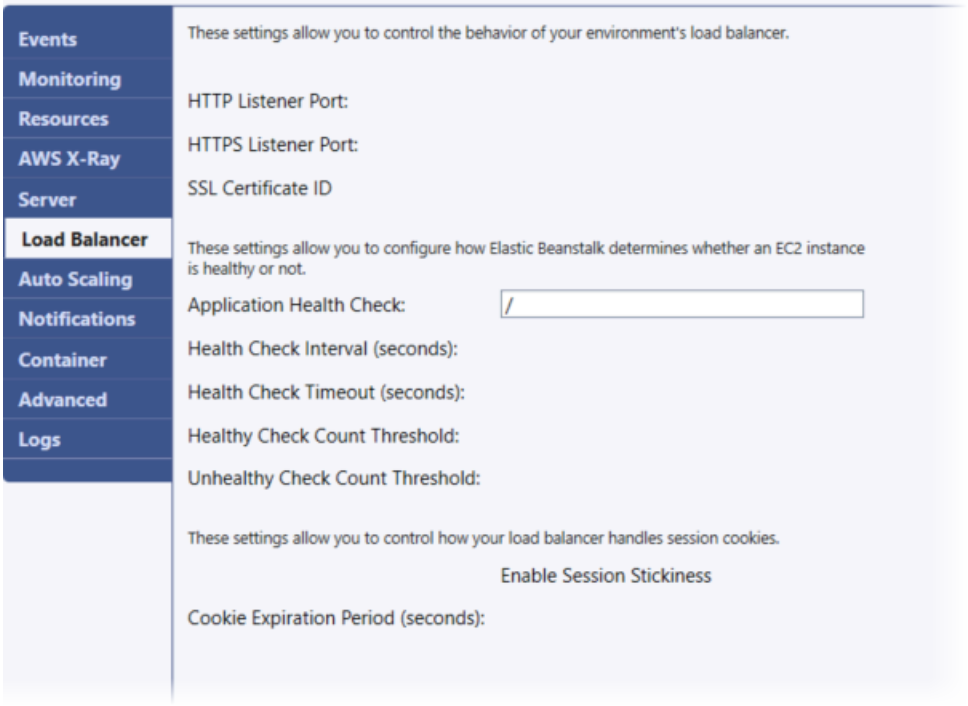
> **Important**
>
> O uso da AMI própria é uma tarefa avançada que deve ser feita com cuidado. Se você precisar de uma AMI personalizada, recomendamos começar com a AMI do Elastic Beanstalk padrão e depois modificá-la. Para serem consideradas saudáveis, o Elastic Beanstalk EC2 espera que as instâncias da Amazon atendam a um conjunto de requisitos, incluindo ter um gerenciador de host em execução. Se esses requisitos não forem atendidos, talvez o ambiente não funcione corretamente.

Configurando o Elastic Load Balancing usando AWS o kit de ferramentas para Visual Studio

O Elastic Load Balancing é um serviço web da Amazon que ajuda você a aprimorar a disponibilidade e a escalabilidade de seu aplicativo. Esse serviço facilita a distribuição de cargas de aplicativos entre duas ou mais EC2 instâncias da Amazon. O Elastic Load Balancing melhora a disponibilidade ao fornecer mais redundância e acomoda o crescimento de tráfego da aplicação.

Com o Elastic Load Balancing, é possível distribuir e equilibrar automaticamente o tráfego de entrada de aplicações entre todas as instâncias em execução. Também é possível adicionar facilmente novas instâncias quando for necessário aumentar a capacidade da aplicação.

O Elastic Beanstalk provisiona automaticamente o Elastic Load Balancing ao implantar uma aplicação. Você pode editar a configuração da instância Amazon do ambiente EC2 do Elastic Beanstalk com a guia Load Balancer dentro da guia do ambiente do aplicativo no Toolkit AWS for Visual Studio.



As seções a seguir descrevem os parâmetros do Elastic Load Balancing que você pode configurar para o seu aplicativo.

Portas

O balanceador de carga provisionado para lidar com solicitações para seu aplicativo Elastic Beanstalk envia solicitações para as instâncias da Amazon que estão executando seu aplicativo. EC2 O balanceador de carga provisionado pode escutar solicitações nas portas HTTP e HTTPS e rotear solicitações para as EC2 instâncias da Amazon em seu aplicativo. AWS Elastic Beanstalk Por padrão, o load balancer lida com solicitações na porta HTTP. Para realizar esse trabalho, pelo menos uma das portas (HTTP ou HTTPS) deve estar ativada.



> **Important**
>
> A porta especificada não deve estar bloqueada. Caso contrário, você não conseguirá se conectar à aplicação do Elastic Beanstalk.

Controle da porta HTTP

Para desativar a porta HTTP, selecione OFF em HTTP Listener Port. Para ativar a porta HTTP, você deve selecionar uma porta HTTP (por exemplo, a 80) na lista.

> **Note**
>
> Para acessar o ambiente usando uma porta diferente da porta 80 padrão, como a porta 8080, adicione uma escuta ao load balancer existente e configure a nova escuta nessa porta. Por exemplo, usando o AWS CLI for Classic load balancers, digite o comando a seguir, *LOAD_BALANCER_NAME* substituindo-o pelo nome do seu load balancer para o Elastic Beanstalk.
>
> ```
> aws elb create-load-balancer-listeners --load-balancer-name LOAD_BALANCER_NAME
>  --listeners "Protocol=HTTP, LoadBalancerPort=8080, InstanceProtocol=HTTP,
>  InstancePort=80"
> ```
>
> Por exemplo, usando o AWS CLI for Application Load Balancers, digite o comando a seguir, *LOAD_BALANCER_ARN* substituindo-o pelo ARN do seu load balancer para o Elastic Beanstalk.
>
> ```
> aws elbv2 create-listener --load-balancer-arn LOAD_BALANCER_ARN --protocol HTTP
>  --port 8080
> ```
>
> Se você quiser que o Elastic Beanstalk monitore o ambiente, não remova a escuta na porta 80.

Controle da porta HTTPS

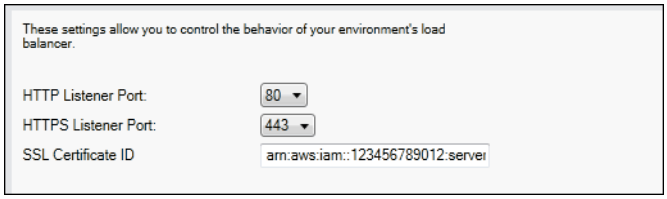
O Elastic Load Balancing oferece suporte ao protocolo HTTPS/TLS para permitir a criptografia do tráfego para conexões de clientes com o load balancer. As conexões do balanceador de carga com as EC2 instâncias usam criptografia de texto simples. Por padrão, a porta HTTPS é desativada.

Para ativar a porta HTTPS

1. Crie um novo certificado usando AWS Certificate Manager (ACM) ou carregue um certificado e uma chave para AWS Identity and Access Management (IAM). Para obter mais informações sobre como solicitar um certificado do ACM, consulte Solicitação de um certificado no Manual

do usuário do AWS Certificate Manager . Para obter mais informações sobre a importação de certificados de terceiros para o ACM, consulte Importação de certificados no Manual do usuário do AWS Certificate Manager . Se o ACM não estiver disponível na sua região, use AWS Identity and Access Management (IAM) para carregar um certificado de terceiros. Os serviços ACM e IAM armazenarão o certificado e fornecerão um nome de recurso da Amazon (ARN) para o certificado SSL. Para obter mais informações sobre a criação e o upload de certificados para o IAM, consulte Trabalhar com certificados de servidor no Trabalhar com certificados de servidor no Guia do usuário do IAM.

2. Especifique a porta HTTPS selecionando uma porta em HTTPS Listener Port.



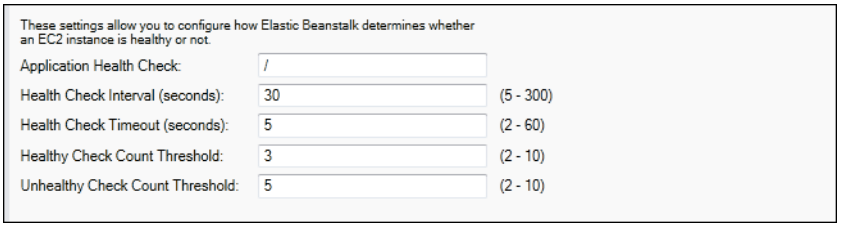
3. Para SSL Certificate ID (ID de certificado SSL), insira o Nome de recurso da Amazon (ARN) do seu certificado SSL. Por exemplo, o **`arn:aws:iam::123456789012:server-certificate/abc/certs/build`** ou o **`arn:aws:acm:us-east-2:123456789012:certificate/12345678-12ab-34cd-56ef-12345678`**. Use o certificado SSL que você criou ou do qual fez upload na etapa 1.

Para desativar a porta HTTPS, selecione OFF em HTTPS Listener Port.

Verificações de integridade

A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Por padrão, o Elastic Beanstalk usa TCP: 80 para contêineres não legados e HTTP: 80 para contêineres legados. É possível substituir o URL padrão para corresponder a um recurso existente na aplicação (por exemplo, `/myapp/default.aspx`), inserindo-o na caixa Application Health Check URL (URL de verificação de integridade da aplicação). Se você substituir o URL padrão, o Elastic Beanstalk usará HTTP para consultar o recurso. Para verificar se você está usando um tipo de contêiner legado, consulte the section called “Por que algumas versões de plataforma são marcadas como legadas?”

Você pode controlar as configurações da verificação de saúde usando a seção EC2 Instance Health Check do painel Load Balancing.



A definição de verificação de integridade inclui um URL a ser consultado para saber a integridade da instância. Substitua o URL padrão para corresponder a um recurso existente na aplicação (por exemplo, `/myapp/index.jsp`), inserindo-o na caixa Application Health Check URL (URL de verificação de integridade da aplicação).

A lista a seguir descreve os parâmetros de verificação de integridade que você pode definir para o aplicativo.

- Em Health Check Interval (segundos), insira o número de segundos que o Elastic Load Balancing espera entre as verificações de saúde das instâncias da Amazon do seu aplicativo. EC2
- Em Health Check Timeout (seconds), especifique o número de segundos que o Elastic Load Balancing aguarda por uma resposta antes que ele considere que a instância não está respondendo.
- Em Healthy Check Count Threshold e Unhealthy Check Count Threshold, especifique o número de testes consecutivos de URL bem-sucedidos ou sem êxito antes que o Elastic Load Balancing altere o status de integridade da instância. Por exemplo, especificar **5** em Unhealthy Check Count Threshold (Limite de contagem de verificações de falta de integridade) significa que o URL deve retornar uma mensagem de erro ou de tempo esgotado cinco vezes consecutivas antes que o Elastic Load Balancing considere que a verificação de integridade falhou.

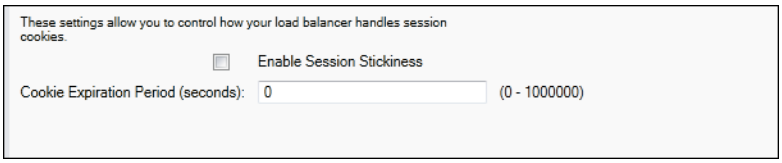
Sessões

Por padrão, um load balancer roteia cada solicitação de forma independente para a instância de servidor com a menor carga. Por comparação, uma sticky session vincula a sessão do usuário a uma determinada instância do servidor, para que todas as solicitações desse usuário durante a sessão sejam enviadas para a mesma instância do servidor.

O Elastic Beanstalk usa cookies HTTP gerados pelo balanceador de carga quando as sticky sessions são permitidas em uma aplicação. O load balancer usa um cookie especial gerado pelo load balancer para rastrear cada solicitação na instância do aplicativo. Quando o load balancer receber uma solicitação, ele primeiro verificará se esse cookie está presente na solicitação. Se estiver presente, a solicitação será enviada para a instância da aplicação especificada no cookie. Se não

houver um cookie, o load balancer selecionará uma instância de aplicativo com base no algoritmo de balanceamento de carga existente. Um cookie será inserido na resposta para vincular as solicitações subsequentes do mesmo usuário para essa instância do aplicativo. A configuração da política define a expiração do cookie, que estabelece a validade de cada cookie.

É possível usar a seção Sessions (Sessões) na guia Load Balancer para especificar se deseja que o load balancer da aplicação permita a perdurabilidade da sessão.



Para obter mais informações sobre Elastic Load Balancing, consulte o Guia do desenvolvedor de Elastic Load Balancing.

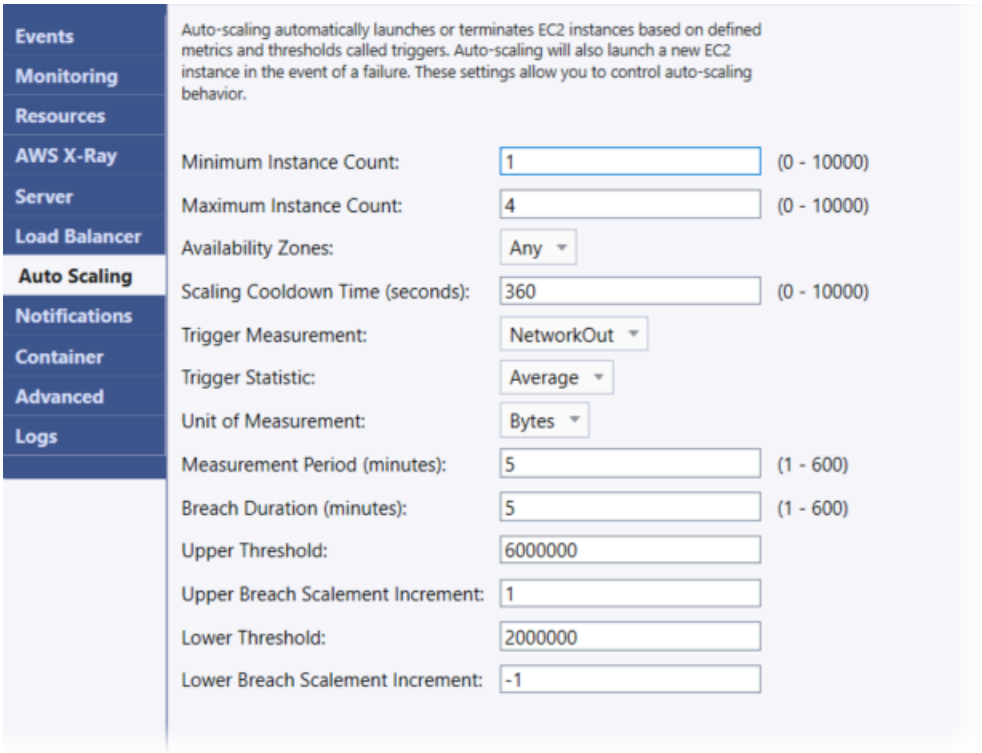
Configurando o Auto Scaling usando o kit de ferramentas AWS para Visual Studio

O Amazon EC2 Auto Scaling é um serviço web da Amazon projetado para iniciar ou encerrar automaticamente instâncias da EC2 Amazon com base em acionadores definidos pelo usuário. É possível configurar grupos de Auto Scaling e associar triggers a esses grupos para escalar automaticamente recursos de computação com base em métricas como uso de largura de banda ou utilização da CPU. O Amazon EC2 Auto Scaling trabalha com CloudWatch a Amazon para recuperar métricas para as instâncias do servidor que executam seu aplicativo.

O Amazon EC2 Auto Scaling permite que você pegue um grupo de EC2 instâncias da Amazon e defina vários parâmetros para que esse grupo aumente ou diminua automaticamente em número. O Amazon EC2 Auto Scaling pode adicionar ou remover EC2 instâncias da Amazon desse grupo para ajudá-lo a lidar perfeitamente com as mudanças de tráfego em seu aplicativo.

O Amazon EC2 Auto Scaling também monitora a integridade de cada EC2 instância da Amazon que ele executa. Se alguma instância for encerrada inesperadamente, o Amazon EC2 Auto Scaling detecta o encerramento e executa uma instância substituta. Esse recurso permite que você mantenha automaticamente um número fixo e desejado de EC2 instâncias da Amazon.

O Elastic Beanstalk provisiona o EC2 Amazon Auto Scaling para sua aplicação. Você pode editar a configuração da instância Amazon do ambiente EC2 do Elastic Beanstalk com a guia Auto Scaling dentro da guia do ambiente do aplicativo no Toolkit for AWS Visual Studio.

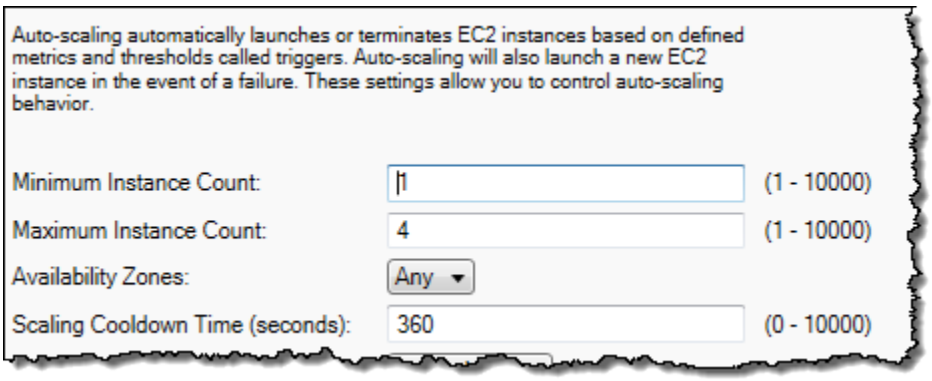


A seção a seguir discute como configurar os parâmetros de Auto Scaling para seu aplicativo.

Iniciar a configuração

Você pode editar a configuração de lançamento para controlar como seu aplicativo Elastic Beanstalk provisiona os recursos do EC2 Amazon Auto Scaling.

As caixas Minimum Instance Count (Contagem de instância mínima) e Maximum Instance Count (Contagem de instância máxima) permitem que você especifique os tamanhos mínimo e máximo do grupo de Auto Scaling que a aplicação do Elastic Beanstalk usa.



> **Note**
>
> Para manter um número fixo de EC2 instâncias da Amazon, defina a Contagem mínima de instâncias e a Contagem máxima de instâncias com o mesmo valor.

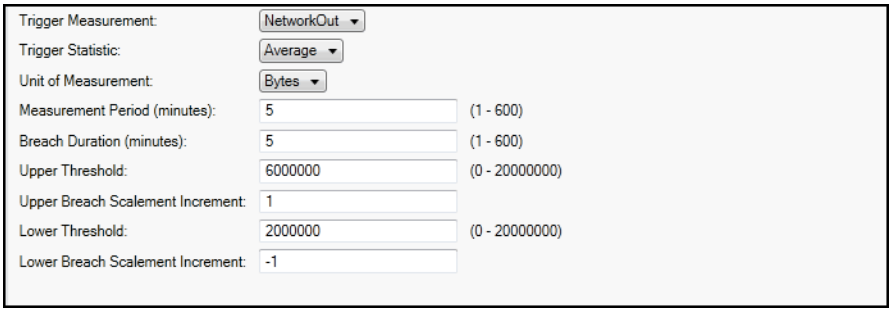
A caixa Zonas de disponibilidade permite que você especifique o número de zonas de disponibilidade nas quais você deseja que suas EC2 instâncias da Amazon estejam. É importante definir esse número se quiser criar aplicativos tolerantes a falhas. Se uma zona de disponibilidade ficar inativa, as instâncias continuarão sendo executadas em nas outras zonas de disponibilidade.

> **Note**
>
> No momento, não é possível especificar a zona de disponibilidade na qual sua instância estará.

Acionadores

Um gatilho é um mecanismo do Amazon EC2 Auto Scaling que você configura para informar ao sistema quando você deseja aumentar (escalar) ou diminuir (escalar) o número de instâncias. Você pode configurar gatilhos para serem acionados em qualquer métrica publicada na Amazon CloudWatch (por exemplo, utilização da CPU) e determinar se as condições especificadas foram atendidas. Quando o limite superior ou inferior das condições que você especificou para a métrica forem violados para o período especificado, o trigger iniciará um processo de longa duração chamado de ação de escalabilidade.

Você pode definir um gatilho de escalabilidade para seu aplicativo Elastic Beanstalk usando o Toolkit AWS for Visual Studio.



Os gatilhos do Amazon EC2 Auto Scaling funcionam monitorando uma CloudWatch métrica específica da Amazon de uma instância específica. As métricas incluem utilização de CPU, tráfego

de rede e a atividade do disco. Use a configuração Trigger Measurement para selecionar uma métrica para o trigger.

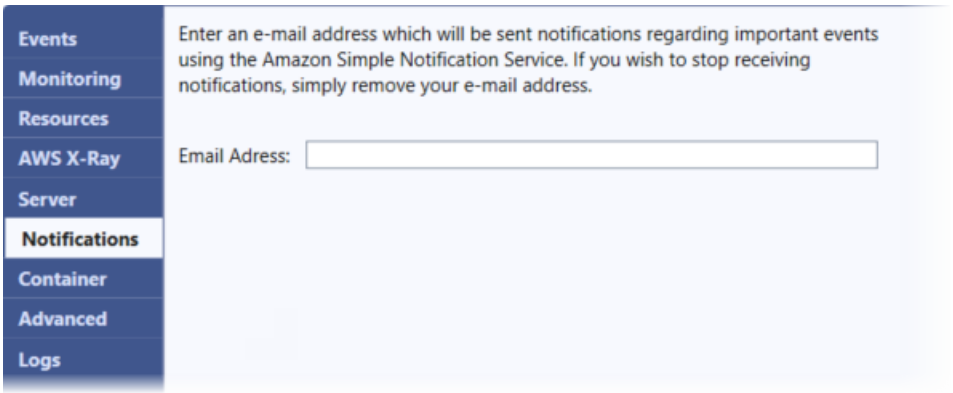
A lista a seguir descreve os parâmetros de acionamento que você pode configurar usando o AWS Management Console.

- Você pode especificar quais estatísticas o trigger deve usar. Você pode selecionar Minimum, Maximum, Sum ou Average em Trigger Statistic.
- Em Unit of Measurement, especifique a unidade de medida do trigger.
- O valor na caixa Período de medição especifica com que frequência a Amazon CloudWatch mede as métricas do seu gatilho. A Breach Duration (Duração da ruptura) é a quantidade de tempo que uma métrica pode durar além do limite definido (conforme especificado em Upper Threshold (Limite superior) e Lower Threshold (Limite inferior)) antes que o trigger seja acionado.
- Para incremento superior da escala de violação e incremento inferior da escala de violação, especifique quantas EC2 instâncias da Amazon adicionar ou remover ao realizar uma atividade de escalabilidade.

Para obter mais informações sobre o Amazon EC2 Auto Scaling, consulte a seção Amazon Auto EC2 Scaling na documentação do Amazon Elastic Compute Cloud.

Configurando notificações usando o AWS kit de ferramentas para Visual Studio

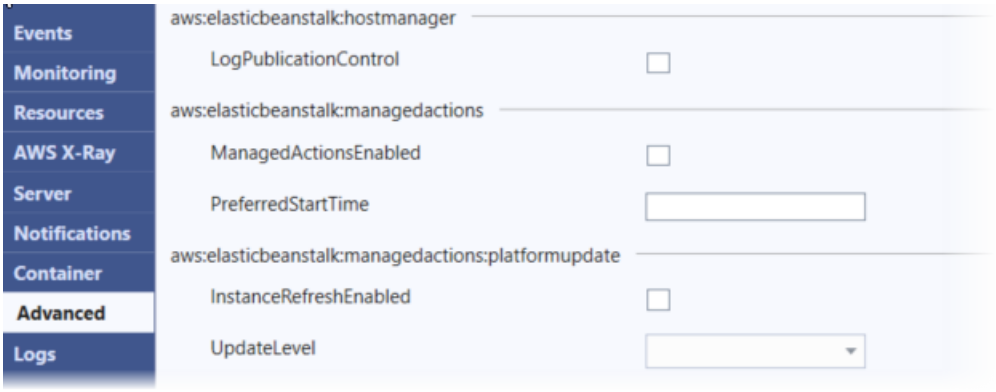
O Elastic Beanstalk usa o Amazon Simple Notification Service (Amazon SNS) para notificar você de eventos importantes que afetam sua aplicação. Para permitir notificações do Amazon SNS, insira seu endereço de e-mail na caixa Email Address (Endereço de e-mail). Para desativar essas notificações, remova seu endereço de e-mail da caixa.



Configurando opções de ambiente adicionais usando o AWS kit de ferramentas para Visual Studio

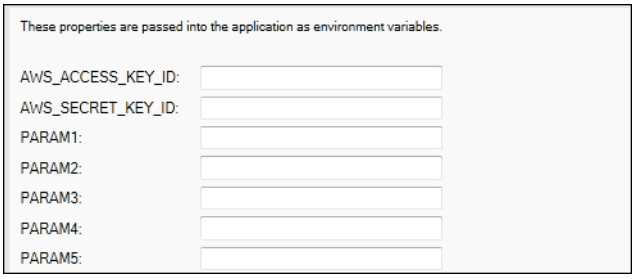
O Elastic Beanstalk define um grande número de opções de configuração que podem ser usadas para configurar o comportamento de seu ambiente e os recursos contidos neles. As opções de configuração são organizadas em namespaces como `aws:autoscaling:asg`. Cada namespace define opções para o grupo de Auto Scaling de um ambiente. O painel Advanced (Avançado) lista os namespaces da opção de configuração em ordem alfabética que você pode atualizar após a criação do ambiente.

Para obter uma lista completa de namespaces e opções, incluindo os valores padrão e compatíveis de cada um deles, consulte Opções gerais para todos os ambientes e Opções da plataforma .NET Core no Linux.



Configurando contêineres.NET Core usando o AWS kit de ferramentas para Visual Studio

O painel Container (Contêiner) permite especificar variáveis de ambiente que podem ser lidas a partir do código da aplicação.



Monitorar a integridade do aplicativo

Este tópico explica como monitorar a integridade do site da sua aplicação. É importante saber que seu de produção está disponível e respondendo às solicitações. O Elastic Beanstalk oferece

recursos para ajudar você a monitorar a capacidade de resposta da aplicação. Ele monitora as estatísticas sobre a aplicação e alerta quando os limites são excedidos.

Para obter informações sobre o monitoramento de integridade fornecido pelo Elastic Beanstalk, consulte Relatórios de integridade básica.

Você pode acessar informações operacionais sobre seu aplicativo usando o AWS Toolkit for Visual Studio ou AWS o Management Console.

O toolkit exibe o status do ambiente e a integridade do aplicativo no campo Status.

Para monitorar a integridade do aplicativo

1. No AWS Toolkit for Visual Studio, AWS no Explorer, expanda o nó do Elastic Beanstalk e, em seguida, expanda o nó do seu aplicativo.
2. Abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione View Status (Visualizar status).
3. Na guia do ambiente de aplicação, selecione Monitoring (Monitoramento).

   O painel Monitoramento inclui um conjunto de gráficos que mostra o uso de recursos do seu ambiente de aplicativos específico.

   > **Note**
   >
   > Por padrão, o intervalo de tempo é definido como a última hora. Para modificar essa configuração, na lista Time Range (Período), selecione um período diferente.

Você pode usar o AWS Toolkit for Visual Studio ou AWS o Management Console para visualizar eventos associados ao seu aplicativo.

Para visualizar eventos do aplicativo

1. No AWS Toolkit for Visual Studio, AWS no Explorer, expanda o nó do Elastic Beanstalk e o nó do seu aplicativo.
2. Abra o menu de contexto (clique com o botão direito do mouse) do ambiente de aplicação e selecione View Status (Visualizar status).
3. Na guia do ambiente de aplicação, selecione Events (Eventos).

Migração da plataforma .NET no Windows Server para a plataforma .NET Core no Linux no Elastic Beanstalk

É possível migrar aplicações executadas em plataformas .NET no Windows Server para as plataformas .NET no Linux. Veja a seguir algumas considerações ao migrar de plataformas Windows para Linux.

Considerações sobre a migração para a plataforma .NET Core no Linux

Área	Alterações e informações
Configuração do aplicativo	Em plataformas Windows, use um manifesto de implantação para especificar os aplicativos executados em seu ambiente. As plataformas .NET Core no Linux usam um Procfile para especificar as aplicações executadas nas instâncias do seu ambiente. Para obter detalhes sobre o empacotamento de aplicativos, consulte the section called “Empacotar aplicativos”.
Servidor de proxy	Nas plataformas Windows, use o IIS como servidor de proxy do aplicativo. As plataformas .NET Core no Linux incluem nginx como um proxy reverso por padrão. É possível escolher não usar nenhum servidor de proxy e usar o Kestrel como servidor Web do aplicativo. Para saber mais, consulte the section called “Servidor de proxy”.
Roteamento	Em plataformas Windows, use o IIS no código do aplicativo e inclua um manifesto de implantação para configurar o caminho do IIS. Para a plataforma .NET Core no Linux, use o roteamento ASP .NET Core no código da aplicação e atualize a configuração nginx do ambiente. Para saber mais, consulte the section called “Servidor de proxy”.
Logs	As plataformas Linux e Windows fazem streaming de logs diferentes. Para obter mais detalhes, consulte the section called “Como o Elastic Beanstalk configura os registros CloudWatch ”.

Implantar aplicações Go com o Elastic Beanstalk

Este capítulo fornece instruções para configurar e implantar sua aplicação Web Go para AWS Elastic Beanstalk. O Elastic Beanstalk facilita a implantação, o gerenciamento e o dimensionamento das aplicações Web Go usando a Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk ou o. AWS CLI APIs

Siga as step-by-step instruções QuickStart para Go para criar e implantar um aplicativo web Hello World Go com a CLI do EB.

Tópicos

- QuickStart: Implante um aplicativo Go no Elastic Beanstalk
- Configurar seu ambiente de desenvolvimento Go para o Elastic Beanstalk
- Usar a plataforma Go do Elastic Beanstalk

QuickStart: Implante um aplicativo Go no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo Go e sua implantação em um AWS Elastic Beanstalk ambiente.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação Go

- Etapa 2: implantar sua aplicação Go com a EB CLI
- Etapa 3: executar a aplicação no Elastic Beanstalk
- Etapa 4: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços de que você precisa.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Etapa 1: criar uma aplicação Go

Crie um diretório do projeto.

```
~$ mkdir eb-go
~$ cd eb-go
```

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Criaremos um serviço RESTful web “Hello World”.

Este exemplo imprime uma saudação personalizada que varia de acordo com o caminho usado para acessar o serviço.

Crie um arquivo de texto nesse diretório chamado `application.go` com o seguinte conteúdo:

Example **`~/eb-go/application.go`**

```
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)
}
```

Etapa 2: implantar sua aplicação Go com a EB CLI

Depois, você criará o ambiente da aplicação e implantará a aplicação configurada com o Elastic Beanstalk.

Para criar um ambiente e implantar o aplicativo Go

1. Inicialize o repositório da EB CLI com o comando eb init:

```
~/eb-go$ eb init -p go go-tutorial --region us-east-2
```

```
Application go-tutorial has been created.
```

Esse comando cria uma aplicação denominada `go-tutorial` e configura seu repositório local para criar ambientes com a versão da plataforma Go mais recente.

2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

```
~/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 ]
```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

```
~/eb-go$ eb create go-env
```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 3: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~/eb-go$ eb open
```

Parabéns! Você implantou uma aplicação Go com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 4: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~/eb-go$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

Depois de implantar uma ou duas aplicações demonstrativas, quando você estiver pronto para começar a desenvolver e executar aplicações Go localmente, consulte Configurar seu ambiente de desenvolvimento Go para o Elastic Beanstalk.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Configurar seu ambiente de desenvolvimento Go para o Elastic Beanstalk

Este tópico fornece instruções para configurar um ambiente de desenvolvimento Go para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Instalação do Go

Para executar aplicativos Go localmente, instale o Go Se você não precisa de uma versão específica, obtenha a versão mais recente compatível com o Elastic Beanstalk. Para obter uma lista das versões suportadas, consulte Go no documento Plataformas do AWS Elastic Beanstalk .

Baixe Go em https://golang. org/doc/install.

Instalação do AWS SDK for Go

Se você precisar gerenciar AWS recursos de dentro do seu aplicativo, instale o AWS SDK for Go usando o comando a seguir.

```
$ go get github.com/aws/aws-sdk-go
```

Para obter mais informações, consulte AWS SDK for Go.

Usar a plataforma Go do Elastic Beanstalk

Este tópico descreve como configurar, criar e executar suas aplicações Go no Elastic Beanstalk.

AWS Elastic Beanstalk suporta várias ramificações de plataforma para diferentes versões da linguagem de programação Go. Consulte Go no documento Plataformas do AWS Elastic Beanstalk para uma lista completa.

Quanto aos aplicativos simples em Go, há duas formas de implantá-los:

- Forneça um pacote de origem com um arquivo de origem na raiz chamado `application.go`, que contém o pacote principal do seu aplicativo. O Elastic Beanstalk cria o binário usando o seguinte comando:

  ```
  go build -o bin/application application.go
  ```

  Depois que a aplicação é criada, o Elastic Beanstalk a inicia na porta 5000.
- Forneça um pacote de origem com um arquivo binário chamado `application`. O arquivo binário pode ficar localizado na raiz do pacote de origem ou no diretório `bin/` do pacote de origem. Se você colocar o arquivo binário `application` nos dois locais, o Elastic Beanstalk usará o arquivo do diretório `bin/`.

  O Elastic Beanstalk inicia essa aplicação na porta 5000.

Em ambos os casos, com nossas ramificações da plataforma Go compatíveis, você também pode fornecer requisitos de módulo em um arquivo `go.mod`. Para obter mais informações, consulte Migrar para módulos Go no blog do Go.

Quanto aos aplicativos mais complexos em Go, há duas formas de implantá-los:

- Forneça um pacote de origem que inclua os arquivos de origem do aplicativo, juntamente com um Buildfile e um Procfile. O Buildfile inclui um comando para criar o aplicativo e o Procfile inclui instruções para executar o aplicativo.

- Forneça um pacote de origem que inclua os arquivos binários do aplicativo, juntamente com um Procfile. O Procfile inclui instruções para executar o aplicativo.

A plataforma Go inclui um servidor de proxy para atender a ativos estáticos e encaminhar o tráfego para sua aplicação. Você pode estender ou substituir a configuração de proxy padrão para cenários avançados.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Configurar o ambiente Go

As configurações da plataforma Go permitem que você ajuste o comportamento de suas instâncias da Amazon EC2 . Você pode editar a configuração da instância Amazon do ambiente do Elastic Beanstalk usando o console EC2 do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar o ambiente Go no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Propriedades de ambiente

A seção Propriedades do ambiente permite que você especifique as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

No ambiente Go executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando a função `os.Getenv`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código:

```
endpoint := os.Getenv("API_ENDPOINT")
```

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespace de configuração do Go

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A plataforma Go não define nenhum namespace específico da plataforma. É possível configurar o proxy para servir arquivos estáticos usando o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obter detalhes e um exemplo, consulte the section called “Arquivos estáticos”.

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console,

configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

A plataforma Go da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o ambiente Go do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Namespaces de configuração Go — Amazon Linux AMI () AL1

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

> **Note**
>
> As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.

A plataforma Go da AMI do Amazon Linux é compatível com um namespace de configuração específico da plataforma, além dos namespaces compatíveis com todas as plataformas. O

namespace `aws:elasticbeanstalk:container:golang:staticfiles` permite que você defina opções que mapeiam os caminhos em seu aplicativo Web para as pastas em seu pacote de origem do aplicativo que incluem o conteúdo estático.

Por exemplo, este arquivo de configuração informa ao servidor de proxy que deve servir arquivos da pasta `staticimages` no caminho `/images`:

Example .ebextensions/go-settings.config

```
option_settings:
  aws:elasticbeanstalk:container:golang:staticfiles:
    /html: statichtml
    /images: staticimages
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Configurar comandos de inicialização personalizados com um Procfile no Elastic Beanstalk

Para especificar comandos personalizados para iniciar um aplicativo em Go, inclua um arquivo chamado `Procfile` na raiz do pacote de origem.

Para obter detalhes sobre como gravar e usar um `Procfile`, consulte Buildfile e Procfile.

Example Procfile

```
web: bin/server
queue_process: bin/queue_processor
foo: bin/fooapp
```

Você deve chamar a aplicação principal `web` e listá-la como o primeiro comando no seu `Procfile`. O Elastic Beanstalk expõe a aplicação principal `web` no URL raiz do ambiente, por exemplo, `http://my-go-env.elasticbeanstalk.com`.

O Elastic Beanstalk também executa quaisquer aplicações cujo nome não tenha o prefixo `web_`, mas essas aplicações estão disponíveis somente dentro da instância.

O Elastic Beanstalk espera que os processos sejam executados a partir do `Procfile` para serem executados continuamente. O Elastic Beanstalk monitora essas aplicações e reinicia qualquer processo que seja encerrado. Para processos de curta execução, use um comando do Buildfile.

Usar um Procfile na AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o ambiente Go do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Passagem de portas — Amazon Linux AMI (AL1)

> **Note**
>
> As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.

O Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à aplicação no número de porta especificado na propriedade de ambiente `PORT` da aplicação. O aplicativo sempre deve escutar nessa porta. Você pode acessar essa variável em seu aplicativo chamando o método `os.Getenv("PORT")`.

O Elastic Beanstalk usa o número de porta especificado na propriedade de ambiente PORT para a porta da primeira aplicação no `Procfile` e, depois, incrementa o número da porta em 100 para cada aplicação subsequente no `Procfile`. Se a propriedade de ambiente PORT não estiver definida, o Elastic Beanstalk usará 5000 para a porta inicial.

No exemplo anterior, a propriedade de ambiente PORT para a aplicação `web` é 5000, para a aplicação `queue_process` é 5100 e para a aplicação `foo` é 5200.

Você pode especificar a porta inicial definindo a opção PORT com o namespace aws:elasticbeanstalk:application:environment, como mostra o exemplo a seguir.

```
option_settings:
  - namespace:  aws:elasticbeanstalk:application:environment
    option_name:  PORT
    value:  <first_port_number>
```

Para obter mais informações sobre como configurar variáveis de ambiente para a aplicação, consulte Configurações de opção.

Construção e configuração personalizadas com um Buildfile no Elastic Beanstalk

Para especificar um comando de criação e configuração personalizado para o aplicativo em Go, inclua um arquivo chamado `Buildfile` na raiz do pacote de origem. O nome do arquivo diferencia maiúsculas de minúsculas. Use o seguinte formato para o `Buildfile`:

```
<process_name>: <command>
```

O comando no `Buildfile` deve corresponder à expressão regular: `^[A-Za-z0-9_]+:\s*.+$`.

O Elastic Beanstalk não monitora a aplicação executada com um `Buildfile`. Use um `Buildfile` para comandos que são executados por breves períodos e são encerrados após a conclusão das tarefas. Para processos de aplicativo de longa execução que não devem ser encerrados, use o Procfile.

No seguinte exemplo de `Buildfile`, `build.sh` é um script de shell localizado na raiz do pacote de origem:

```
make: ./build.sh
```

Todos os caminhos no `Buildfile` são relativos à raiz do pacote de origem. Se você já sabe onde os arquivos residem na instância, inclua os caminhos absolutos no `Buildfile`.

Configurar o servidor de proxy

O Elastic Beanstalk usa o nginx como proxy reverso para mapear sua aplicação para o balanceador de carga do Elastic Load Balancing na porta 80. O Elastic Beanstalk oferece uma configuração nginx padrão que você pode estender ou substituir completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a propriedade do ambiente `PORT` para a porta de escuta do seu aplicativo principal.

> **Note**
>
> A porta que seu aplicativo escuta não afeta a porta na qual o servidor nginx escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as AL2 plataformas AL2 023/ suportam um recurso de configuração de proxy uniforme. Para obter mais informações sobre como configurar o servidor proxy nas versões da sua plataforma executando AL2 023/AL2, consulte. Configuração de proxy reverso

Configurar o proxy da AMI do Amazon Linux (anterior ao Amazon Linux 2)

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Se o ambiente Go do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações nesta seção.

Estendendo e substituindo a configuração padrão do proxy — Amazon Linux AMI () AL1

O Elastic Beanstalk usa nginx como proxy reverso para mapear a aplicação para o balanceador de carga na porta 80. Para fornecer a sua própria configuração de nginx, substitua a configuração padrão fornecida pelo Elastic Beanstalk, incluindo o arquivo `.ebextensions/nginx/nginx.conf` no pacote de origem. Se esse arquivo estiver presente, o Elastic Beanstalk o usará no lugar do arquivo de configuração padrão de nginx.

Para incluir diretivas além daquelas no bloco `nginx.conf` `http`, forneça também os arquivos de configuração adicionais no diretório `.ebextensions/nginx/conf.d/` do pacote de origem. Todos os arquivos desse diretório devem ter a extensão `.conf`.

Para aproveitar a funcionalidade fornecida pelo Elastic Beanstalk, como Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk, mapeamentos de aplicações automáticos e arquivos estáticos, inclua a linha a seguir no bloco `server` do arquivo de configuração de nginx:

```
include conf.d/elasticbeanstalk/*.conf;
```

Implantação de aplicações Java com o Elastic Beanstalk

Este capítulo fornece instruções para configurar e implantar seus aplicativos Java no. AWS Elastic Beanstalk O Elastic Beanstalk facilita a implantação, o gerenciamento e o dimensionamento de suas aplicações da Web Java usando o Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk, ou o. AWS CLI APIs

Siga as step-by-step instruções QuickStart para Java para criar e implantar um aplicativo web Hello World Java com a CLI do EB. Se você estiver interessado em step-by-step instruções para criar um aplicativo Java JSP simples Hello World para implantar com o EB CLI em nossa plataforma baseada em Tomcat, experimente o. QuickStart para Java no Tomcat

As ramificações da plataforma Java

AWS Elastic Beanstalk suporta duas plataformas para aplicativos Java.

- Tomcat — Uma plataforma baseada no Apache Tomcat, um contêiner web de código aberto para aplicativos que usam servlets Java e JavaServer Pages (JSPs) para atender solicitações HTTP. O Tomcat facilita o desenvolvimento de aplicativos web, fornecendo multithreading, configuração de segurança declarativa e ampla personalização. O Elastic Beanstalk tem ramificações de plataforma para cada uma das principais versões atuais do Tomcat. Para obter mais informações, consulte A Plataforma Tomcat.
- Java SE — Uma plataforma para aplicativos que não usam um contêiner da web ou usam outro que não seja o Tomcat, como Jetty ou. GlassFish Você pode incluir qualquer biblioteca Java Archives (JARs) usada pelo seu aplicativo no pacote de origem que você implanta no Elastic Beanstalk. Para obter mais informações, consulte Plataforma Java SE.

As filiais recentes das plataformas Tomcat e Java SE são baseadas no Amazon Linux 2 e versões posteriores e usam o Corretto AWS — a distribuição Java SE. Os nomes dessas ramificações de plataforma incluem a palavra Corretto em vez de Java.

Para obter uma lista das versões atuais da plataforma, consulte Tomcat e Java SE no guia Plataformas AWS Elastic Beanstalk .

AWS ferramentas

AWS fornece várias ferramentas para trabalhar com Java e Elastic Beanstalk. Independentemente da ramificação de plataforma escolhida, você pode usar o AWS SDK for Java para usar AWS outros serviços de dentro do seu aplicativo Java. O AWS SDK for Java é um conjunto de bibliotecas que permite que você AWS APIs use o código do seu aplicativo sem escrever as chamadas HTTP brutas do zero.

Se você preferir gerenciar suas aplicações a partir da linha de comando, instale a interface de linha de comando do Elastic Beanstalk (EB CLI) e use-a para criar, monitorar e gerenciar seus ambientes do Elastic Beanstalk. Se você executar vários ambientes para o seu aplicativo, a EB CLI se integrará com o Git para permitir que você associe cada um de seus ambientes a diferentes ramificações Git.

Tópicos

- QuickStart: Implante um aplicativo Java no Elastic Beanstalk
- QuickStart: implante um aplicativo web Java JSP para Tomcat no Elastic Beanstalk
- Configurar seu ambiente de desenvolvimento Java
- Mais aplicações de exemplo e tutoriais do Elastic Beanstalk para Java
- Usar a plataforma Tomcat do Elastic Beanstalk
- Usar a plataforma Java SE do Elastic Beanstalk
- Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Java Elastic Beanstalk
- Ferramentas e recursos Java

QuickStart: Implante um aplicativo Java no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo Java e sua implantação em um AWS Elastic Beanstalk ambiente.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação Java
- Etapa 2: executar a aplicação localmente
- Etapa 3: implantar sua aplicação Java com a EB CLI
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços de que você precisa.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Java e Maven

Se você não tiver o Amazon Corretto instalado no computador local, poderá instalá-lo seguindo as instruções de instalação no Guia do Usuário do Amazon Corretto.

Verifique sua instalação do Java executando o seguinte comando.

```
~$ java -version
```

Este tutorial usa o Maven. Siga as instruções de download e instalação no site do Projeto Apache Maven. Para obter mais informações sobre o Maven, consulte o Centro de usuários do Maven no site do Projeto Apache Maven.

Verifique sua instalação do Maven executando o seguinte comando.

```
~$ mvn -v
```

Etapa 1: criar uma aplicação Java

Crie um diretório do projeto.

```
~$ mkdir eb-java
~$ cd eb-java
```

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Criaremos um serviço RESTful web “Hello World”.

Este exemplo usa a framework Spring Boot. Essa aplicação abre um receptor na porta 5000. O Elastic Beanstalk encaminha solicitações para sua aplicação na porta 5000 por padrão.

Crie os seguintes arquivos:

Esse arquivo cria uma aplicação Spring Boot simples.

Example **~/eb-java/src/main/java/com/example/Application.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);
    }
}
```

Esse arquivo cria um mapeamento que retorna um String que definimos aqui.

Example **~/eb-java/src/main/java/com/example/Controller.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!";
    }
}
```

Esse arquivo define a configuração do projeto Maven.

Example **~/eb-java/pom.xml**

```
<?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>
```

Esse arquivo de propriedades substitui a porta padrão, que passa a ser 5000. Essa é a porta padrão à qual o Elastic Beanstalk envia tráfego para aplicações Java.

Example **`~/eb-java/application.properties`**

```
server.port=5000
```

Etapa 2: executar a aplicação localmente

Empacote sua aplicação com o seguinte comando:

```
~/eb-java$ mvn clean package
```

Execute sua aplicação localmente com o seguinte comando:

```
~/eb-java$ java -jar target/BeanstalkJavaExample-1.0-SNAPSHOT.jar
```

Enquanto a aplicação estiver em execução, navegue até `http://127.0.0.1:5000/` no navegador. Você verá o texto "Hello Elastic Beanstalk!".

Etapa 3: implantar sua aplicação Java com a EB CLI

Antes de implantar sua aplicação Java no Elastic Beanstalk, vamos limpar a aplicação de compilação do diretório e criar um Buildfile e um Procfile para controlar como ela é compilada e executada no ambiente Elastic Beanstalk.

Para preparar e configurar a implantação da aplicação

1. Limpe a aplicação criada.

   ```
   ~/eb-java$ mvn clean
   ```

2. Crie o `Buildfile`.

   Example **`~/eb-java/Buildfile`**

   ```
   build: mvn clean package
   ```

   Este `Buildfile` especifica o comando usado para criar a aplicação. Se você não incluir um `Buildfile` para uma aplicação Java, o Elastic Beanstalk não tentará criar essa aplicação.

3. Crie o `Procfile`.

   Example **`~/eb-java/Procfile`**

   ```
   web: java -jar target/BeanstalkJavaExample-1.0-SNAPSHOT.jar
   ```

   Este `Procfile` especifica o comando usado para executar a aplicação. Se você não incluir um `Procfile` para uma aplicação Java, o Elastic Beanstalk presumirá que há um arquivo JAR na raiz do pacote de código-fonte e tentará executá-lo com o comando `java -jar`.

Agora que você definiu os arquivos de configuração para criar e iniciar a aplicação, está pronto para implantá-la.

Para criar um ambiente e implantar sua aplicação Java

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/eb-java eb init -p corretto java-tutorial --region us-east-2
   ```

```
Application java-tutorial has been created.
```

Esse comando cria uma aplicação `java-tutorial` e configura seu repositório local para criar ambientes com a versão mais recente da plataforma Java.

2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

```
~/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 ]
```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

```
~/eb-java$ eb create java-env
```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~/eb-java eb open
```

Parabéns! Você implantou uma aplicação Java com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 5: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~/eb-java$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

> **Explore mais tutoriais**
>
> Se quiser experimentar outros tutoriais com diferentes aplicações de exemplo, consulte Aplicações de exemplo e tutoriais.

Depois de implantar uma ou duas aplicações de exemplo e estiver pronto para começar a desenvolver e executar aplicações Java localmente, consulte Configurar seu ambiente de desenvolvimento Java.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

QuickStart: implante um aplicativo web Java JSP para Tomcat no Elastic Beanstalk

Este tutorial mostra o processo de criação de uma aplicação web Java simples usando JavaServer Pages (JSPs). Se quiser agrupar vários aplicações Web na forma de arquivos WAR em um único ambiente do Elastic Beanstalk, consulte Empacotamento de vários arquivos WAR para ambientes Tomcat.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação Java JSP
- Etapa 2: implantar a aplicação Java JSP com a EB CLI
- Etapa 3: executar a aplicação no Elastic Beanstalk
- Etapa 4: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços de que você precisa.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Etapa 1: criar uma aplicação Java JSP

Crie um diretório do projeto.

```
~$ mkdir eb-tomcat
~$ cd eb-tomcat
```

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Criaremos uma aplicação Web "Hello World" .

Crie um arquivo JSP simples `index.jsp`.

Example **`~/eb-tomcat/index.jsp`**

```
<html>
  <body>
    <%out.println("Hello Elastic Beanstalk!");%>
  </body>
</html>
```

Etapa 2: implantar a aplicação Java JSP com a EB CLI

Execute os seguintes comandos para criar um ambiente Elastic Beanstalk para essa aplicação.

Para criar um ambiente e implantar sua aplicação Java JSP

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/eb-tomcat$ eb init -p tomcat tomcat-tutorial --region us-east-2
   ```

   Esse comando cria uma aplicação `tomcat-tutorial` e configura seu repositório local para criar ambientes com a versão mais recente da plataforma Tomcat.

2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

   ```
   ~/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 ]
   ```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

```
~/eb-tomcat$ eb create tomcat-env
```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 3: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~/eb-tomcat$ eb open
```

Parabéns! Você implantou uma aplicação Java JSP com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 4: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~/eb-tomcat$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

> **Explore mais tutoriais**
>
> Se quiser experimentar outros tutoriais com diferentes aplicações de exemplo, consulte Aplicações de exemplo e tutoriais.

Depois de implantar um ou dois aplicações de exemplo e estiver pronto para começar a desenvolver e executar aplicações Java em um contêiner Web Tomcat local, consulte Configurar seu ambiente de desenvolvimento Java.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Configurar seu ambiente de desenvolvimento Java

Este tópico fornece instruções para configurar um ambiente de desenvolvimento Java para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Seções

- Instalar o Java Development Kit
- Instalar um contêiner da web
- Download de bibliotecas
- Instalando o AWS SDK for Java
- Instalar um editor de texto ou IDE

Instalar o Java Development Kit

Instale o Java Development Kit (JDK). Se você não tiver uma preferência, obtenha a versão mais recente. Faça download do JDK em oracle.com

O JDK inclui o compilador Java, que você pode usar para criar arquivos de origem em arquivos de classe que podem ser executados em um servidor Web Elastic Beanstalk.

Instalar um contêiner da web

Se ainda não tiver outro contêiner ou framework Web, instale uma versão do Tomcat compatível com o Elastic Beanstalk para seu sistema operacional Amazon Linux. Para obter uma lista das versões atuais do Apache Tomcat compatíveis com o Elastic Beanstalk, consulte Tomcat no documento de

Plataformas do AWS Elastic Beanstalk . Baixe a versão do Tomcat que se aplica ao seu ambiente no site do Apache Tomcat.

Download de bibliotecas

As plataformas Elastic Beanstalk incluem poucas bibliotecas por padrão. Faça download das bibliotecas que seu aplicativo usará e salve-as na pasta do projeto para implantar no pacote de origem do aplicativo.

Se você instalou o Tomcat localmente, pode copiar as bibliotecas da API do servlet e da API JavaServer Pages (JSP) da pasta de instalação. Se você implantar em uma versão de plataforma Tomcat, não precisará incluir esses arquivos em seu pacote de origem, mas deverá tê-los no `classpath` para compilar qualquer classe que os utilizem.

JUnit, Google Guava e Apache Commons fornecem várias bibliotecas úteis. Visite a página inicial deles para saber mais:

- Baixar JUnit
- Download do Google Guava
- Download do Apache Commons

Instalando o AWS SDK for Java

Se você precisar gerenciar AWS recursos de dentro do seu aplicativo, instale o AWS SDK for Java. Por exemplo, com o AWS SDK para Java, você pode usar o Amazon DynamoDB (DynamoDB) para compartilhar estados de sessão de aplicativos Apache Tomcat em vários servidores web. Para obter mais informações, consulte Gerenciar o estado da sessão do Tomcat com o Amazon DynamoDB na documentação do AWS SDK for Java.

Visite a página inicial do AWS SDK for Java para obter mais informações e instruções de instalação.

Instalar um editor de texto ou IDE

Ambientes de desenvolvimento integrados (IDEs) fornecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você não usou um IDE para desenvolvimento Java, tente o Eclipse e o IntelliJ e veja qual é melhor para você.

- Instalação do IDE Eclipse para Java EE Developers

- Instalação do IntelliJ

Um IDE pode adicionar arquivos à pasta do seu projeto, que talvez você não queira confirmar no controle de código-fonte. Para evitar a confirmação desses arquivos no controle de código-fonte, use `.gitignore` ou a ferramenta de controle de código-fonte equivalente.

Se você deseja apenas começar a codificação e não precisa de todos os recursos de um IDE, considere instalar o Sublime Text.

> **Note**
>
> Em 31 de maio de 2023, o AWS Toolkit for Eclipse chegou ao fim de sua vida útil e não é mais suportado pelo AWS. Para obter detalhes adicionais sobre o fim do ciclo de vida do AWS Toolkit for Eclipse, consulte o arquivo README.md no repositório. AWS Toolkit for Eclipse GitHub

Mais aplicações de exemplo e tutoriais do Elastic Beanstalk para Java

Esta seção fornece aplicações e tutoriais adicionais. Os tópicos QuickStart para Java e QuickStart para Java no Tomcat, localizados anteriormente neste tópico, orientam no lançamento de uma aplicação Java de exemplo com a EB CLI.

Para começar a usar aplicativos Java AWS Elastic Beanstalk, tudo o que você precisa é de um pacote de origem do aplicativo para carregar como sua primeira versão do aplicativo e implantá-lo em um ambiente. Quando você cria um ambiente, o Elastic Beanstalk aloca todos os recursos da AWS necessários para executar uma aplicação Web escalável.

Iniciar um ambiente com um aplicativo Java de exemplo

O Elastic Beanstalk fornece exemplos de aplicativos de página única para cada plataforma, bem como exemplos mais complexos que mostram o uso de recursos AWS adicionais, como Amazon RDS e recursos específicos da linguagem ou da plataforma e. APIs

Os exemplos de página única são o mesmo código que você obtém ao criar um ambiente sem fornecer seu próprio código-fonte. Os exemplos mais complexos estão hospedados GitHub e podem precisar ser compilados ou criados antes da implantação em um ambiente do Elastic Beanstalk.

Amostras

Name	Versões com suporte	Tipo de ambie	Origem	Descrição
Tomc (págii única	Todas as ramificações da plataforma Tomcat com Corretto	Servic Web Opera	tomcat.zip	Aplicativo web Tomcat com uma única página (`index.jsp`) configurada para ser exibida na raiz do site. Para ambientes de operador, este exemplo inclui um arquivo `cron.yaml` que configura uma tarefa programada que chama `scheduled.jsp` uma vez por minuto. Quando `scheduled.jsp` é chamado, ele grava em um arquivo de log em `/tmp/sample-app.log`. Por fim, um arquivo de configuração é incluído em `.ebextensions` que copia os logs de `/tmp/` para os locais lidos pelo Elastic Beanstalk quando você solicita logs de ambiente. Se você habilitar a integração com o X-Ray em um ambiente que executa este exemplo, a aplicação mostrará conteúdo adicional relativo ao X-Ray e fornecerá uma opção para gerar informações de depuração que você pode visualizar no console do X-Ray.
Corre (págii única	Corretto 11 Corretto 8	Servic Web	corretto.zip	Aplicação Corretto com arquivos de configuração `Buildfile` e `Procfile`. Se você habilitar a integração com o X-Ray em um ambiente que executa este exemplo, a aplicação mostrará conteúdo adicional relativo ao X-Ray e fornecerá

Name	Versões com suporte	Tipo de ambie	Origem	Descrição
				uma opção para gerar informações de depuração que você pode visualizar no console do X-Ray.

Name	Versões com suporte	Tipo de ambie	Origem	Descrição
Score	Java 8	Servic Web	Clone o repositório em .com GitHub	O Scorekeep é uma API RESTful da web que usa a estrutura Spring para fornecer uma interface para criar e gerenciar usuários, sessões e jogos. A API faz parte do pacote de um aplicativo Web Angular 1.5 que a consome via HTTP. O aplicativo usa recursos da plataform a Java SE para fazer download de dependências e criar na instância, minimizando o tamanho do pacote de origem. O aplicativo também inclui arquivos de configuração nginx que substituem a configuração padrão para servir o aplicativ o web de front-end estaticamente na porta 80 através do proxy e encaminhar as solicitações para atalhos em `/api` para a API em execução em `localhost:5000` . O Scorekeep também inclui uma ramificaç ão `xray` que mostra como instrumentar uma aplicação Java para uso com o AWS X-Ray. Ele mostra a instrumentação das solicitações HTTP recebidas com um filtro de servlet, instrumentação automática e manual do cliente AWS SDK, configuração do gravador e instrumentação de solicitaç ões HTTP de saída e clientes SQL. Consulte o arquivo readme para obter instruções ou use o tutorial sobre conceitos básicos do AWS X-Ray para experimentar a aplicação com o X-Ray.

Name	Versões com suporte	Tipo de ambie	Origem	Descrição
Does it Have Snak	Tomcat 8 com Java 8	Servic Web	Clone o repositór io em .com GitHub	Tem cobras? é um aplicativo web Tomcat que mostra o uso dos arquivos de configuração do Elastic Beanstalk, Amazon RDS, JDBC, PostgreSQL, JSPs Servlets, Simple Tag Support, Tag Files, Log4J, Bootstrap e Jackson. O código-fonte deste projeto inclui um script de compilação mínimo que compila os servlets e os modelos em arquivos de classe e empacota os arquivos necessári os em um arquivo web que você pode implantar em um ambiente do Elastic Beanstalk. Consulte o arquivo readme no repositório do projeto para obter instruções completas.
Locus Load Gene	Java 8	Servic Web	Clone o repositór io em .com GitHub	Aplicação Web que pode ser usada para fazer um teste de carga em outra aplicação Web em execução em um ambiente diferente do Elastic Beanstalk . Mostra o uso de arquivos `Buildfile` e `Procfile`, DynamoDB e Locust, uma ferramenta de teste de carga de código aberto.

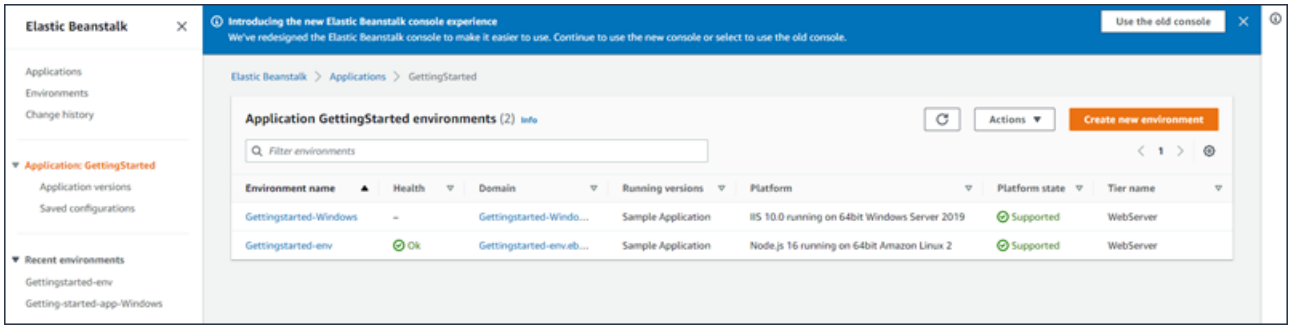
Faça download de qualquer uma das aplicações demonstrativas e implante-a no Elastic Beanstalk seguindo estas etapas:

Para iniciar um ambiente com um aplicativo de exemplo (console)

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS

2. No painel de navegação, escolha Aplicativos. Selecione uma aplicação existente na lista. Também é possível optar por criar uma aplicação, seguindo as instruções em Como gerenciar aplicações do .
3. Na página de visão geral da aplicação, escolha Create a new environment (Criar um novo ambiente).

   A imagem a seguir mostra a página de visão geral da aplicação.



   Isso inicia o assistente Create environment (Criar ambiente). O assistente fornece um conjunto de etapas para a criação de um novo ambiente.
4. Em Nível de ambiente, escolha o nível de ambiente de Ambiente do servidor Web ou Ambiente do operador. Você não pode alterar o nível de um ambiente após a criação.

   > **Note**
   >
   > O .NET na Plataforma de servidor Windows não tem suporte para a camada de ambiente do operador.

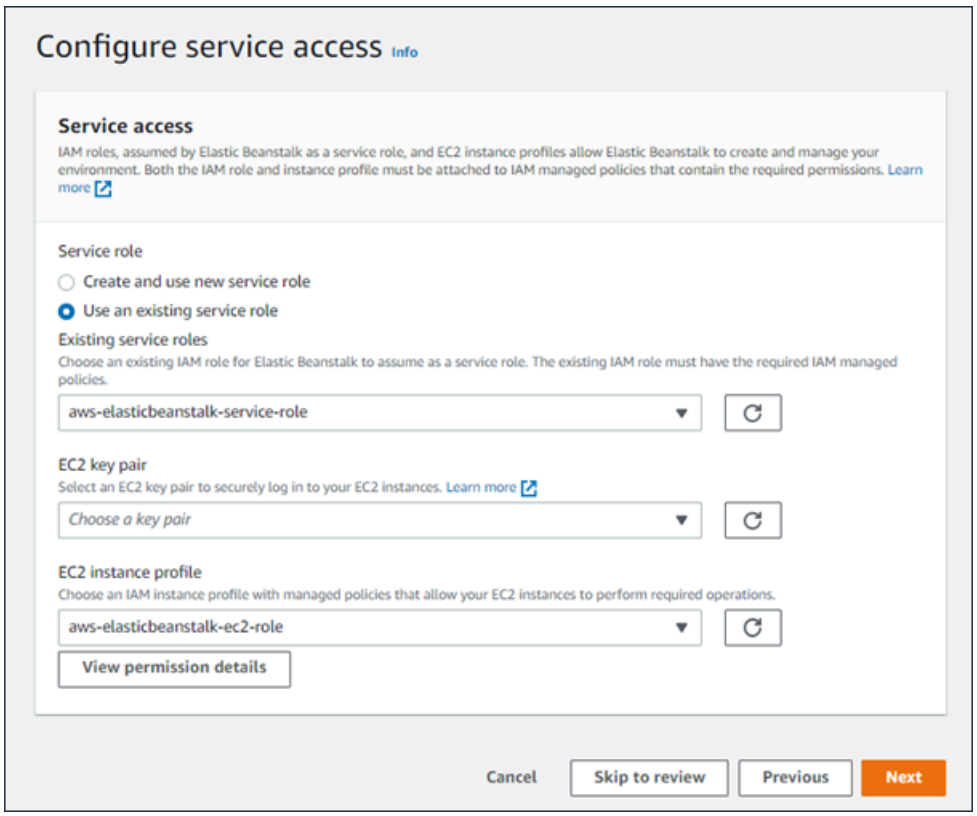
   Os campos Informações sobre a aplicação são definidos por padrão, com base na aplicação que você escolheu anteriormente.

   No grupo Informações do ambiente, o Nome do ambiente é definido por padrão com base no nome da aplicação. Se preferir um nome de ambiente diferente, insira outro valor no campo. Existe a opção de inserir um nome de Domínio. Se isso não for feito, o Elastic Beanstalk gerará automaticamente um valor. Você também pode opcionalmente inserir uma Descrição do ambiente.
5. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

> **Note**
>
> O Elastic Beanstalk é compatível com várias versões para a maioria das plataformas listadas. Por padrão, o console seleciona a versão recomendada da plataforma e a ramificação da plataforma escolhida. Se a aplicação exigir outra versão, você poderá selecioná-la aqui. Para obter informações sobre versões de plataforma compatíveis, consulte the section called “Plataformas compatíveis”.

6. Em Código da aplicação, existem algumas opções para iniciar uma aplicação de exemplo.
   - Para iniciar a aplicação de exemplo padrão sem fornecer o código-fonte, escolha Aplicação de exemplo. Essa ação escolhe a aplicação de página única que o Elastic Beanstalk fornece para a plataforma selecionada anteriormente.
   - Se você baixou uma aplicação de exemplo deste guia ou de outra fonte, siga as etapas abaixo.
     a. Selecione Carregue seu código.
     b. Em seguida, escolha Arquivo local e, em Carregar aplicação, selecione Escolher arquivo.
     c. O sistema operacional do computador apresentará uma interface para selecionar o arquivo local baixado. Selecione o arquivo do pacote de origem e continue.
7. Em Predefinições, escolha Instância única.
8. Escolha Próximo.
9. A página Configurar acesso ao serviço é exibida.

   A imagem a seguir ilustra a página Configurar acesso ao serviço.



10. Escolha um valor do menu suspenso Perfis de serviço existentes.
11. (Opcional) Se você criou um par de EC2 chaves anteriormente, poderá selecioná-lo na lista suspensa do campo de par de EC2 chaves. Você o usaria para fazer login com segurança na EC2 instância da Amazon que o Elastic Beanstalk provisiona para seu aplicativo. Se você pular essa etapa, sempre poderá criar e atribuir um par de EC2 chaves após a criação do ambiente. Para obter mais informações, consulte EC2 key pair.
12. Em seguida, vamos nos concentrar na lista suspensa do perfil da EC2 instância. Os valores exibidos nessa lista suspensa podem variar, dependendo de a conta ter criado anteriormente um novo ambiente.

    Escolha um dos seguintes itens, com base nos valores exibidos na lista.

    - Se `aws-elasticbeanstalk-ec2-role` aparecer na lista suspensa, selecione-o.
    - Se outro valor for exibido na lista e for o perfil de EC2 instância padrão destinado aos seus ambientes, selecione-o na lista suspensa.
    - Se a lista suspensa do perfil da EC2 instância não listar nenhum valor, você precisará criar um perfil da instância.

> **criar um perfil de instância**
>
> Para criar um perfil de instância, faremos um desvio para outro procedimento nessa mesma página. Vá até o final desse procedimento e expanda o procedimento a seguir, Create IAM Role for EC2 instance profile.
> Conclua as etapas em Criar função do IAM para o perfil da EC2 instância para criar uma função do IAM que você possa selecionar posteriormente para o perfil da EC2 instância. Depois, retorne para esta etapa.

Agora que você criou um perfil do IAM e atualizou a lista, ele será exibido como uma opção na lista suspensa. Selecione a função do IAM que você acabou de criar na lista suspensa do perfil da EC2 instância.

13. Escolha Skip to Review (Pular para revisão) na página Configure service access (Configurar acesso a serviço).

    Isso selecionará os valores padrão para essa etapa e pulará as etapas opcionais.

14. A página Review (Revisão) exibe um resumo de todas as suas escolhas.

    Para personalizar ainda mais o ambiente, escolha Edit (Editar) ao lado da etapa que inclui os itens que você deseja configurar. Você pode configurar as opções a seguir somente durante a criação do ambiente:

    - Nome do ambiente
    - Nome de domínio
    - Versão da plataforma
    - Processador
    - VPC
    - Nível

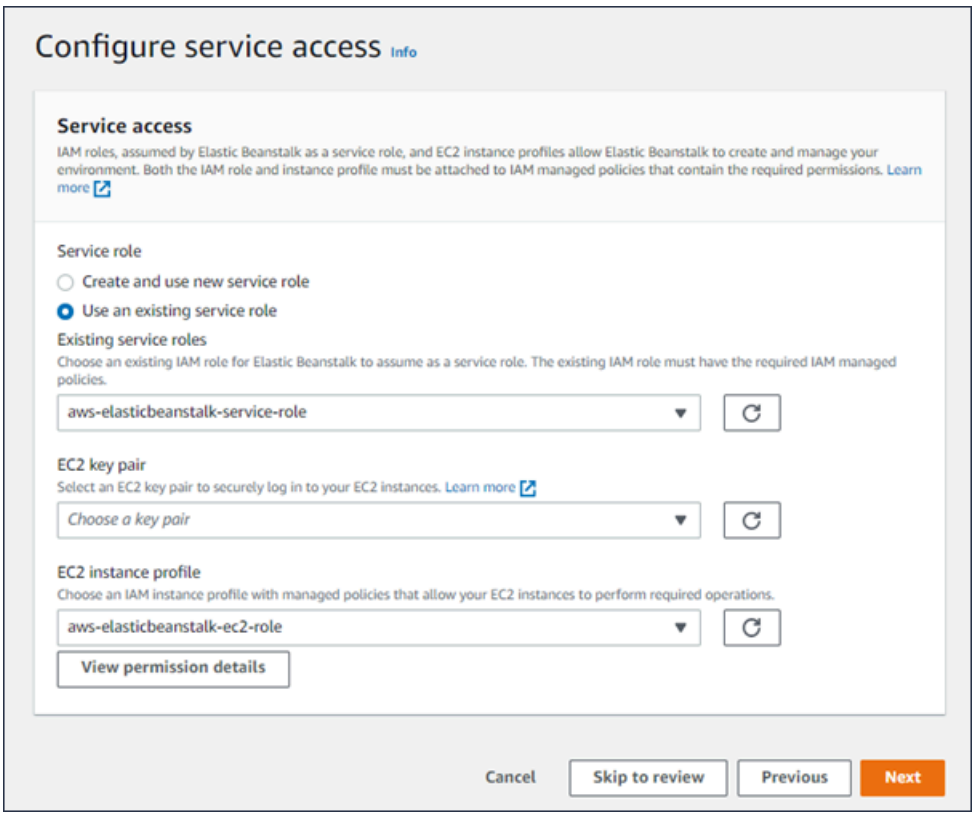
    Você pode alterar as configurações a seguir após a criação do ambiente, mas elas exigem que novas instâncias ou outros recursos sejam provisionados, e pode levar muito tempo para que elas sejam aplicadas:

- Tipo de instância, volume raiz, par de chaves e função AWS Identity and Access Management (IAM)
- Banco de dados interno do Amazon RDS
- Load balancer

Para obter detalhes sobre todas as configurações disponíveis, consulte O assistente de criação de novo ambiente.

15. Escolha Submit (enviar) na parte inferior da página para inicializar a criação do novo ambiente.

Crie uma função do IAM para o perfil da EC2 instância



Para criar o perfil da EC2 instância

1. Selecione Criar perfil.
2. Em Tipo de entidade confiável, escolha Serviços da AWS .
3. Para o caso de uso, escolha Elastic Beanstalk — Compute.
4. Escolha Próximo.

5. Verifique se as políticas de permissões incluem o seguinte e escolha Avançar:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
6. Selecione Criar perfil.
7. Volte para a guia Configurar acesso ao serviço, atualize a lista e selecione o perfil de EC2 instância recém-criado.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo completamente diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2

Depois de implantar um ou dois aplicativos de exemplo e estiver pronto para começar a desenvolver e executar aplicativos Java localmente, consulte a próxima seção para configurar um ambiente de desenvolvimento Java com todas as ferramentas e bibliotecas necessárias.

Usar a plataforma Tomcat do Elastic Beanstalk

Este tópico descreve como configurar, criar e executar suas aplicações Java que são executadas na plataforma Tomcat do Elastic Beanstalk.

A plataforma AWS Elastic Beanstalk Tomcat é um conjunto de versões da plataforma para aplicativos web Java que podem ser executados em um contêiner web Tomcat. Tomcat é executado atrás de um servidor de proxy nginx. Cada ramificação de plataforma corresponde a uma versão principal do Tomcat.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou

implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

A plataforma Tomcat do Elastic Beanstalk inclui um proxy reverso que encaminha solicitações para a aplicação. É possível usar opções de configuração para configurar o servidor de proxy para servir os ativos estáticos de uma pasta em seu código-fonte para reduzir a carga em seu aplicativo. Em cenários avançados, é possível incluir seus próprios arquivos `.conf` em seu pacote de origem para estender a configuração de proxy do Elastic Beanstalk; ou substituí-la completamente.

> **Note**
>
> O Elastic Beanstalk é compatível com nginx (o padrão) e ao Apache HTTP Server como servidores de proxy na plataforma Tomcat. Se o seu ambiente do Elastic Beanstalk Tomcat usar uma ramificação de plataforma Amazon Linux AMI (anterior ao Amazon Linux 2), você também terá a opção de usar o Apache HTTP Server Versão 2.2. O Apache (mais recente) é o padrão nessas ramificações de plataforma mais antigas.
> Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Você deve empacotar os aplicativos Java em um arquivo WAR (Web application archive) com uma estrutura específica. Para obter informações sobre a estrutura necessária e como ela está relacionada à estrutura do diretório de seu projeto, consulte Estruturar a pasta do projeto.

Para executar vários aplicativos no mesmo servidor Web, você pode empacotar vários arquivos WAR em um único pacote de origem. Cada aplicativo em um pacote de origem WAR múltiplo é executado no caminho raiz (`ROOT.war` é executado em *`myapp`*`.elasticbeanstalk.com/`) ou em um caminho diretamente abaixo dele (`app2.war` é executado no *`myapp`*`.elasticbeanstalk.com/`*`app2/`*), conforme determinado pelo nome do WAR. Em um pacote de origem de WAR único, o aplicativo sempre é executado no caminho raiz.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console.

Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Tópicos

- Configurar o grupo do seu ambiente Tomcat
- Namespaces de configuração do Tomcat
- Empacotamento de vários arquivos WAR para ambientes Tomcat
- Estruturar a pasta do projeto
- Configurar o servidor de proxy

Configurar o grupo do seu ambiente Tomcat

A plataforma Tomcat do Elastic Beanstalk fornece algumas opções específicas da plataforma, além de opções padrão que todas as plataformas têm. Essas opções permitem que você configure a máquina virtual Java (JVM), que é executada nos servidores web do ambiente, e defina propriedades do sistema que fornecem strings de configuração de informações ao aplicativo.

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar seu ambiente Tomcat no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de contêiner

É possível definir estas opções específicas da plataforma:

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, é usado nginx.

Opções de contêiner JVM

O tamanho do heap na máquina virtual Java (JVM) determina quantos objetos seu aplicativo pode criar na memória antes da execução da coleta de lixo. É possível modificar o Tamanho do heap JVM inicial (`-Xms option`) e um Tamanho do heap JVM máximo (opção `-Xmx`). Um tamanho de heap inicial maior permite que mais objetos sejam criados antes da execução da coleta de lixo, mas também significa que o coletor de lixo levará mais tempo para compactar o heap. O tamanho máximo do heap especifica a quantidade máxima de memória que a JVM pode alocar ao expandir o heap durante uma atividade intensa.

> **Note**
>
> A memória disponível depende do tipo de EC2 instância da Amazon. Para obter mais informações sobre os tipos de EC2 instância disponíveis para seu ambiente do Elastic Beanstalk, consulte Tipos de instância no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

A geração permanente é uma seção do heap da JVM que armazena definições de classe e metadados associados. Para modificar o tamanho da geração permanente, digite o novo tamanho na opção Tamanho máximo da JVM PermGen (`-XX:MaxPermSize`). Essa configuração se aplica apenas ao Java 7 e versões anteriores. Essa opção foi descontinuada no JDK 8 e substituída pela opção Size (). MaxMetaspace `-XX:MaxMetaspaceSize`

> **Important**
>
> O JDK 17 removeu a compatibilidade com a opção `-XX:MaxPermSize` do Java. O uso dessa opção com um ambiente em execução em uma ramificação da plataforma Elastic Beanstalk com o Corretto 17 resultará em um erro. O Elastic Beanstalk lançou sua primeira ramificação da plataforma executando o Tomcat com o Corretto 17 em 13 de julho de 2023. Para obter mais informações, consulte os recursos a seguir.
>
> - Site de documentação do Oracle Java: Opções do Java removidas
> - Site de documentação do Oracle Java: seção Metadados da classe em Outras considerações

Para obter mais informações sobre as plataformas do Elastic Beanstalk e seus componentes, consulte Plataformas compatíveis no Guia de plataformas do AWS Elastic Beanstalk .

Opções de log

A seção Log Options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Propriedades de ambiente

Na seção Propriedades do ambiente, você pode especificar as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

A plataforma Tomcat define uma propriedade de espaço reservado denominada `JDBC_CONNECTION_STRING` para ambientes Tomcat, para passar uma string de conexão a um banco de dados externo.

> **Note**
>
> Se você anexar uma instância de banco de dados RDS ao seu ambiente, construa a string de conexão JDBC dinamicamente a partir das propriedades do ambiente do Amazon Relational Database Service (Amazon RDS) fornecidas pelo Elastic Beanstalk. Use

JDBC_CONNECTION_STRING apenas para instâncias de banco de dados que não são provisionadas usando o Elastic Beanstalk.
Para obter mais informações sobre o uso do Amazon RDS com a aplicação Java, consulte Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Java Elastic Beanstalk.

Para versões da plataforma Tomcat lançadas antes de 26 de março de 2025, as variáveis de ambiente podem ser acessadas usando`System.getProperty()`. Por exemplo, você pode ler uma propriedade nomeada `API_ENDPOINT` a partir de uma variável com o código a seguir.

```
String endpoint = System.getProperty("API_ENDPOINT");
```

As versões da plataforma Tomcat lançadas em ou após 26 de março de 2025 também podem ser usadas `System.getenv` para acessar variáveis de ambiente de texto simples. Você pode continuar usando `System.getProperty` para acessar variáveis de ambiente de texto simples. No entanto, as variáveis de ambiente armazenadas como segredos só estão disponíveis usando`System.getenv`. Por exemplo, você pode ler uma variável de ambiente chamada `API_KEY` com o código a seguir.

```
String apiKey = System.getenv("API_KEY");
```

Important

A adição do `System.getenv()` acesso às variáveis de ambiente nas versões da plataforma Tomcat lançadas em ou após 26 de março de 2025 pode causar um comportamento inesperado em aplicativos que dão precedência às variáveis de ambiente sobre as propriedades do sistema Java ou ao mudar explicitamente de para. `System.getProperty()` `System.getenv()`
Como as propriedades do sistema (passadas pela linha de comando) exigem o escape do shell para caracteres especiais, enquanto as variáveis de ambiente não, os valores podem ser resolvidos de forma diferente ao usar variáveis de ambiente em vez das propriedades do sistema Java.
Se seu aplicativo for afetado, considere:

- Removendo caracteres de escape dos valores de propriedade do seu ambiente ao usar `System.getenv()`
- Configurando seu aplicativo para uso explícito `System.getProperty()`

- Teste minuciosamente seu aplicativo durante a atualização para garantir um comportamento consistente

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespaces de configuração do Tomcat

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A Plataforma Tomcat é compatível com as opções nos namespaces a seguir, além das opções compatíveis com todos os ambientes do Elastic Beanstalk:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: modificar as configurações do JVM. As opções desse namespace correspondem às do console de gerenciamento, da seguinte maneira:
  - `Xms` – JVM command line options (Opções de linha de comando da JVM
  - `JVM Options` – JVM command line options (Opções de linha de comando da JVM
- `aws:elasticbeanstalk:environment:proxy`: escolha o servidor de proxy do ambiente.

O seguinte arquivo de configuração de exemplo mostra o uso das opções de configuração específicas do 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
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console,

configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

A plataforma Tomcat da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Tomcat do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Namespaces de configuração do Tomcat — Amazon Linux AMI () AL1

A plataforma Tomcat da AMI do Amazon Linux é compatível com as opções adicionais nos seguintes namespaces:

- `aws:elasticbeanstalk:container:tomcat:jvmoptions`: além das opções mencionadas anteriormente nesta página para esse namespace, versões mais antigas da plataforma da AMI do Amazon Linux também são compatíveis com:
  - `XX:MaxPermSize` – Maximum JVM permanent generation size (Tamanho máximo de geração permanente da JVM
- `aws:elasticbeanstalk:environment:proxy`: além de escolher o servidor de proxy, configure também a compactação de resposta.

O exemplo de arquivo de configuração a seguir mostra o uso das opções de configuração de namespace do proxy.

Example .ebextensions/tomcat-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    GzipCompression: 'true'
    ProxyServer: nginx
```

Incluir arquivos de configuração do Elastic Beanstalk — Amazon Linux AMI () AL1

Para implantar arquivos de configuração `.ebextensions`, inclua-os na origem do aplicativo. Para um único aplicativo, adicione `.ebextensions` em um arquivo WAR compactado executando o seguinte comando:

Example

```
zip -ur your_application.war .ebextensions
```

Para uma aplicação que exige vários arquivos WAR, consulte Empacotamento de vários arquivos WAR para ambientes Tomcat para obter mais instruções.

Empacotamento de vários arquivos WAR para ambientes Tomcat

Se o aplicativo Web consiste em vários componentes do aplicativo da Web, você pode simplificar as implantações e reduzir os custos operacionais, ao executar componentes em um único ambiente, em vez de executar um ambiente separado para cada componente. Essa estratégia é eficaz para aplicativos leves e econômicos que não exigem uma grande quantidade de recursos e para ambientes de desenvolvimento e teste.

Para implantar vários aplicativos web no ambiente, reúna os arquivos WAR (web application archive) de cada componente em um único pacote de origem.

Para criar um pacote de origem do aplicativo que contenha vários arquivos WAR, organize os arquivos WAR usando a seguinte estrutura.

```
MyApplication.zip
### .ebextensions
### .platform
### foo.war
### bar.war
### ROOT.war
```

Quando você implanta um pacote de origem contendo vários arquivos WAR em um AWS Elastic Beanstalk ambiente, cada aplicativo pode ser acessado por um caminho diferente fora do nome de domínio raiz. O exemplo anterior inclui três aplicações: `foo`, `bar` e `ROOT`. `ROOT.war` é um nome de arquivo especial que instrui o Elastic Beanstalk a executar essa aplicação no domínio raiz, de forma que as três aplicações estejam disponíveis em `http://MyApplication.elasticbeanstalk.com/foo`, `http://MyApplication.elasticbeanstalk.com/bar` e `http://MyApplication.elasticbeanstalk.com`.

O pacote de origem pode incluir arquivos WAR, uma pasta `.ebextensions` opcional e uma pasta `.platform` opcional. Para obter detalhes sobre essas pastas de configuração opcionais, consulte the section called “Estender plataformas Linux”.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk com este link pré-configurado: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo aplicativo ou a plataforma do Docker para aplicativos baseados em contêiner.
3. Em Application code (Código do aplicativo), escolha Upload your code (Fazer upload do código).
4. Escolha Local file (Arquivo local) e Choose file (Escolher arquivo) e abra o pacote de origem.
5. Selecione Review and launch.
6. Revise as configurações disponíveis e, em seguida, escolha Create app.

Para obter informações sobre como criar pacotes de origem, consulte Crie um pacote de origem da aplicação Elastic Beanstalk.

Estruturar a pasta do projeto

Para funcionarem quando implantados em um servidor Tomcat, os arquivos de aplicativo web (arquivos WAR) do Java Platform Enterprise Edition (Java EE) precisam ser estruturados de acordo com determinadas diretrizes. O diretório de seu projeto não precisa atender aos mesmos padrões, mas é uma boa ideia estruturá-lo da mesma forma para simplificar a compilação e o empacotamento. A estruturação da pasta do projeto de modo similar ao conteúdo do arquivo WAR também ajuda você a entender como os arquivos estão relacionados e como eles se comportam em um servidor da web.

Na hierarquia recomendada a seguir, o código-fonte do aplicativo web é colocado em um diretório `src`, a fim e isolá-lo do script de compilação e do arquivo WAR gerado.

```
~/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
```

O conteúdo da pasta `src` corresponde ao que você empacotará e implantará no servidor, com exceção da pasta `com`. A pasta `com` contém as classes não compiladas (arquivos `.java`). Elas precisam ser compiladas e colocadas no diretório `WEB-INF/classes` para ficarem acessíveis do código do aplicativo.

O diretório `WEB-INF` contém código e configurações que não são atendidos publicamente no servidor web. As outras pastas na raiz do diretório de origem (`css`, `images` e `js`) estão disponíveis publicamente no caminho correspondente no servidor web.

O exemplo a seguir é idêntico ao diretório do projeto anterior, exceto pelo fato de conter mais arquivos e subdiretórios. Esse exemplo de projeto inclui tags, modelos e classes de suporte simples e um arquivo Java Server Pages (JSP) para um recurso `record`. Também inclui uma folha de estilo e, JavaScript para o Bootstrap, um arquivo JSP padrão e uma página de erro para erros 404.

`WEB-INF/lib` inclui um arquivo Java Archive (JAR) contendo o driver Java Database Connectivity (JDBC) para PostgreSQL. `WEB-INF/classes` está vazio porque os arquivos de classe não foram compilados ainda.

```
~/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
```

Criar um arquivo WAR com um script Shell

`build.sh` é um script de shell muito simples que compila classes Java, constrói um arquivo WAR e o copia no diretório `webapps` do Tomcat para testes locais.

```
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 ../
```

Dentro do arquivo WAR, você encontra a mesma estrutura existente no diretório `src` no exemplo anterior, com exceção da pasta `src/com`. O comando `jar` cria automaticamente o arquivo `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
```

Usar o **`.gitignore`**

Para evitar a confirmação de arquivos de classe compilados e de arquivos WAR em seu repositório Git ou a visualização de mensagens sobre a exibição deles quando você executa comandos do Git, adicione os tipos de arquivo relevantes a um arquivo denominado `.gitignore` na pasta do projeto.

~/workspace/myapp/.gitignore

```
*.zip
```

```
*.class
```

Configurar o servidor de proxy

A plataforma Tomcat usa nginx (o padrão) ou Apache HTTP Server como o proxy reverso para retransmitir solicitações da porta 80 na instância para o contêiner da Web Tomcat escutando na porta 8080. O Elastic Beanstalk oferece uma configuração de proxy padrão que pode ser estendida ou substituída completamente por sua própria configuração.

Configurar o servidor proxy na sua versão da plataforma

Todas as AL2 plataformas AL2 023/ suportam um recurso de configuração de proxy uniforme. Para obter mais informações sobre como configurar o servidor proxy nas versões da sua plataforma executando AL2 023/AL2, consulte. Configuração de proxy reverso

Configurar o proxy na plataforma Tomcat da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Tomcat do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Escolha de um servidor proxy para seu ambiente Tomcat — Amazon Linux AMI () AL1

As versões da plataforma Tomcat baseadas na AMI do Amazon Linux (anterior ao Amazon Linux 2) usam o Apache 2.4 para o proxy por padrão. Você pode optar por usar o Apache 2.2 ou o nginx incluindo um arquivo de configuração no código-fonte. O exemplo a seguir configura o Elastic Beanstalk para usar nginx.

Example .ebextensions/nginx-proxy.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: nginx
```

Migração do Apache 2.2 para o Apache 2.4 — Amazon Linux AMI () AL1

Se seu aplicativo foi desenvolvido para o Apache 2.2, leia esta seção para saber mais sobre como migrar para o Apache 2.4.

A partir das configurações da versão 3.0.0 da plataforma Tomcat, que foram lançadas com a atualização da plataforma Java com Tomcat em 24 de maio de 2018, o Apache 2.4 é o proxy padrão da plataforma Tomcat. Os arquivos `.conf` do Apache 2.4 são, em sua maioria, mas não inteiramente, compatíveis com os do Apache 2.2. O Elastic Beanstalk inclui arquivos `.conf` padrão que funcionam corretamente com cada versão do Apache. Se seu aplicativo não personalizar a configuração do Apache, conforme explicado em Estendendo e substituindo a configuração padrão do Apache — Amazon Linux AMI () AL1, ele deve migrar para o Apache 2.4 sem problemas.

Se o seu aplicativo ampliar ou substituir a configuração do Apache, talvez você precise fazer algumas alterações para migrar para o Apache 2.4. Para obter mais informações, consulte Atualizar da versão 2.2 para 2.4 no site The Apache Software Foundation. Como uma medida temporária, até você migrar com êxito para o Apache 2.4, você pode optar por usar o Apache 2.2 com seu aplicativo incluindo o seguinte arquivo de configuração em seu código-fonte.

Example .ebextensions/ .config apache-legacy-proxy

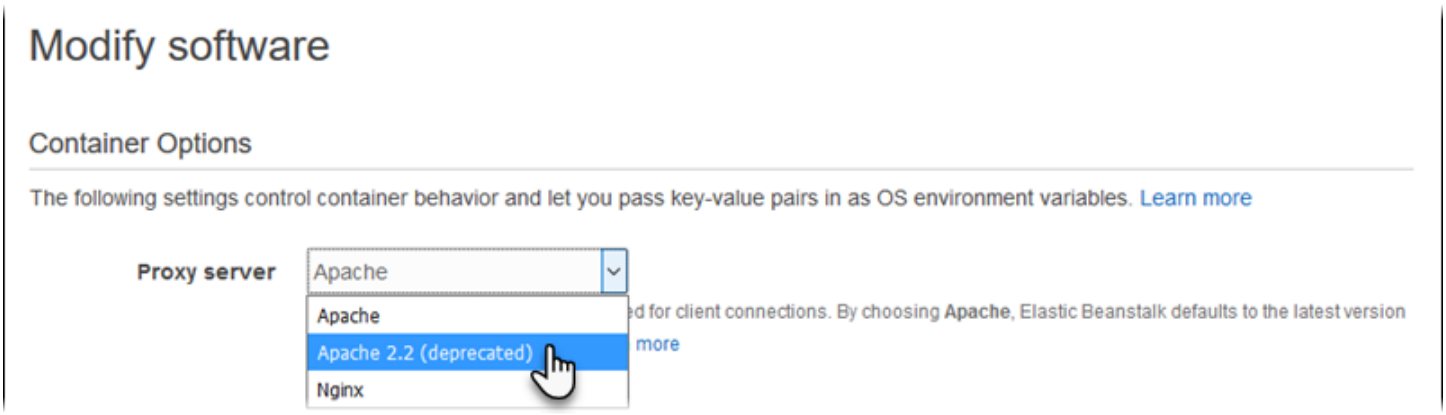
```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache/2.2
```

Para uma correção rápida, também é possível selecionar o servidor de proxy no console do Elastic Beanstalk.

Como selecionar o proxy no ambiente Tomcat no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Para Proxy server (Servidor de proxy), escolha `Apache 2.2 (deprecated)`.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.



Estendendo e substituindo a configuração padrão do Apache — Amazon Linux AMI () AL1

É possível estender a configuração padrão do Apache do Elastic Beanstalk com seus arquivos de configuração adicionais. Também há opção de substituir completamente a configuração padrão do Apache do Elastic Beanstalk.

> **Note**
>
> - Todas as plataformas do Amazon Linux 2 são compatíveis com um recurso de configuração de proxy uniforme. Para obter detalhes sobre como configurar o servidor proxy em versões da plataforma Tomcat que executam o Amazon Linux 2, consulte Configuração de proxy reverso.
> - Se você estiver migrando a aplicação do Elastic Beanstalk para uma plataforma Amazon Linux 2, leia também as informações em the section called “Migrar para 023/ AL2 AL2”.

Para estender a configuração padrão do Apache do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta chamada `.ebextensions/httpd/conf.d` no pacote de origem da aplicação. A configuração do Apache do Elastic Beanstalk inclui os arquivos `.conf` nessa pasta automaticamente.

```
~/workspace/my-app/
```

```
|-- .ebextensions
|   -- httpd
|      -- conf.d
|         -- myconf.conf
|         -- ssl.conf
-- index.jsp
```

Por exemplo, a configuração do Apache 2.4 a seguir adiciona um ouvinte na 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>
```

Para substituir completamente a configuração padrão do Apache do Elastic Beanstalk, inclua uma configuração no pacote de origem em `.ebextensions/httpd/conf/httpd.conf`.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- httpd
|       `-- conf
|           `-- httpd.conf
`-- index.jsp
```

Se você substituir a configuração do Apache do Elastic Beanstalk, adicione as linhas a seguir a `httpd.conf` para extrair as configurações do Elastic Beanstalk para Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk, compactação de resposta e arquivos estáticos.

```
IncludeOptional conf.d/*.conf
IncludeOptional conf.d/elasticbeanstalk/*.conf
```

Se seu ambiente usar o Apache 2.2 como o proxy, substitua as diretivas `IncludeOptional` por `Include`. Para obter detalhes sobre o comportamento dessas duas diretivas nas duas versões do Apache, consulte Incluir no Apache 2.4, no Apache 2.4 e Incluir IncludeOptional no Apache 2.2.

> **Note**
>
> Para substituir o listener padrão na porta 80, inclua um arquivo denominado `00_application.conf` em `.ebextensions/httpd/conf.d/elasticbeanstalk/` para substituir a configuração do Elastic Beanstalk.

Para ver um exemplo prático, examine o arquivo de configuração padrão do Elastic Beanstalk em `/etc/httpd/conf/httpd.conf` em uma instância no ambiente. Todos os arquivos na pasta `.ebextensions/httpd` em seu pacote de origem são copiados para `/etc/httpd` durante as implantações.

Estendendo a configuração padrão do nginx — Amazon Linux AMI () AL1

Para estender a configuração padrão do nginx do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta denominada `.ebextensions/nginx/conf.d/` no pacote de origem da sua aplicação. A configuração nginx do Elastic Beanstalk inclui arquivos `.conf` nesta pasta automaticamente.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- conf.d
|           |-- elasticbeanstalk
|           |   `-- my-server-conf.conf
|           `-- my-http-conf.conf
`-- index.jsp
```

Os arquivos com a extensão .conf na pasta `conf.d` são incluídos no bloco `http` da configuração padrão. Os arquivos na pasta `conf.d/elasticbeanstalk` são incluídos no bloco `server` dentro do bloco `http`.

Para substituir completamente a configuração nginx padrão do Elastic Beanstalk, inclua uma configuração em seu pacote de origem em `.ebextensions/nginx/nginx.conf`.

```
~/workspace/my-app/
```

```
|-- .ebextensions
|   `-- nginx
|       `-- nginx.conf
`-- index.jsp
```

Observações

- Se você substituir a configuração nginx do Elastic Beanstalk, adicione a linha a seguir ao bloco `server` da configuração para extrair as configurações do Elastic Beanstalk para o listener da porta 80, a compactação de resposta e os arquivos estáticos.

  ```
  include conf.d/elasticbeanstalk/*.conf;
  ```

- Para substituir o listener padrão na porta 80, inclua um arquivo denominado `00_application.conf` em `.ebextensions/nginx/conf.d/elasticbeanstalk/` para substituir a configuração do Elastic Beanstalk.
- Inclua também a linha a seguir no bloco `http` da configuração para extrair as configurações do Elastic Beanstalk para Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk e registro em log.

  ```
    include       conf.d/*.conf;
  ```

Para ver um exemplo prático, examine o arquivo de configuração padrão do Elastic Beanstalk em `/etc/nginx/nginx.conf` em uma instância no ambiente. Todos os arquivos na pasta `.ebextensions/nginx` em seu pacote de origem são copiados para `/etc/nginx` durante as implantações.

Usar a plataforma Java SE do Elastic Beanstalk

Este tópico descreve como configurar, criar e executar seus aplicativos Java que são executados na plataforma AWS Elastic Beanstalk Java SE.

A plataforma Java SE do Elastic Beanstalk é um conjunto de versões de plataforma para aplicações Web Java que podem ser executadas por conta própria a partir de um arquivo JAR compilado. Você pode compilar seu aplicativo localmente ou fazer o upload do código-fonte com um script de

compilação para compilá-lo na instância. As versões da plataforma Java SE são agrupadas em ramificações de plataforma, cada uma das quais corresponde a uma versão principal do Java.

> **Note**
>
> O Elastic Beanstalk não analisa o arquivo JAR da sua aplicação. Mantenha os arquivos de que o Elastic Beanstalk precisa fora do arquivo JAR. Por exemplo, inclua o arquivo `cron.yaml` de um ambiente de trabalho na raiz do pacote de origem do seu aplicativo, ao lado do arquivo JAR.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

A plataforma Java SE do Elastic Beanstalk inclui um servidor nginx que atua como um proxy reverso, suprindo conteúdo estático em cache e enviando solicitações à aplicação. A plataforma também oferece opções de configuração para definir o servidor de proxy para suprir os ativos estáticos de uma pasta em seu código-fonte para reduzir a carga em seu aplicativo. Em cenários avançados, você pode incluir seus próprios arquivos .conf em seu pacote de origem para estender a configuração de proxy do Elastic Beanstalk; ou substituí-la completamente.

Se você fornecer somente um único arquivo JAR para a origem da aplicação (por conta própria, não dentro de um pacote de origem), o Elastic Beanstalk altera o nome do arquivo JAR para `application.jar` e o executa usando `java -jar application.jar`. Para configurar os processos que são executados em instâncias do servidor em seu ambiente, inclua um Procfile opcional em seu pacote de origem. Um `Procfile` será necessário se você tiver mais de um JAR na raiz do pacote de origem, ou se desejar personalizar o comando Java para definir opções de JVM.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos são executados pelo Elastic Beanstalk para o aplicativo e quais argumentos são recebidos por esses processos.

Para compilar classes Java e executar outros comandos de compilação nas EC2 instâncias do seu ambiente no momento da implantação, inclua um Buildfile no pacote de origem do aplicativo. A `Buildfile` permite que você implante seu código-fonte no estado em que se encontra e construa no servidor em vez de compilar localmente JARs . A plataforma Java SE inclui ferramentas de build comuns para permitir que você crie no servidor.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Configurar o ambiente Java SE

As configurações da plataforma Java SE permitem que você ajuste o comportamento de suas instâncias da Amazon EC2 . Você pode editar a configuração da instância Amazon do ambiente do Elastic Beanstalk usando o console EC2 do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Como configurar seu ambiente Java SE no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Propriedades de ambiente

A seção Propriedades do ambiente permite que você especifique as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

No ambiente Java SE executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando `System.getenv()`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código:

```
String endpoint = System.getenv("API_ENDPOINT");
```

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespace de configuração do Java SE

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A plataforma Java SE não define nenhum namespace específico da plataforma.
É possível configurar o proxy para servir arquivos estáticos usando o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obter detalhes e um exemplo, consulte the section called “Arquivos estáticos”.

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console,

configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

A plataforma Java SE da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Java SE do Elastic Beanstalk usa uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Namespaces de configuração Java SE — Amazon Linux AMI () AL1

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A Plataforma Java SE oferece suporte a um namespace de configuração específico da Plataforma, além dos namespaces compatíveis com todas as Plataformas. O namespace `aws:elasticbeanstalk:container:java:staticfiles` permite que você defina opções que mapeiam os caminhos em seu aplicativo Web para as pastas em seu pacote de origem do aplicativo que incluem o conteúdo estático.

Por exemplo, este trecho de option_settings define duas opções no namespace de arquivos estáticos. A primeira mapeia o caminho `/public` para uma pasta denominada `public`, e a segunda mapeia o caminho `/images` para uma pasta denominada `img`:

```
option_settings:
```

```
  aws:elasticbeanstalk:container:java:staticfiles:
    /html: statichtml
    /images: staticimages
```

As pastas que você mapeia usando esse namespace devem ser pastas reais na raiz do seu pacote de origem. Você não pode mapear um caminho para uma pasta em um arquivo JAR.

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Construindo JARs no servidor com um arquivo de construção

Você pode criar os arquivos de classe e os JAR (s) do seu aplicativo nas EC2 instâncias do seu ambiente invocando um comando de construção de um `Buildfile` arquivo no seu pacote de origem.

Comandos no arquivo `Buildfile` são executados apenas uma vez e devem ser encerrados no momento da conclusão, enquanto os comandos no arquivo Procfile devem ser executados por toda a vida útil do aplicativo e serão reiniciados se forem encerrados. Para executar o JARs em seu aplicativo, use um`Procfile`.

Para obter detalhes sobre o posicionamento e a sintaxe de um `Buildfile`, consulte Buildfile e Procfile.

O exemplo acima de `Buildfile` executa o Apache Maven para criar um aplicativo web do código-fonte. Para ver um aplicativo de exemplo que usa esse recurso, consulte Exemplos de aplicativos web Java.

Example Buildfile

```
build: mvn assembly:assembly -DdescriptorId=jar-with-dependencies
```

A plataforma Java SE inclui as seguintes ferramentas de build, que você pode invocar do script de build:

- `javac`: compilador Java
- `ant`: Apache Ant
- `mvn`: Apache Maven

- `gradle`: Gradle

Configurar o processo do aplicativo com um Procfile

Se você tem mais de um arquivo JAR na raiz do seu pacote de origem da aplicação, deve incluir um arquivo `Procfile` que informa o Elastic Beanstalk quais JARs devem ser executados. Você também pode incluir um arquivo `Procfile` para um único aplicativo JAR para configurar a Java virtual machine (JVM – máquina virtual Java) que executa seu aplicativo.

Recomendamos sempre fornecer um `Procfile` no pacote de origem juntamente com o aplicativo. Dessa forma, você controla com precisão quais processos são executados pelo Elastic Beanstalk para o aplicativo e quais argumentos são recebidos por esses processos.

Para obter detalhes sobre como gravar e usar um `Procfile`, consulte Buildfile e Procfile.

Example Procfile

```
web: java -Xms256m -jar server.jar
cache: java -jar mycache.jar
web_foo: java -jar other.jar
```

O comando que executa o JAR principal em seu aplicativo deve ser denominado `web` e deve ser o primeiro comando listado no `Procfile`. O servidor nginx encaminha todas as solicitações HTTP que ele recebe do balanceador de carga do seu ambiente para essa aplicação.

O Elastic Beanstalk pressupõe que todas as entradas no Procfile devam ser sempre executadas e reinicia automaticamente qualquer aplicação encerrada definida no Procfile. Para executar comandos que serão encerrados e não deverão ser reiniciados, use um `Buildfile`.

Usar um Procfile na AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Java SE do Elastic Beanstalk usa uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.

- Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Passagem de portas — Amazon Linux AMI (AL1)

Por padrão, o Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a propriedade do ambiente `PORT` para a porta de escuta do seu aplicativo principal.

Se você usar o `Procfile` para executar várias aplicações, as versões da plataforma da AMI do Amazon Linux do Elastic Beanstalk esperam que cada aplicação adicional escute em uma porta 100 maior que a anterior. O Elastic Beanstalk define a variável PORT acessível de cada aplicação como a porta na qual ele espera que a aplicação seja executada. Você pode acessar essa variável no código do seu aplicativo chamando `System.getenv("PORT")`.

No exemplo acima de `Procfile`, o aplicativo `web` escuta na porta 5.000, `cache` escuta na porta 5.100 e `web_foo` escuta na porta 5.200. `web` configura a porta de escuta lendo a variável `PORT` e acrescenta 100 a esse número para determinar em qual porta `cache` está escutando para que ele possa enviar solicitações a ela.

Configurar o servidor de proxy

O Elastic Beanstalk usa o nginx como proxy reverso para mapear sua aplicação para o balanceador de carga do Elastic Load Balancing na porta 80. O Elastic Beanstalk oferece uma configuração nginx padrão que você pode estender ou substituir completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy nginx para encaminhar solicitações à sua aplicação na porta 5000. Você pode substituir a porta padrão definindo a propriedade do ambiente `PORT` para a porta de escuta do seu aplicativo principal.

> **Note**
>
> A porta que seu aplicativo escuta não afeta a porta na qual o servidor nginx escuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as AL2 plataformas AL2 023/ suportam um recurso de configuração de proxy uniforme. Para obter mais informações sobre como configurar o servidor proxy nas versões da sua plataforma executando AL2 023/AL2, consulte. Configuração de proxy reverso

Configurar o proxy da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o seu ambiente Java SE do Elastic Beanstalk usa uma versão da plataforma da AMI do Amazon Linux (anterior ao Amazon Linux 2), leia as informações adicionais nesta seção.

Observações

- As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
- Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Estendendo e substituindo a configuração padrão do proxy — Amazon Linux AMI () AL1

Para estender a configuração padrão do nginx do Elastic Beanstalk, adicione arquivos de configuração `.conf` a uma pasta denominada `.ebextensions/nginx/conf.d/` no pacote de origem da sua aplicação. A configuração nginx do Elastic Beanstalk inclui os arquivos `.conf` nessa pasta automaticamente.

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- conf.d
|           `-- myconf.conf
`-- web.jar
```

Para substituir completamente a configuração nginx padrão do Elastic Beanstalk, inclua uma configuração em seu pacote de origem em `.ebextensions/nginx/nginx.conf`:

```
~/workspace/my-app/
|-- .ebextensions
|   `-- nginx
|       `-- nginx.conf
`-- web.jar
```

Se você substituir a configuração nginx do Elastic Beanstalk, adicione a linha a seguir ao arquivo `nginx.conf` para extrair as configurações do Elastic Beanstalk para Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk, mapeamentos automáticos de aplicações e arquivos estáticos.

```
 include conf.d/elasticbeanstalk/*.conf;
```

A configuração demonstrativa a seguir da aplicação de exemplo Scorekeep substitui a configuração padrão do Elastic Beanstalk para atender a uma aplicação Web estática do subdiretório `public` de `/var/app/current`, em que a plataforma Java SE copia o código-fonte da aplicação. O local `/api` encaminha o tráfego para rotas sob `/api/` para o aplicativo Spring que escuta na porta 5000. O restante do tráfego é atendido pelo aplicativo web no caminho raiz.

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;
  }
}
```

Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Java Elastic Beanstalk

Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser anexado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou criado e gerenciado externamente.

Se você estiver usando o Amazon RDS pela primeira vez, adicione uma instância de banco de dados a um ambiente de teste utilizando o console do Elastic Beanstalk e verifique se a aplicação pode se conectar a ele.

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).

Nome da propriedade	Descrição	Valor da propriedade
	varia entre os mecanismos de banco de dados.	
RDS_DB_NAME	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados interna, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk. Para obter instruções sobre como configurar um banco de dados externo para uso com o Elastic Beanstalk, consulte Usar o Elastic Beanstalk com o Amazon RDS.

Para se conectar ao banco de dados, adicione o arquivo JAR do driver adequado à sua aplicação, carregue a classe do driver no seu código e crie um objeto de conexão com as propriedades de ambiente fornecidas pelo Elastic Beanstalk.

Seções

- Fazer download do driver do JDBC
- Conectar a um banco de dados (plataformas Java SE)
- Conectar a um banco de dados (plataformas Tomcat)
- Solução de problemas de conexões do banco de dados

Fazer download do driver do JDBC

Você precisará do arquivo JAR do driver JDBC para o mecanismo de banco de dados escolhido. Salve o arquivo JAR em seu código-fonte e o inclua em seu classpath quando você compilar a classe que cria conexões com o banco de dados.

Você pode encontrar o driver mais recente para o seu mecanismo de banco de dados nos seguintes locais:

- MySQL: MySQL Connector/J
- Oracle SE-1: Oracle JDBC Driver
- Postgres: PostgreSQL JDBC Driver
- SQL Server: Microsoft JDBC Driver

Para usar o driver JDBC, chame `Class.forName()` para carregá-lo antes de criar a conexão com `DriverManager.getConnection()` em seu código.

O JDBC usa uma string de conexão no seguinte formato:

```
jdbc:driver://hostname:port/dbName?user=userName&password=password
```

É possível recuperar o nome do host, a porta, o nome do banco de dados, o nome de usuário e a senha das variáveis de ambiente que o Elastic Beanstalk fornece à aplicação. O nome do driver é específico do tipo de banco de dados e da versão do driver. Os exemplos a seguir são de nomes de driver:

- `mysql` para MySQL
- `postgresql` para PostgreSQL
- `oracle:thin` para Oracle Thin
- `oracle:oci` para Oracle OCI
- `oracle:oci8` para Oracle OCI 8
- `oracle:kprb` para Oracle KPRB
- `sqlserver` para SQL Server

Conectar a um banco de dados (plataformas Java SE)

Em um ambiente Java SE, use `System.getenv()` para ler as variáveis de conexão do ambiente. O exemplo de código a seguir mostra uma classe que cria uma conexão com um banco de dados 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;
  }
```

Conectar a um banco de dados (plataformas Tomcat)

Em um ambiente Tomcat, as propriedades do ambiente são fornecidas como propriedades do sistema que podem ser acessadas com `System.getProperty()`.

O exemplo de código a seguir mostra uma classe que cria uma conexão com um banco de dados 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;
  }
```

Se você tiver problemas para obter uma conexão ou executar instruções SQL, coloque o código a seguir em um arquivo JSP. Esse código conecta a uma instância de banco de dados, cria uma tabela e grava nela.

```
<%@ 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) {}
  }
%>
```

Para exibir os resultados, coloque o seguinte código no corpo da parte HTML do arquivo JSP.

```
<p>Established connection to RDS. Read first two rows: <%= results %></p>
```

Solução de problemas de conexões do banco de dados

Se você tiver problemas para se conectar a um banco de dados a partir do seu aplicativo, revise o banco de dados e o log do contêiner da web.

Revisar logs

É possível visualizar todos os logs do seu ambiente do Elastic Beanstalk a partir do Eclipse. Se você não tiver a exibição do AWS Explorer aberta, escolha a seta ao lado do AWS ícone laranja na barra de ferramentas e escolha Mostrar exibição do AWS Explorer. Expanda AWS Elastic Beanstalk e o nome do seu ambiente, em seguida, abra o menu de contexto (clique com o botão direito do mouse) do servidor. Escolha Open in WTP Server Editor.

Escolha a guia Log da visualização Server para ver os logs agregados do ambiente. Para abrir os logs mais recentes, escolha o botão Refresh no canto superior direito da página.

Role para baixo para localizar os logs do Tomcat em `/var/log/tomcat7/catalina.out`. Se você carregou a página da web do exemplo anterior várias vezes, talvez veja o seguinte.

```
-------------------------------------
/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.
```

Todas as informações que o aplicativo web envia para a saída padrão aparecem no log do contêiner da web. No exemplo anterior, o aplicativo tenta criar a tabela sempre que a página carrega. Isso resulta na detecção de uma exceção SQL em cada carregamento de página após o primeiro.

Como exemplo, o antecedente é aceitável. Mas em aplicativos reais, mantenha suas definições de banco de dados em objetos de esquema, execute transações de dentro de classes do modelo e coordene solicitações com servlets de controlador.

Conectar a uma instância de banco de dados do RDS

É possível se conectar diretamente à instância de banco de dados do RDS no seu ambiente do Elastic Beanstalk usando a aplicação cliente MySQL.

Primeiro, abra o security group de sua instância de banco de dados do RDS para permitir o tráfego de seu computador.

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Ao lado de Endpoint, escolha o link do console do Amazon RDS.
6. Na página de detalhes da instância RDS Dashboard, em Security and Network, escolha o security group que começa com rds- ao lado de Security Groups.

> **Note**
>
> O banco de dados pode ter várias entradas com o rótulo Security Groups. Use o primeiro, que começa com awseb, somente se você tiver uma conta antiga que não tenha uma Amazon Virtual Private Cloud (Amazon VPC) padrão.

7. Em Security group details, escolha a guia Inbound e, em seguida, Edit.
8. Adicione uma regra para MySQL (porta 3306) que permita o tráfego de seu endereço IP, especificado no formato CIDR.
9. Escolha Save (Salvar). As alterações entram em vigor imediatamente.

Retorne aos detalhes de configuração do Elastic Beanstalk do seu ambiente e observe o endpoint. Você usará o nome de domínio para se conectar à instância de banco de dados do RDS.

Instale o cliente MySQL e inicie uma conexão com o banco de dados na porta 3306. No Windows, instale o MySQL Workbench a partir da página inicial do MySQL e siga as instruções.

No Linux, instale o cliente MySQL usando o gerenciador de pacote para sua distribuição. O exemplo a seguir funciona no Ubuntu e em outros derivados de 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
...
```

Depois de conectado, execute comandos SQL para ver o status do banco de dados, verificar se suas tabelas e linhas foram criadas e outras informações.

```
mysql> SELECT Resource from Beanstalk;
+--------------+
| Resource     |
+--------------+
| EC2 Instance |
| RDS Instance |
+--------------+
2 rows in set (0.01 sec)
```

Ferramentas e recursos Java

Há vários lugares que você pode ir para obter ajuda adicional durante o desenvolvimento de seus aplicativos Java.

Recurso	Descrição
O Fórum de Desenvolvimento AWS Java	Publique suas perguntas e receba comentários.
Centro do desenvolvedor Java	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.

Implantação de aplicações Node.js com o Elastic Beanstalk

Este capítulo fornece instruções para configurar e implantar seu aplicativo web Node.js no. AWS Elastic Beanstalk Ele também fornece orientações para tarefas comuns, como integração de banco de dados e trabalho com a framework Express. O Elastic Beanstalk facilita a implantação, o gerenciamento e o dimensionamento de suas aplicações da Web Node.js usando o Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk, ou o. AWS CLI APIs

Siga as step-by-step instruções QuickStart para Node.js para criar e implantar um aplicativo web Hello World Node.js com a CLI do EB.

Tópicos

- QuickStart: implante um aplicativo Node.js no Elastic Beanstalk
- Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk
- Usar a plataforma Node.js do Elastic Beanstalk
- Mais aplicações de exemplo e tutoriais do Elastic Beanstalk para Node.js
- Implantar uma aplicação Node.js Express no Elastic Beanstalk
- Implantar uma aplicação Node.js Express com clustering no Elastic Beanstalk
- Implantar uma aplicação Node.js com o DynamoDB no Elastic Beanstalk
- Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente do Node.js Elastic Beanstalk
- Ferramentas e recursos Node.js

QuickStart: implante um aplicativo Node.js no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo Node.js e sua implantação em um AWS Elastic Beanstalk ambiente.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação Node.js
- Etapa 2: executar a aplicação localmente
- Etapa 3: implantar a aplicação Node.js com a EB CLI
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços necessários.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Node.js

Instale o Node.js na máquina local seguindo Como instalar o Node.js no site do Node.js.

Verifique sua instalação do Node.js executando o comando a seguir.

```
~$ node -v
```

Etapa 1: criar uma aplicação Node.js

Crie um diretório do projeto.

```
~$ mkdir eb-nodejs
~$ cd eb-nodejs
```

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Criaremos um serviço RESTful web “Hello World”.

Example **~/eb-nodejs/server.js**

```
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}/`);
});
```

Esta aplicação abre um receptor na porta 8080. O Elastic Beanstalk encaminha solicitações para sua aplicação na porta 8080 por padrão para Node.js.

Etapa 2: executar a aplicação localmente

Execute o seguinte comando para executar sua aplicação localmente.

```
~/eb-nodejs$ node server.js
```

Você deverá ver o seguinte texto.

```
Server running at http://127.0.0.1:8080/
```

Insira o endereço URL `http://127.0.0.1:8080/` no navegador. O navegador deve exibir “Hello Elastic Beanstalk!”.

Etapa 3: implantar a aplicação Node.js com a EB CLI

Execute os seguintes comandos para criar um ambiente Elastic Beanstalk para essa aplicação.

Para criar um ambiente e implantar sua aplicação Node.js

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/eb-nodejs$ eb init -p node.js nodejs-tutorial --region us-east-2
   ```

   Este comando cria uma aplicação `nodejs-tutorial` e configura seu repositório local para criar ambientes com a versão mais recente da plataforma Node.js.

2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

   ```
   ~/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 ]
   ```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para seu aplicativo e o implanta em uma instância no ambiente. EC2 Após implantar sua aplicação, o Elastic Beanstalk o inicia na porta 8080.

```
~/eb-nodejs$ eb create nodejs-env
```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~/eb-nodejs$ eb open
```

Parabéns! Você implantou uma aplicação Node.js com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 5: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~/eb-nodejs$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

> **Explore mais tutoriais**
>
> Se quiser experimentar outros tutoriais com diferentes aplicações de exemplo, consulte Mais aplicações de exemplo e tutoriais do Elastic Beanstalk para Node.js.

Depois de implantar uma ou duas aplicações de exemplo e quando estiver pronto para começar a desenvolver e executar aplicações Node.js localmente, consulte Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk

Este tópico fornece instruções para configurar um ambiente de desenvolvimento Node.js para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Tópicos

- Instale o Node.js.
- Confirmar instalação do NPM
- Instale o AWS SDK para Node.js
- Instale o gerador Express
- Configurar uma estrutura e um servidor Express

Instale o Node.js.

Instale Node.js para executar aplicativos Node.js localmente. Se você não tiver uma preferência, obtenha a versão mais recente compatível com o Elastic Beanstalk. Consulte Node.js no documento Plataformas do AWS Elastic Beanstalk para obter uma lista das versões suportadas.

Faça download do Node.js em nodejs.org.

Confirmar instalação do NPM

O Node.js usa um gerenciador de pacotes do NPM que ajuda a instalar ferramentas e frameworks de trabalho para uso em seu aplicativo. Como o NPM é distribuído com Node.js, você o instalará

automaticamente quando baixar e instalar o Node.js. Para confirmar que você tem o NPM instalado, você pode executar o seguinte comando:

```
$ npm -v
```

Para obter mais informações sobre o NPM, visite o site npmjs .

Instale o AWS SDK para Node.js

Se você precisar gerenciar AWS recursos de dentro do seu aplicativo, instale o AWS SDK para JavaScript no Node.js. Instale o SDK com o NPM:

```
$ npm install aws-sdk
```

Visite a página inicial do AWS SDK for JavaScript in Node.js para obter mais informações.

Instale o gerador Express

O Express é um framework de aplicativo web que é executado no Node.js. Para usá-lo, primeiro instale o aplicativo de linha de comando do gerador Express. Depois que o gerador Express estiver instalado, você poderá executar o comando express para gerar uma estrutura de projeto base para sua aplicação web. Depois que o projeto base, os arquivos e as dependências estiverem instalados, você poderá iniciar um servidor Express local em sua máquina de desenvolvimento.

> **Note**
>
> - Estas etapas orientam você na instalação do gerador Express em um sistema operacional Linux.
> - Para Linux, dependendo do seu nível de permissão para diretórios do sistema, pode ser necessário prefixar alguns desses comandos com `sudo`.

Para instalar o gerador Express em seu ambiente de desenvolvimento

1. Crie um diretório de trabalho para sua estrutura e servidor Express.

   ```
   ~$ mkdir node-express
   ```

```
~$ cd node-express
```

2. Instale o Express globalmente para ter acesso ao comando `express`.

```
~/node-express$ npm install -g express-generator
```

3. Dependendo de seu sistema operacional, pode ser necessário configurar seu caminho para executar o comando `express`. A saída da etapa anterior fornece informações se você precisar definir sua variável de caminho. O seguinte é um exemplo para Linux.

```
~/node-express$ export PATH=$PATH:/usr/local/share/npm/bin/express
```

   Ao seguir os tutoriais deste capítulo, você precisará executar o comando express de diferentes diretórios. Cada tutorial configura uma estrutura básica do projeto Express em seu próprio diretório.

Agora você instalou o gerador de linha de comando Express. Você pode usá-lo para criar um diretório de estrutura para seu aplicativo da web, configurar dependências e iniciar o servidor de aplicativos da web. Em seguida, seguiremos as etapas para fazer isso no diretório `node-express` que criamos.

Configurar uma estrutura e um servidor Express

Siga estas etapas para criar os diretórios e conteúdos básicos da estrutura do Express. Os tutoriais neste capítulo também incluem estas etapas para configurar a estrutura básica do Express em cada um dos diretórios de aplicativos do tutorial.

Configurar uma estrutura e um servidor Express

1. Execute o comando `express`. Isso gera `package.json`, `app.js` e alguns diretórios.

```
~/node-express$ express
```

   Quando solicitado, digite **y** se você deseja continuar.

2. Configure as dependências locais.

```
~/node-express$ npm install
```

3. Verifique se o servidor de aplicativos da web é inicializado.

```
~/node-express$ npm start
```

Você deve ver uma saída semelhante a:

```
> nodejs@0.0.0 start /home/local/user/node-express
> node ./bin/www
```

Por padrão, o servidor é executado na porta 3000. Para testá-lo, execute `curl http://localhost:3000` em outro terminal ou abra um navegador no computador local e insira o endereço URL `http://localhost:3000`.

Pressione Ctrl+C para interromper o servidor.

Usar a plataforma Node.js do Elastic Beanstalk

Este tópico descreve como configurar, criar e executar suas aplicações Node.js no Elastic Beanstalk.

AWS Elastic Beanstalk suporta várias ramificações de plataforma para diferentes versões da linguagem de programação Node.js. Consulte Node.js no documento Plataformas do AWS Elastic Beanstalk para uma lista completa.

O Elastic Beanstalk fornece opções de configuração que podem ser usadas para personalizar o software que é executado nas instâncias do EC2 no ambiente Elastic Beanstalk. É possível configurar as variáveis de ambiente necessárias à aplicação, habilitar a alternância de logs para o Amazon S3 e mapear pastas da origem da aplicação que contenham arquivos estáticos para caminhos atendidos pelo servidor de proxy.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

É possível incluir um arquivo `Package.json` no pacote de fontes para instalar pacotes durante a implantação, fornecer um comando iniciar e especificar a versão do Node.js que você deseja que a aplicação use. Você pode incluir um arquivo `npm-shrinkwrap.json` para bloquear versões de dependência.

A plataforma Node.js inclui um servidor proxy para servir ativos estáticos, encaminhar tráfego para a aplicação e compactar respostas. Você pode estender ou substituir a configuração de proxy padrão para cenários avançados.

Existem várias opções para iniciar seu aplicativo. Você pode adicionar um Procfile ao pacote de fontes para especificar o comando que inicia a aplicação. Quando você não fornece um `Procfile`, mas fornece um arquivo `package.json`, o Elastic Beanstalk executa `npm start`. Se você não fornecer nenhum dos dois, o Elastic Beanstalk procurará o arquivo `app.js` ou `server.js`, nessa ordem, e o executará.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Configurar o ambiente Node.js

Você pode usar as configurações da Node.js plataforma para ajustar o comportamento de suas instâncias da Amazon EC2 . Você pode editar a configuração da EC2 instância da Amazon para seu ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk.

Você pode usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Para configurar o ambiente Node.js no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de contêiner

É possível definir estas opções específicas da plataforma:

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, o NGINX é usado.

Opções de log

A seção Log Options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Propriedades de ambiente

Use a seção Propriedades do ambiente para especificar as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. Essas definições são passadas para o aplicativo como pares de chave-valor.

Dentro do Node.js ambiente em que é executado AWS Elastic Beanstalk, você pode acessar as variáveis de ambiente executando`process.env.ENV_VARIABLE`.

```
var endpoint = process.env.API_ENDPOINT
```

A plataforma Node.js define a variável de ambiente PORT como a porta para a qual o servidor proxy direciona o tráfego. Para obter mais informações, consulte Configurar o servidor de proxy.

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Configurar um ambiente Node.js do Amazon Linux AMI (anterior ao Amazon Linux 2)

As categorias a seguir de configuração de software do console só são compatíveis com um ambiente Node.js do Elastic Beanstalk que use uma versão da plataforma Amazon Linux AMI (anterior ao Amazon Linux 2).

Observações

- As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
- Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Opções de contêiner — Amazon Linux AMI (AL1)

Na página de configuração, especifique o seguinte:

- Proxy server (Servidor proxy): especifica o servidor Web a ser usado para conexões do proxy com o Node.js. Por padrão, o NGINX é usado. Se você selecionar none (nenhum), os mapeamentos de arquivos não terão efeito e a compactação GZIP será desabilitada.
- Node.js version (Versão do): especifica a versão do Node.js. Para obter uma lista das versões compatíveis do Node.js, consulte Node.js, no guia Plataformas AWS Elastic Beanstalk .

- GZIP compression (Compactação): especifica se a compactação GZIP está habilitada. Por padrão, a compactação GZIP está habilitada.
- Node command (Comando de nó): permite inserir o comando usado para iniciar a aplicação Node.js. Uma string vazia (o padrão) significa que o Elastic Beanstalk usa `app.js`, em seguida `server.js` e depois `npm start`.

Namespace de configuração do Node.js

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

É possível escolher o proxy a ser usado nas instâncias do seu ambiente usando o namespace `aws:elasticbeanstalk:environment:proxy`. O exemplo a seguir configura o ambiente para usar o servidor proxy Apache HTTPD.

Example .ebextensions/nodejs-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: apache
```

É possível configurar o proxy para servir arquivos estáticos usando o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles`. Para obter mais informações e um exemplo, consulte the section called “Arquivos estáticos”.

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

A plataforma Node.js do Amazon Linux AMI (anterior ao Amazon Linux 2)

Se o ambiente Node.js do Elastic Beanstalk usar uma versão da plataforma Amazon Linux AMI (anterior ao Amazon Linux 2), considere as configurações e recomendações específicas nesta seção.

Observações

- As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
- Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Node.jsopções de configuração específicas da plataforma — Amazon Linux AMI () AL1

O Elastic Beanstalk é compatível com algumas opções de configurações específicas da plataforma para versões da plataforma Node.js do Amazon Linux AMI. Você pode escolher o servidor proxy para ser executado na frente da aplicação, uma versão específica do Node.js para ser executada e o comando usado para executar a aplicação.

Como servidor proxy, você pode usar um servidor proxy NGINX ou Apache. Você pode definir o valor none para a opção `ProxyServer`. Nesse caso, o Elastic Beanstalk executa a aplicação como autônoma, não por atrás de algum servidor de proxy. Se o ambiente executar uma aplicação autônoma, atualize o código para escutar a porta para a qual o NGINX direciona o tráfego.

```
var port = process.env.PORT || 8080;

app.listen(port, function() {
  console.log('Server running at http://127.0.0.1:%s', port);
});
```

Node.jsversões de idioma — Amazon Linux AMI (AL1)

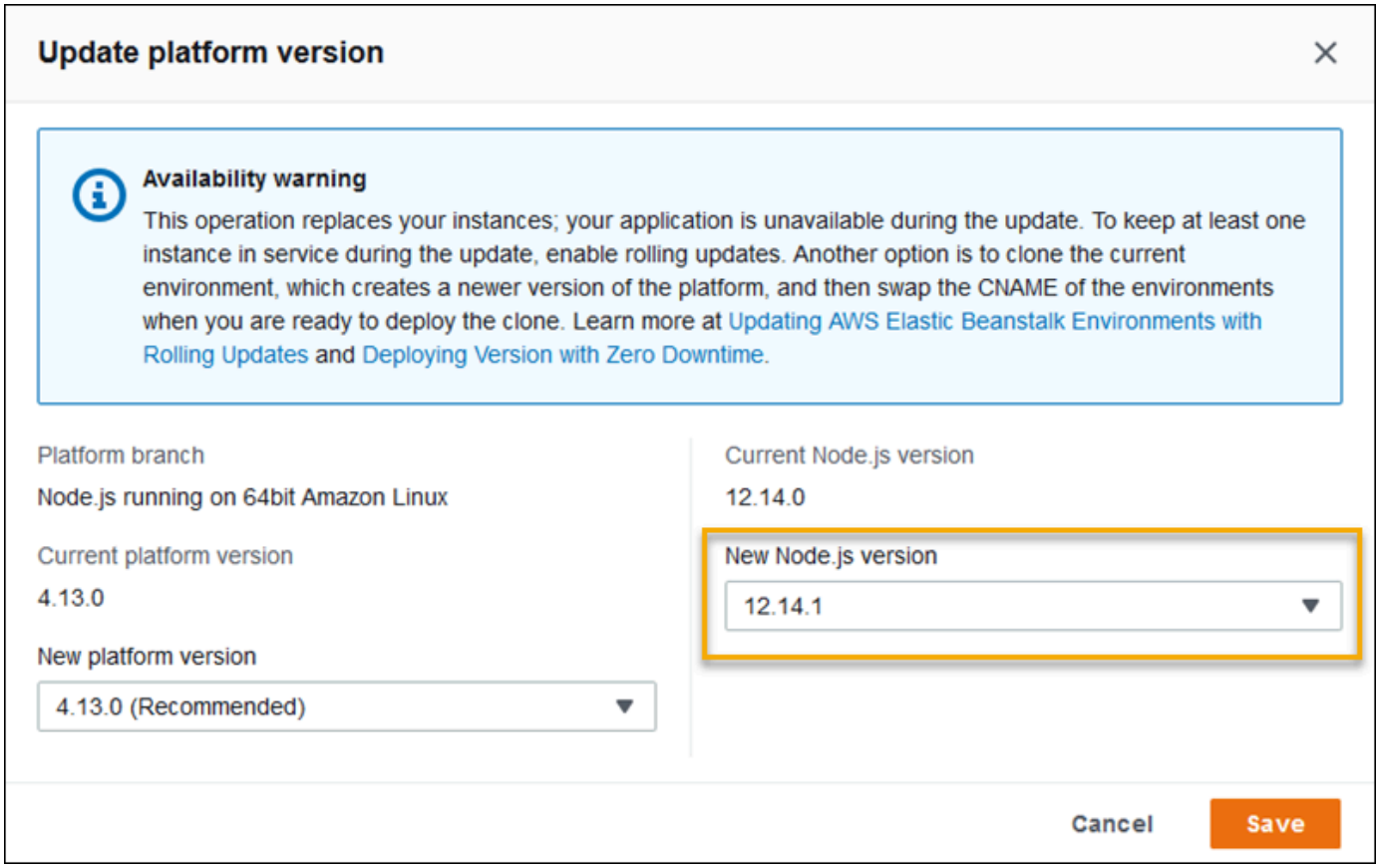
Em termos de versão de linguagem compatível, a plataforma Node.js do Amazon Linux AMI é diferente de outras plataformas gerenciadas pelo Elastic Beanstalk. Isso porque cada versão da plataforma Node.js só é compatível com algumas versões da linguagem Node.js. Para obter uma lista das versões compatíveis do Node.js, consulte Node.js, no guia Plataformas AWS Elastic Beanstalk .

Você pode usar uma opção de configuração específica da plataforma para definir a versão da linguagem. Para instruções, consulte the section called “Configurar o ambiente Node.js”. Ou então, use o console do Elastic Beanstalk para atualizar a versão do Node.js que o ambiente usa, como parte da atualização da versão da plataforma.

> **Note**
>
> Quando o suporte oferecido para a versão do Node.js que você estiver usando for removido da plataforma, será necessário alterar ou remover a configuração da versão antes de fazer a atualização da plataforma. Isso pode ocorrer quando uma vulnerabilidade de segurança é identificada para uma ou mais versões do Node.js.
> Quando isso acontece, a tentativa de atualizar para uma nova versão da plataforma que não suporta a configuração NodeVersionfalha. Para evitar a necessidade de criar um novo ambiente, altere a opção de NodeVersionconfiguração para uma versão do Node.js compatível com a versão antiga e a nova da plataforma, ou remova a configuração da opção e execute a atualização da plataforma.

Para configurar a versão do Node.js do ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, em Platform (Plataforma), escolha Change (Alterar).
4. Na caixa de diálogo Update platform version (Atualizar versão da plataforma), selecione uma versão do Node.js.



5. Escolha Salvar.

Node.jsnamespaces de configuração — Amazon Linux AMI () AL1

A plataforma Node.js do Amazon Linux AMI define opções adicionais nos namespaces `aws:elasticbeanstalk:container:nodejs:staticfiles` e `aws:elasticbeanstalk:container:nodejs`.

O arquivo de configuração a seguir diz ao Elastic Beanstalk para usar `npm start` para executar a aplicação. Ele também define o tipo de proxy como Apache e permite compactação. Por fim, ele configura o proxy para servir arquivos estáticos de dois diretórios de origem. Uma origem são os arquivos HTML no caminho `html` na raiz do site a partir do diretório de origem `statichtml`. A outra fonte são arquivos de imagem no `images` caminho sob a raiz do site a partir do diretório de origem `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
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Configurar comandos de inicialização personalizados com um Procfile no Elastic Beanstalk

Você pode incluir um arquivo denominado `Procfile` na raiz do pacote de fontes para especificar o comando que inicia a aplicação.

Example Procfile

```
web: node index.js
```

Para obter informações sobre o uso de `Procfile`, consulteBuildfile e Procfile.

> **Note**
>
> Esse recurso substitui a opção `NodeCommand` herdada no namespace `aws:elasticbeanstalk:container:nodejs`.

Configurar dependências da sua aplicação no Elastic Beanstalk

A aplicação pode ter dependências em alguns módulos Node.js, como os que você especifica nas instruções `require()`. Esses módulos são armazenados em um diretório `node_modules`. Quando a aplicação é executada, o Node.js carrega os módulos desse diretório. Para obter mais informações, consulte Loading from node_modules folders (Carregar das pastas node_modules) na documentação do Node.js.

Você pode especificar essas dependências usando um arquivo `package.json`. Se o Elastic Beanstalk detectar esse arquivo e não houver um diretório `node_modules` presente, o Elastic

Beanstalk executa `npm install` como usuário de aplicativo da web. O comando `npm install` instala as dependências no diretório `node_modules`, que o Elastic Beanstalk cria de antemão. O comando `npm install` acessa os pacotes listados no arquivo `package.json` do registro npm público ou outros locais. Para obter mais informações, consulte o site npm Docs.

Se o Elastic Beanstalk detectar o diretório `node_modules`, não executa o `npm install`, mesmo que exista um arquivo `package.json`. O Elastic Beanstalk pressupõe que os pacotes de dependências estejam disponíveis no diretório `node_modules` para o Node.js acessar e carregar.

As seções a seguir fornecem mais informações sobre como estabelecer as dependências do módulo Node.js para a aplicação.

> **Note**
>
> Se você tiver algum problema de implantação quando o Elastic Beanstalk estiver executando o `npm install`, considere a adoção de uma abordagem alternativa. Inclua o diretório `node_modules` com os módulos de dependências no pacote de fontes da aplicação. Dessa forma você contorna os problemas de instalação das dependências do registro npm público enquanto você investiga o problema. Como a origem dos módulos de dependências é um diretório local, fazer isso também pode ajudar a reduzir o tempo de implantação. Para obter mais informações, consulte Incluir dependências do Node.js em um diretório node_modules.

Especificar as dependências do Node.js com um arquivo package.json

Use um arquivo `package.json` na raiz do código-fonte do projeto para especificar pacotes de dependência e para fornecer um comando para iniciar. Quando existe um arquivo `package.json` presente e não existe um diretório `node_modules` na raiz da origem do projeto, o Elastic Beanstalk executa `npm install` como usuário do aplicativo da web para instalar dependências do registro npm público. O Elastic Beanstalk também usa o comando `start` para iniciar a aplicação. Para obter mais informações sobre o arquivo `package.json`, consulte Specifying dependencies in a `package.json` file (Especificar dependências em um arquivo) no site npm Docs.

Use a palavra-chave `scripts` para fornecer um comando para iniciar. Atualmente, a palavra-chave `scripts` é usada em vez da opção herdada `NodeCommand` no namespace `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"
    }
  }
```

Modo de produção e dependências de desenvolvimento

Para especificar suas dependências no arquivo `package.json`, use os atributos dependências e devDependencies. O atributo dependências designa os pacotes exigidos pelo seu aplicativo em produção. O atributo devDependencies designa pacotes que são necessários apenas para desenvolvimento e testes locais.

O Elastic Beanstalk executa `npm install` como usuário do aplicativo da web com os seguintes comandos. As opções de comando variam dependendo da versão npm incluída na ramificação da plataforma em que seu aplicativo é executado.

- npm v6 — O Elastic Beanstalk instala dependências no modo de produção por padrão. Ele usa o comando `npm install --production`.
- npm v7 ou superior — O Elastic Beanstalk omite as devDependencies. Ele usa o comando `npm install --omit=dev`.

Ambos os comandos listados acima não instalam os pacotes que são devDependencies.

Se você precisar instalar os pacotes devDependencies, defina a propriedade de ambiente NPM_USE_PRODUCTION como`false`. Com esta configuração, não usaremos as opções acima ao executar o npm install. Isso resultará na instalação dos pacotes devDependencies.

SSH e HTTPS

A partir do lançamento da plataforma Amazon Linux 2 de 7 de março de 2023, você também pode usar os protocolos SSH e HTTPS para recuperar pacotes de um repositório Git. A ramificação da plataforma Node.js 16 suporta os protocolos SSH e HTTPS. Node.js 14 suporta somente o protocolo HTTPS.

Example package.json —Node.js 16 suporta HTTPS e 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"
    }
```

Versões e intervalos de versões

> **Important**
>
> O recurso para especificar intervalos de versões não está disponível para ramificações da plataforma Node.js executadas em AL2 023. Só oferecemos suporte a uma versão do Node.js em uma ramificação específica do Node.js em AL2 023. Se o arquivo `package.json` especificar um intervalo de versões, ele será ignorado e a versão da ramificação da plataforma do Node.js será usada por padrão.

Use a palavra-chave `engines` no arquivo `package.json` para especificar a versão do Node.js que você deseja que a aplicação use. Você também pode especificar um intervalo de versões usando a notação npm. Para obter mais informações sobre a sintaxe para intervalos de versões, consulte Semantic Versioning using npm (Versionamento semântico usando o npm) no site Node.js. A palavra-chave `engines` no arquivo `package.json` do Node.js substitui a opção herdada `NodeVersion` no namespace `aws:elasticbeanstalk:container:nodejs`.

Example **`package.json`**: versão única do Node.js

```
{
    ...
    "engines": { "node" : "14.16.0" }
  }
```

Example **`package.json`**: intervalo de versões do Node.js

```
{
    ...
    "engines": { "node" : ">=10 <11" }
  }
```

Quando um intervalo de versões é indicado, o Elastic Beanstalk instala a versão do Node.js mais recente que a plataforma tem disponível dentro do intervalo. Neste exemplo, o intervalo indica que a versão deve ser maior ou igual à versão 10, mas menor que a versão 11. Como resultado, o Elastic Beanstalk instala a versão mais recente do Node.js, a 10.x.y, disponível na plataforma compatível.

Lembre-se de que você só pode especificar uma versão do Node.js que corresponda à ramificação da plataforma. Por exemplo, se você estiver usando a ramificação da plataforma Node.js 16, só poderá especificar uma versão 16.x.y do Node.js. Você pode usar as opções de intervalo de versões suportadas pelo npm para permitir mais flexibilidade. Para saber quais são as versões válidas do Node.js para cada ramificação da plataforma, consulte Node.js no guia Plataformas do AWS Elastic Beanstalk .

> **Note**
>
> Quando o suporte oferecido para a versão do Node.js que você estiver usando for removido da plataforma, será necessário alterar ou remover a configuração da versão do Node.js antes de fazer a atualização da plataforma. Isso pode ocorrer quando uma vulnerabilidade de segurança é identificada para uma ou mais versões do Node.js.
> Nesse caso, ocorre uma falha na tentativa de atualizar para uma nova versão da plataforma não compatível com a versão do Node.js configurada. Para evitar a necessidade de criar um novo ambiente, altere a configuração da versão Node.jspackage.json para uma versão Node.js suportada pela versão antiga da plataforma e a nova. Você tem a opção de especificar um intervalo de versão Node.js que inclua uma versão suportada, conforme descrito anteriormente neste tópico. Você também tem a opção de remover a configuração e, em seguida, implantar o novo pacote de fonte.

Incluir dependências do Node.js em um diretório node_modules

Para implantar pacotes de dependência em instâncias de ambiente juntamente com o código da aplicação, inclua-os em um diretório chamado `node_modules` na raiz da fonte de seu projeto. Para

obter mais informações, consulte Downloading and installing packages locally (Baixar e instalar pacotes localmente) no site npm Docs.

Quando você implanta um `node_modules` diretório em uma versão da AL2 Node.js plataforma AL2 023/, o Elastic Beanstalk presume que você está fornecendo seus próprios pacotes de dependências e evita instalar dependências especificadas em um arquivo package.json. Node.jsprocura dependências no `node_modules` diretório. Para obter mais informações, consulte Loading from node_modules Folders (Carregar das pastas node_modules) na documentação do Node.js.

> **Note**
>
> Se você tiver algum problema de implantação enquanto o Elastic Beanstalk estiver executando `npm install`, considere a abordagem descrita neste tópico como uma solução alternativa enquanto investiga o problema.

Bloquear dependências com npm shrinkwrap no Elastic Beanstalk

A plataforma Node.js executa o `npm install` como usuário do aplicativo da web sempre que você faz uma implantação. Quando há novas versões das dependências disponíveis, elas são instaladas quando você implanta a aplicação, o que pode causar demora na implantação.

Você pode evitar a atualização de dependências criando um arquivo `npm-shrinkwrap.json` que bloqueia as dependências do aplicativo para a versão atual.

```
$ npm install
$ npm shrinkwrap
wrote npm-shrinkwrap.json
```

Inclua esse arquivo no seu pacote de origem para garantir que as dependências sejam instaladas somente uma vez.

Configurar o servidor de proxy

O Elastic Beanstalk usa o NGINX ou o Apache HTTPD como proxy reverso para mapear a aplicação para o balanceador de carga do Elastic Load Balancing na porta 80. O padrão é NGINX. O Elastic Beanstalk oferece uma configuração de proxy padrão que pode ser estendida ou modificada completamente por sua própria configuração.

Por padrão, o Elastic Beanstalk configura o proxy para encaminhar solicitações à sua aplicação na porta 5000. É possível modificar a porta padrão definindo a PORT propriedade de ambiente para a porta de escuta da sua aplicação principal.

> **Note**
>
> A porta na qual a aplicação escuta não afeta a porta à qual o servidor NGINXescuta para receber solicitações do balanceador de carga.

Configurar o servidor proxy na sua versão da plataforma

Todas as AL2 plataformas AL2 023/ suportam um recurso de configuração de proxy uniforme. Para obter mais informações sobre como configurar o servidor proxy nas versões da sua plataforma executando AL2 023/AL2, consulte. Configuração de proxy reverso

Configurar o proxy da AMI do Amazon Linux (anterior ao Amazon Linux 2)

Se o ambiente Node.js do Elastic Beanstalk usar uma versão da plataforma Amazon Linux AMI (anterior ao Amazon Linux 2), leia as informações desta seção.

> **Observações**
>
> - As informações neste tópico se aplicam somente às ramificações da plataforma baseadas no Amazon Linux AMI (AL1). AL2As ramificações AL2 da plataforma 023/ são incompatíveis com as versões anteriores da plataforma Amazon Linux AMI (AL1) e exigem configurações diferentes.
> - Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Estendendo e substituindo a configuração padrão do proxy — Amazon Linux AMI () AL1

A plataforma Node.js usa um proxy reverso para retransmitir solicitações da porta 80 na instância para a aplicação que está escutando na porta 8081. O Elastic Beanstalk oferece uma configuração de proxy padrão que pode ser estendida ou modificada completamente por sua própria configuração.

Para estender a configuração padrão, adicione arquivos `.conf` a `/etc/nginx/conf.d` com um arquivo de configuração. Para um exemplo específico, consulte Encerramento de HTTPS em EC2 instâncias que executam o Node.js.

A plataforma Node.js define a variável de ambiente PORT como a porta para a qual o servidor proxy direciona o tráfego. Leia essa variável no seu código para configurar a porta da sua aplicação.

```
    var port = process.env.PORT || 3000;

    var server = app.listen(port, function () {
        console.log('Server running at http://127.0.0.1:' + port + '/');
    });
```

A configuração padrão do NGINX encaminha o tráfego para um servidor upstream denominado `nodejs` em `127.0.0.1:8081`. É possível remover a configuração padrão e fornecer a sua própria em um arquivo de configuração.

Example .ebextensions/proxy.config

O exemplo a seguir remove a configuração padrão e adiciona uma configuração personalizada que encaminha o tráfego para a porta 5000, em vez de 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"
```

A configuração de exemplo (`/etc/nginx/conf.d/proxy.conf`), usa a configuração padrão em `/etc/nginx/conf.d/00_elastic_beanstalk_proxy.conf` como base para incluir o bloco de

servidor padrão com configurações de compactação e log, bem como um mapeamento de arquivo estático.

O comando `removeconfig` remove a configuração padrão do contêiner para garantir que o servidor proxy use a configuração personalizada. O Elastic Beanstalk recria a configuração padrão quando cada configuração é implantada. Para explicar isso, no exemplo a seguir, um post-configuration-deployment hook (`/opt/elasticbeanstalk/hooks/configdeploy/post/99_kill_default_nginx.sh`) é adicionado. Isso remove a configuração padrão e reinicia o servidor de proxy.

> **Note**
>
> A configuração padrão pode ser alterada em versões futuras da plataforma Node.js. Use a versão mais recente da configuração como base para suas personalizações para garantir a compatibilidade.

Se você substituir a configuração padrão, deverá definir os mapeamento de arquivos estáticos e a compactação GZIP. Isso ocorre porque a plataforma não pode aplicar as configurações padrão.

Mais aplicações de exemplo e tutoriais do Elastic Beanstalk para Node.js

Esta seção fornece aplicações e tutoriais adicionais. O tópico QuickStart para Node.js, localizado anteriormente neste tópico, orienta pela inicialização da aplicação de exemplo Node.js com a EB CLI.

Para começar a usar os aplicativos Node.js AWS Elastic Beanstalk, tudo o que você precisa é de um pacote de origem do aplicativo para carregar como sua primeira versão do aplicativo e implantá-lo em um ambiente.

Iniciar um ambiente com um aplicativo Node.js de exemplo

O Elastic Beanstalk fornece exemplos de aplicativos de página única para cada plataforma, bem como exemplos mais complexos que mostram o uso de recursos AWS adicionais, como Amazon RDS e recursos específicos da linguagem ou da plataforma e. APIs

> **Note**
>
> Siga as etapas no arquivo `README.md` do pacote de origem para implantá-lo.

Amostras

Tipo de ambiente	Pacote de origem	Descrição
Servidor Web	nodejs.zip	Aplicativo de página única. Para iniciar a aplicação de exemplo com a EB CLI, consulte QuickStart para Node.js. Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.
Servidor Web com o Amazon RDS	nodejs-example-express-rds.zip	Aplicativo de log de caminhada que usa a framework Express e um Amazon Relational Database Service (RDS). Tutorial
Servidor Web com Amazon ElastiCache	nodejs-example-express-elasticache.zip	Aplicativo web expresso que usa a Amazon ElastiCache para agrupamento. O clustering aprimora a alta disponibilidade, a performance e a segurança do seu aplicativo web. Tutorial
Servidor Web com DynamoDB, Amazon SNS	nodejs-example-	Site Express que coleta informações de contato dos usuários para uma nova campanha de marketing da empresa. Usa o AWS SDK do Node.js para gravar entradas JavaScript em uma tabela do DynamoDB e os

Tipo de ambiente	Pacote de origem	Descrição
e Amazon SQS	dynamo.zip	arquivos de configuração do Elastic Beanstalk para criar recursos no DynamoDB, no Amazon SNS e no Amazon SQS. Tutorial

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo completamente diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Para obter detalhes sobre a implantação de aplicativo, consulte Implantar uma nova versão do seu aplicativo.

Depois de implantar um ou dois aplicativos de exemplo e quando estiver pronto para começar a desenvolver e executar aplicativos Node.js localmente, consulte Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk para configurar um ambiente de desenvolvimento Node.js com todas as ferramentas e bibliotecas necessárias.

Implantar uma aplicação Node.js Express no Elastic Beanstalk

Esta seção na implantação de uma aplicação de amostra no Elastic Beanstalk usando a Interface de Linha de Comando do Elastic Beanstalk (EB CLI) e, em seguida, atualizando a aplicação para usar a framework Express.

Pré-requisitos

Este tutorial requer os seguintes pré-requisitos:

- Os tempos de execução do Node.js
- O software gerenciador de pacotes Node.js padrão, npm
- O gerador de linha de comando Express
- A Elastic Beanstalk Command Line Interface (EB CLI)

Para obter detalhes sobre como instalar os três primeiros componentes listados e configurar seu ambiente de desenvolvimento local, consulte Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk. Para este tutorial, você não precisa instalar o AWS SDK para Node.js, que também é mencionado no tópico referenciado.

Para obter detalhes sobre como instalar e configurar a EB CLI, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Criar um ambiente do Elastic Beanstalk

Seu diretório de aplicativos

Este tutorial usa um diretório chamado `nodejs-example-express-rds` para o pacote de origem do aplicativo. Crie o diretório `nodejs-example-express-rds` para este tutorial.

```
~$ mkdir nodejs-example-express-rds
```

> **Note**
>
> Cada tutorial neste capítulo usa seu próprio diretório para o pacote de origem do aplicativo. O nome do diretório corresponde ao nome do aplicativo de amostra usado pelo tutorial.

Altere seu diretório de trabalho atual para `nodejs-example-express-rds`.

```
~$ cd nodejs-example-express-rds
```

Agora, vamos configurar um ambiente do Elastic Beanstalk executando a plataforma Node.js e o aplicativo de amostra. Usaremos a interface de linha de comando do Elastic Beanstalk (EB CLI).

Para configurar um repositório EB CLI para seu aplicativo e criar um ambiente Elastic Beanstalk executando a plataforma Node.js

1. Crie um repositório com o comando eb init.

   ```
   ~/nodejs-example-express-rds$ eb init --platform node.js --region <region>
   ```

Esse comando cria um arquivo de configuração em uma pasta chamada `.elasticbeanstalk` que especifica as configurações para a criação de ambientes para a aplicação e cria uma aplicação do Elastic Beanstalk com nome baseado na pasta atual.

2. Crie um ambiente executando um aplicativo de exemplo com o comando eb create.

```
~/nodejs-example-express-rds$ eb create --sample nodejs-example-express-rds
```

Esse comando cria um ambiente com balanceamento de carga com as configurações padrão da plataforma do Node.js e os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.

- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*. *region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

3. Quando a criação do ambiente for concluída, use o comando eb open para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-express-rds$ eb open
```

Agora você criou um ambiente Node.js Elastic Beanstalk com um aplicativo de amostra. Você pode atualizá-lo com seu próprio aplicativo. Em seguida, atualizamos o aplicativo de exemplo para usar a framework do Express.

Atualize o aplicativo para usar o Express

Depois de criar um ambiente com um aplicativo de amostra, você pode atualizá-lo com seu próprio aplicativo. Neste procedimento, primeiro executamos os comandos npm install e express para configurar a estrutura Express no diretório do seu aplicativo. Em seguida, usamos a EB CLI para atualizar seu ambiente Elastic Beanstalk com o aplicativo atualizado.

Para atualizar seu aplicativo para usar o Express

1. Execute o comando `express`. Isso gera `package.json`, `app.js` e alguns diretórios.

   ```
   ~/nodejs-example-express-rds$ express
   ```

   Quando solicitado, digite **y** se você deseja continuar.

   > **Note**
   >
   > Se o comando express não funcionar, talvez você não tenha instalado o gerador de linha de comando Express conforme descrito na seção Pré-requisitos anterior. Ou talvez seja necessário configurar a configuração do caminho do diretório para sua máquina local para executar o comando express. Consulte a seção Pré-requisitos para obter etapas detalhadas sobre como configurar seu ambiente de desenvolvimento, para que você possa continuar com este tutorial.

2. Configure as dependências locais.

   ```
   ~/nodejs-example-express-rds$ npm install
   ```

3. (Opcional) Verifique a inicialização do servidor do aplicativo da web.

   ```
   ~/nodejs-example-express-rds$ npm start
   ```

   Você deve ver uma saída semelhante a:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   > node ./bin/www
   ```

   Por padrão, o servidor é executado na porta 3000. Para testá-lo, execute `curl http://localhost:3000` em outro terminal ou abra um navegador no computador local e insira o endereço URL `http://localhost:3000`.

   Pressione Ctrl+C para interromper o servidor.

4. Implante as alterações em seu ambiente Elastic Beanstalk com o comando eb deploy.

   ```
   ~/nodejs-example-express-rds$ eb deploy
   ```

5. Assim que o ambiente estiver verde e pronto, atualize o URL para verificar se funcionou. Você deve ver uma página da web que diz Welcome to Express.

Em seguida, vamos atualizar o aplicativo Express para servir arquivos estáticos e adicionar uma nova página.

Para configurar arquivos estáticos e adicionar uma nova página ao seu aplicativo Express

1. Adicione um segundo arquivo de configuração na pasta `.ebextensions` com o seguinte conteúdo:

   `nodejs-example-express-rds/.ebextensions/staticfiles.config`

   ```
   option_settings:
       aws:elasticbeanstalk:environment:proxy:staticfiles:
           /stylesheets: public/stylesheets
   ```

   Isso configura o servidor de proxy para servir arquivos na pasta `public` no caminho `/public` do aplicativo. Servir arquivos estaticamente do servidor proxy reduz a carga em seu aplicativo. Para obter mais informações, consulte Arquivos estáticos anterior deste capítulo.
2. (Opcional) Para confirmar se os mapeamentos estáticos estão configurados corretamente, comente a configuração do mapeamento estático em `nodejs-example-express-rds/app.js`. Isso remove o mapeamento do aplicativo de nó.

   ```
   //  app.use(express.static(path.join(__dirname, 'public')));
   ```

   Os mapeamentos de arquivo estático no arquivo `staticfiles.config` da etapa anterior ainda devem carregar a folha de estilo com êxito, mesmo depois de comentar essa linha. Para verificar se os mapeamentos de arquivo estático são carregados por meio da configuração de arquivo estático do proxy, em vez do aplicativo expresso, remova os valores a seguir `option_settings:`. Depois de ter sido removido da configuração do arquivo estático e do aplicativo de nó, a folha de estilo falhará ao carregar.

   Lembre-se de redefinir o conteúdo do `nodejs-example-express-rds/app.js` e `staticfiles.config` quando terminar o teste.
3. Adicionar `nodejs-example-express-rds/routes/hike.js`. Digite o seguinte:

   ```
   exports.index = function(req, res) {
   ```

```
 res.render('hike', {title: 'My Hiking Log'});
};

exports.add_hike = function(req, res) {
};
```

4. Atualize `nodejs-example-express-rds/app.js` para incluir três novas linhas.

   Primeiro, adicione a seguinte linha para adicionar um `require` a essa rota:

   ```
   var hike = require('./routes/hike');
   ```

   O seu arquivo deve ser semelhante ao seguinte snippet:

   ```
   var express = require('express');
   var path = require('path');
   var hike = require('./routes/hike');
   ```

   Em seguida, adicione as seguintes duas linhas ao `nodejs-example-express-rds/app.js` depois de `var app = express();`

   ```
   app.get('/hikes', hike.index);
   app.post('/add_hike', hike.add_hike);
   ```

   O seu arquivo deve ser semelhante ao seguinte snippet:

   ```
   var app = express();
   app.get('/hikes', hike.index);
   app.post('/add_hike', hike.add_hike);
   ```

5. Copie `nodejs-example-express-rds/views/index.jade` para `nodejs-example-express-rds/views/hike.jade`.

   ```
   ~/nodejs-example-express-rds$ cp views/index.jade views/hike.jade
   ```

6. Implante as alterações com o comando eb deploy.

   ```
   ~/nodejs-example-express-rds$ eb deploy
   ```

7. O ambiente será atualizado após alguns minutos. Depois que seu ambiente estiver verde e pronto, verifique se funcionou atualizando seu navegador e anexando **`hikes`** no final do URL (por exemplo, `http://node-express-env-syypntcz2q.elasticbeanstalk.com/hikes`).

   Você deve ver uma página da web intitulada My Hiking Log.

Agora, você criou um aplicativo da Web que usa a estrutura Express. Na próxima seção, modificaremos o aplicativo para usar um Amazon Relational Database Service (RDS) para armazenar um log de caminhada.

Atualize o aplicativo para usar o Amazon RDS

Na próxima etapa, atualizaremos o aplicativo para usar o Amazon RDS para MySQL.

Para atualizar seu aplicativo para usar o RDS for MySQL

1. Para criar um banco de dados RDS para MySQL acoplado ao seu ambiente Elastic Beanstalk, siga as instruções no tópico Adicionar um banco de dados incluído posteriormente neste capítulo. Adicionar uma instância de banco de dados leva cerca de 10 minutos.
2. Atualize a seção de dependências no `package.json` com o seguinte conteúdo:

```
"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. Executar npm install.

```
~/nodejs-example-express-rds$ npm install
```

4. Atualize `app.js` para se conectar ao banco de dados, criar uma tabela e inserir um único log de caminhada padrão. Sempre que este aplicativo for implantado, ele descartará a tabela de caminhadas anterior e a recriará.

```
/**
 * 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. Adicione o seguinte conteúdo ao arquivo `routes/hike.js`. Isso permitirá que as rotas insiram novos logs de caminhada no banco de dados 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. Substitua o conteúdo de `routes/index.js` pelo seguinte:

```
/*
 * GET home page.
 */

exports.index = function(req, res){
  res.render('index', { title: 'Express' });
};
```

7. Adicione o seguinte modelo de jade ao `views/hike.jade` para fornecer a interface do usuário para adicionar logs de caminhada.

```
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. Implante as alterações com o comando eb deploy.

```
~/nodejs-example-express-rds$ eb deploy
```

Limpeza

Se você terminar de trabalhar com o Elastic Beanstalk, poderá encerrar seu ambiente.

Use o comando eb terminate para encerrar seu ambiente e todos os recursos que ele contém.

```
~/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.
...
```

Implantar uma aplicação Node.js Express com clustering no Elastic Beanstalk

Este tutorial mostra como implantar um aplicativo de amostra no Elastic Beanstalk usando a interface de linha de comando (EB CLI) do Elastic Beanstalk e, em seguida, atualizando o aplicativo para usar a estrutura Express, a Amazon e o clustering. ElastiCache O clustering aprimora a alta disponibilidade, a performance e a segurança do seu aplicativo web. Para saber mais sobre a Amazon ElastiCache, acesse O que é a Amazon ElastiCache (Memcached)? no Guia do usuário da Amazon ElastiCache (Memcached).

> **Note**
>
> Este exemplo cria AWS recursos, pelos quais você pode ser cobrado. Para obter mais informações sobre AWS preços, consultehttps://aws.amazon.com/pricing/. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte https://aws.amazon.com/free/ para obter mais informações.

Pré-requisitos

Este tutorial requer os seguintes pré-requisitos:

- Os tempos de execução do Node.js
- O software gerenciador de pacotes Node.js padrão, npm
- O gerador de linha de comando Express
- A Elastic Beanstalk Command Line Interface (EB CLI)

Para obter detalhes sobre como instalar os três primeiros componentes listados e configurar seu ambiente de desenvolvimento local, consulte Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk. Para este tutorial, você não precisa instalar o AWS SDK para Node.js, que também é mencionado no tópico referenciado.

Para obter detalhes sobre como instalar e configurar a EB CLI, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Criar um ambiente do Elastic Beanstalk

Seu diretório de aplicativos

Este tutorial usa um diretório chamado `nodejs-example-express-elasticache` para o pacote de origem do aplicativo. Crie o diretório `nodejs-example-express-elasticache` para este tutorial.

```
~$ mkdir nodejs-example-express-elasticache
```

> **Note**
>
> Cada tutorial neste capítulo usa seu próprio diretório para o pacote de origem do aplicativo. O nome do diretório corresponde ao nome do aplicativo de amostra usado pelo tutorial.

Altere seu diretório de trabalho atual para `nodejs-example-express-elasticache`.

```
~$ cd nodejs-example-express-elasticache
```

Agora, vamos configurar um ambiente do Elastic Beanstalk executando a plataforma Node.js e o aplicativo de amostra. Usaremos a interface de linha de comando do Elastic Beanstalk (EB CLI).

Para configurar um repositório EB CLI para seu aplicativo e criar um ambiente Elastic Beanstalk executando a plataforma Node.js

1. Crie um repositório com o comando eb init.

   ```
   ~/nodejs-example-express-elasticache$ eb init --platform node.js --region <region>
   ```

   Esse comando cria um arquivo de configuração em uma pasta chamada `.elasticbeanstalk` que especifica as configurações para a criação de ambientes para a aplicação e cria uma aplicação do Elastic Beanstalk com nome baseado na pasta atual.
2. Crie um ambiente executando um aplicativo de exemplo com o comando eb create.

   ```
   ~/nodejs-example-express-elasticache$ eb create --sample nodejs-example-express-elasticache
   ```

   Esse comando cria um ambiente com balanceamento de carga com as configurações padrão da plataforma do Node.js e os seguintes recursos:

   - EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

     Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
   - Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
   - Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.

- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

3. Quando a criação do ambiente for concluída, use o comando eb open para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-express-elasticache$ eb open
```

Agora você criou um ambiente Node.js Elastic Beanstalk com um aplicativo de amostra. Você pode atualizá-lo com seu próprio aplicativo. Em seguida, atualizamos o aplicativo de exemplo para usar a framework do Express.

Atualize o aplicativo para usar o Express

Atualize a aplicação demonstrativa no ambiente do Elastic Beanstalk para usar a framework do Express.

Você pode baixar o código-fonte final em nodejs-example-express-elasticache.zip.

Para atualizar seu aplicativo para usar o Express

Depois de criar um ambiente com um aplicativo de amostra, você pode atualizá-lo com seu próprio aplicativo. Neste procedimento, primeiro executamos os comandos npm install eexpress para configurar a estrutura Express no diretório do seu aplicativo.

1. Execute o comando `express`. Isso gera `package.json`, `app.js` e alguns diretórios.

   ```
   ~/nodejs-example-express-elasticache$ express
   ```

   Quando solicitado, digite **y** se você deseja continuar.

   > **Note**
   >
   > Se o comando express não funcionar, talvez você não tenha instalado o gerador de linha de comando Express conforme descrito na seção Pré-requisitos anterior. Ou talvez seja necessário configurar a configuração do caminho do diretório para sua máquina local para executar o comando express. Consulte a seção Pré-requisitos para obter etapas detalhadas sobre como configurar seu ambiente de desenvolvimento, para que você possa continuar com este tutorial.

2. Configure as dependências locais.

   ```
   ~/nodejs-example-express-elasticache$ npm install
   ```

3. (Opcional) Verifique a inicialização do servidor do aplicativo da web.

   ```
   ~/nodejs-example-express-elasticache$ npm start
   ```

   Você deve ver uma saída semelhante a:

   ```
   > nodejs@0.0.0 start /home/local/user/node-express
   ```

```
> node ./bin/www
```

Por padrão, o servidor é executado na porta 3000. Para testá-lo, execute `curl http://localhost:3000` em outro terminal ou abra um navegador no computador local e insira o endereço URL `http://localhost:3000`.

Pressione Ctrl+C para interromper o servidor.

4. Renomeie `nodejs-example-express-elasticache/app.js` para `nodejs-example-express-elasticache/express-app.js`.

```
~/nodejs-example-express-elasticache$ mv app.js express-app.js
```

5. Atualize a linha `var app = express();` no `nodejs-example-express-elasticache/express-app.js` da seguinte forma:

```
var app = module.exports = express();
```

6. No seu computador local, crie um arquivo chamado `nodejs-example-express-elasticache/app.js` com o seguinte código.

```
/**
 * 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. Substitua o conteúdo do arquivo `nodejs-example-express-elasticache/bin/www` com o seguinte:

```
#!/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. Implante as alterações em seu ambiente Elastic Beanstalk com o comando eb deploy.

```
~/nodejs-example-express-elasticache$ eb deploy
```

9. O ambiente será atualizado após alguns minutos. Assim que o ambiente estiver verde e pronto, atualize o URL para verificar se funcionou. Você deverá ver uma página da web dizendo: "Welcome to Express".

Você pode acessar os registros das EC2 instâncias que executam seu aplicativo. Para obter instruções sobre como acessar os logs, consulte Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

Em seguida, vamos atualizar o aplicativo Express para usar a Amazon ElastiCache.

Para atualizar seu aplicativo Express para usar a Amazon ElastiCache

1. No computador local, crie um diretório `.ebextensions` no diretório de nível superior do seu pacote de origem. Neste exemplo, usamos `nodejs-example-express-elasticache/.ebextensions`.
2. Crie um arquivo de configuração `nodejs-example-express-elasticache/.ebextensions/elasticache-iam-with-script.config` com o seguinte snippet. Para obter mais informações sobre o arquivo de configuração, consulte Namespace de configuração do Node.js. Isso cria um usuário do IAM com as permissões necessárias para descobrir os nós do ElastiCache e grava em um arquivo sempre que o cache muda. Você também pode copiar o arquivo do nodejs-example-express-elasticachearquivo.zip. Para obter mais informações sobre as ElastiCache propriedades, consulteExemplo: ElastiCache..

> **Note**
>
> YAML depende de um recuo consistente. Compare o nível de recuo ao substituir o conteúdo em um arquivo de configuração de exemplo e se confira se o editor de texto usa espaços, e não caracteres de tabulação, como recuo.

```
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. No seu computador local, crie um arquivo de configuração `nodejs-example-express-elasticache/.ebextensions/elasticache_settings.config` com o seguinte trecho para configurar. ElastiCache

```
option_settings:
  "aws:elasticbeanstalk:customoption":
     CacheNodeType: cache.t2.micro
     NumCacheNodes: 1
     Engine: memcached
     NodeListPath: /var/nodelist
```

4. No seu computador local, substitua `nodejs-example-express-elasticache/express-app.js` pelo seguinte snippet. Esse arquivo lê a lista de nós do disco (`/var/nodelist`) e configura o Express para usar `memcached` como armazenamento de sessão se os nós estiverem presentes. Seu arquivo deve ter uma aparência semelhante à seguinte:

```
/**
 * 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. No computador local, atualize `package.json` com o seguinte conteúdo:

```
  "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. Executar npm install.

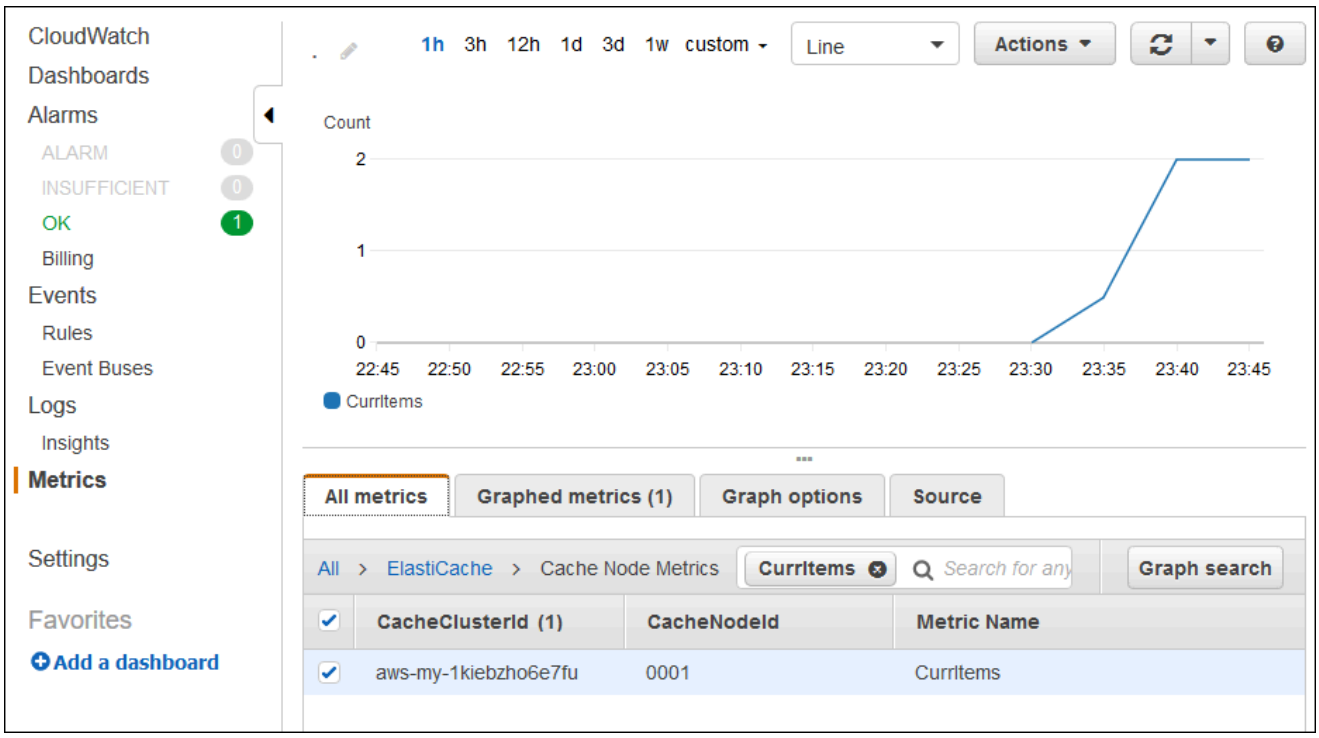
```
~/nodejs-example-express-elasticache$ npm install
```

7. Implante o aplicativo atualizado.

```
~/nodejs-example-express-elasticache$ eb deploy
```

8. O ambiente será atualizado após alguns minutos. Depois que seu ambiente estiver verde e pronto, verifique se o código funcionou.

   a. Verifique o CloudWatch console da Amazon para ver suas ElastiCache métricas. Para visualizar suas ElastiCache métricas, selecione Métricas no painel esquerdo e, em seguida, pesquise por CurrItems. Selecione ElastiCache > Métricas do nó de cache e, em seguida, selecione seu nó de cache para visualizar o número de itens no cache.



> **Note**
>
> Verifique se você está visualizando a mesma região na qual implantou seu aplicativo.

Se você copiar e colar o URL do aplicativo em outro navegador da Web e atualizar a página, verá sua CurrItem contagem aumentar após 5 minutos.

b. Faça um snapshot dos logs. Para obter mais informações sobre recuperação de logs, consulte Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

c. Verifique o arquivo `/var/log/nodejs/nodejs.log` no pacote de logs. Você deverá ver algo semelhante a:

```
Using memcached store nodes:
[ 'aws-my-1oys9co8zt1uo.1iwtrn.0001.use1.cache.amazonaws.com:11211' ]
```

Limpeza

Se não quiser mais executar seu aplicativo, você pode limpar encerrando seu ambiente e excluindo o aplicativo.

Use o comando `eb terminate` para encerrar o seu ambiente e o comando `eb delete` para excluir o seu aplicativo.

Para encerrar seu ambiente

No diretório onde você criou o repositório local, execute `eb terminate`.

```
$ eb terminate
```

Esse processo pode levar alguns minutos. O Elastic Beanstalk exibirá uma mensagem quando o ambiente foi encerrado com êxito.

Implantar uma aplicação Node.js com o DynamoDB no Elastic Beanstalk

Este tutorial e seu aplicativo de exemplo nodejs-example-dynamo.zip orientam você pelo processo de implantação de um aplicativo Node.js que usa o AWS SDK JavaScript em Node.js para interagir com o serviço Amazon DynamoDB. Você criará uma tabela do DynamoDB em um banco de dados desacoplado ou externo do ambiente. AWS Elastic Beanstalk Você também configurará o aplicativo para usar um banco de dados desacoplado. Em um ambiente de produção, é uma prática recomendada usar um banco de dados desacoplado do ambiente Elastic Beanstalk para que seja independente do ciclo de vida do ambiente. Essa prática também permite que você execute implantações azul/verde.

O aplicativo de exemplo ilustra o seguinte:

- Uma tabela do DynamoDB que armazena dados de texto fornecidos pelo usuário.
- Os arquivos de configuração para criar a tabela.
- Um tópico do Amazon Simple Notification Service.
- O uso de um arquivo package.json para instalar pacotes durante a implantação.

Seções

- Pré-requisitos
- Criar um ambiente do Elastic Beanstalk
- Adicionar permissões às instâncias do seu ambiente
- Implante o aplicativo de exemplo
- Criar uma tabela do DynamoDB
- Atualizar os arquivos de configuração do aplicativo
- Configurar seu ambiente para alta disponibilidade
- Limpeza
- Próximas etapas

Pré-requisitos

Este tutorial requer os seguintes pré-requisitos:

- Os tempos de execução do Node.js
- O software gerenciador de pacotes Node.js padrão, npm
- O gerador de linha de comando Express
- A Elastic Beanstalk Command Line Interface (EB CLI)

Para obter detalhes sobre como instalar os três primeiros componentes listados e configurar seu ambiente de desenvolvimento local, consulte Configurar seu ambiente de desenvolvimento Node.js para o Elastic Beanstalk. Para este tutorial, você não precisa instalar o AWS SDK para Node.js, que também é mencionado no tópico referenciado.

Para obter detalhes sobre como instalar e configurar a EB CLI, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Criar um ambiente do Elastic Beanstalk

Seu diretório de aplicativos

Este tutorial usa um diretório chamado `nodejs-example-dynamo` para o pacote de origem do aplicativo. Crie o diretório `nodejs-example-dynamo` para este tutorial.

```
~$ mkdir nodejs-example-dynamo
```

> **Note**
>
> Cada tutorial neste capítulo usa seu próprio diretório para o pacote de origem do aplicativo. O nome do diretório corresponde ao nome do aplicativo de amostra usado pelo tutorial.

Altere seu diretório de trabalho atual para `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

Agora, vamos configurar um ambiente do Elastic Beanstalk executando a plataforma Node.js e o aplicativo de amostra. Usaremos a interface de linha de comando do Elastic Beanstalk (EB CLI).

Para configurar um repositório EB CLI para seu aplicativo e criar um ambiente Elastic Beanstalk executando a plataforma Node.js

1. Crie um repositório com o comando eb init.

   ```
   ~/nodejs-example-dynamo$ eb init --platform node.js --region <region>
   ```

   Esse comando cria um arquivo de configuração em uma pasta chamada `.elasticbeanstalk` que especifica as configurações para a criação de ambientes para a aplicação e cria uma aplicação do Elastic Beanstalk com nome baseado na pasta atual.
2. Crie um ambiente executando um aplicativo de exemplo com o comando eb create.

   ```
   ~/nodejs-example-dynamo$ eb create --sample nodejs-example-dynamo
   ```

   Esse comando cria um ambiente com balanceamento de carga com as configurações padrão da plataforma do Node.js e os seguintes recursos:

   - EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

     Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

3. Quando a criação do ambiente for concluída, use o comando eb open para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-dynamo$ eb open
```

Agora você criou um ambiente Node.js Elastic Beanstalk com um aplicativo de amostra. Você pode atualizá-lo com seu próprio aplicativo. Em seguida, atualizamos o aplicativo de exemplo para usar a framework do Express.

Adicionar permissões às instâncias do seu ambiente

Seu aplicativo é executado em uma ou mais EC2 instâncias por trás de um balanceador de carga, atendendo a solicitações HTTP da Internet. Quando recebe uma solicitação que exige o uso de AWS serviços, o aplicativo usa as permissões da instância em que é executado para acessar esses serviços.

O aplicativo de amostra usa permissões de instância para gravar dados em uma tabela do DynamoDB e enviar notificações para um tópico do Amazon SNS com o SDK em Node.js. JavaScript Adicione as seguintes políticas gerenciadas ao perfil de instância padrão para conceder às EC2 instâncias em seu ambiente permissão para acessar o DynamoDB e o Amazon SNS:

- AmazonDynamoDBFullAcesso
- Amazon SNSFull Access

Para adicionar políticas ao perfil da instância padrão

1. Abra a página Roles (Funções) no console do IAM.
2. Escolha aws-elasticbeanstalk-ec2 funções.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Selecione a política gerenciada para os serviços adicionais que o seu aplicativo utiliza. Para este tutorial, selecione `AmazonSNSFullAccess` e `AmazonDynamoDBFullAccess`.
5. Escolha Anexar política.

Consulte Gerenciar perfis de instância do Elastic Beanstalk para saber mais sobre o gerenciamento de perfis de instância.

Implante o aplicativo de exemplo

Agora seu ambiente está pronto para você implantar e executar o aplicativo de exemplo deste tutorial: nodejs-example-dynamo.zip.

Para implantar e executar o aplicativo de exemplo do tutorial

1. Altere seu diretório de trabalho atual para o diretório do aplicativo `nodejs-example-dynamo`.

```
~$ cd nodejs-example-dynamo
```

2. Baixe e extraia o conteúdo do pacote de origem do aplicativo de exemplo nodejs-example-dynamo.zip para o diretório do aplicativo. `nodejs-example-dynamo`
3. Implante o aplicativo de exemplo em seu ambiente Elastic Beanstalk com o comando eb deploy.

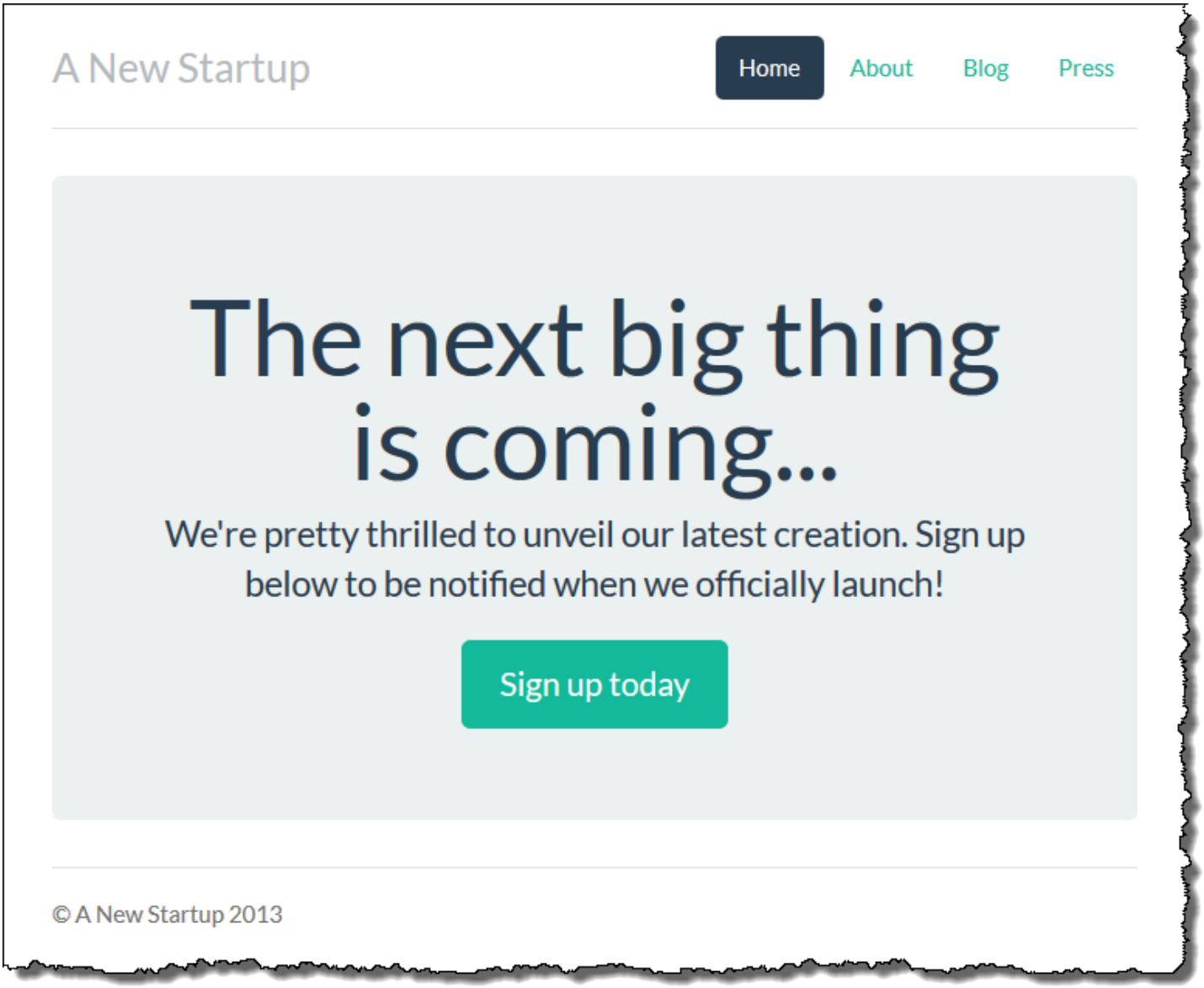
```
~/nodejs-example-dynamo$ eb deploy
```

> **Note**
>
> Por padrão, o comando `eb deploy` cria um arquivo ZIP da pasta do seu projeto. Você pode configurar a EB CLI para implantar um artefato do seu processo de compilação, em vez de criar um arquivo ZIP da pasta do projeto. Para obter mais informações, consulte Implantar um artefato, em vez da pasta do projeto.

4. Quando a criação do ambiente for concluída, use o comando eb open para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-dynamo$ eb open
```

O site coleta informações de contato do usuário e usa uma tabela do DynamoDB para armazenar os dados. Para adicionar uma entrada, escolha Sign up today, insira um nome e um endereço de e-mail e, em seguida, selecione Sign Up!. A aplicação Web grava o conteúdo do formulário na tabela e aciona uma notificação por e-mail do Amazon SNS.



No momento, o tópico do Amazon SNS está configurado com um espaço reservado para notificações de e-mail. Você atualizará a configuração em breve, mas enquanto isso, pode verificar a tabela do DynamoDB e o tópico do Amazon SNS no Console de Gerenciamento da AWS Management Console.

Para visualizar a tabela

1. Abra a página Tables (Tabelas) no console do DynamoDB.
2. Encontre a tabela que o aplicativo criou. O nome começa com awseb e contém. StartupSignupsTable
3. Selecione a tabela, escolha Items e Start search para visualizar todos os itens na tabela.

A tabela contém uma entrada para cada endereço de e-mail enviado no site de inscrição. Além de gravar na tabela, a aplicação envia uma mensagem para um tópico do Amazon SNS que tem duas assinaturas, uma para notificações por e-mail para você e outra para uma fila do Amazon Simple Queue Service que uma aplicação de operador pode ler para processar solicitações e enviar e-mails para os clientes interessados.

Para visualizar o tópico

1. Abra a página Topics (Tópicos) no console do Amazon SNS.
2. Encontre o tópico que o aplicativo criou. O nome começa com awseb e contém. NewSignupTopic
3. Escolha o tópico para visualizar as assinaturas.

O aplicativo (`app.js`) define duas rotas. O caminho raiz (`/`) retorna uma página da Web renderizada a partir de um modelo Embedded JavaScript (EJS) com um formulário que o usuário preenche para registrar seu nome e endereço de e-mail. A submissão do formulário envia uma solicitação POST com os dados do formulário para a rota `/signup`, que grava uma entrada na tabela do DynamoDB e publica uma mensagem para o tópico do Amazon SNS para notificar o proprietário da inscrição.

A aplicação demonstrativa inclui os arquivos de configuração que criam a tabela do DynamoDB, o tópico do Amazon SNS e a fila do Amazon SQS usados pela aplicação. Isso permite que você crie um novo ambiente e teste a funcionalidade imediatamente, mas tem a desvantagem de vincular a tabela do DynamoDB ao ambiente. Para um ambiente de produção, você deve criar a tabela do DynamoDB fora do seu ambiente para evitar perdê-lo quando encerrar o ambiente ou atualizar sua configuração.

Criar uma tabela do DynamoDB

Para usar uma tabela do DynamoDB externa com uma aplicação em execução no Elastic Beanstalk primeiro crie uma tabela no DynamoDB. Quando uma instância é criada fora do Elastic Beanstalk, ela é completamente independente do Elastic Beanstalk e dos ambientes do Elastic Beanstalk e não será encerrada pelo Elastic Beanstalk.

Crie uma tabela com as seguintes configurações:

- Table name (Nome da tabela): **`nodejs-tutorial`**
- chave primária: **`email`**
- Tipo de chave primária: string

Como criar uma tabela do DynamoDB

1. Abra a página Tables (Tabelas) no console de gerenciamento do DynamoDB.
2. Escolha Create table.
3. Digite um nome de tabela e chave primária.
4. Escolha o tipo de chave primária.
5. Escolha Criar.

Atualizar os arquivos de configuração do aplicativo

Atualize os arquivos de configuração na origem do aplicativo para usar a tabela nodejs-tutorial em vez de criar uma nova.

Para atualizar o aplicativo de exemplo para uso em produção

1. Altere seu diretório de trabalho atual para o diretório do aplicativo `nodejs-example-dynamo`.

   ```
   ~$ cd nodejs-example-dynamo
   ```

2. Abra `.ebextensions/options.config` e altere os valores das configurações a seguir:

   - NewSignupEmail— Seu endereço de e-mail.
   - 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"
```

Isso aplica as seguintes configurações para o aplicativo:

- O endereço de e-mail que o tópico do Amazon SNS usa para notificações é definido como seu endereço ou aquele que você insere no arquivo `options.config`.
- A tabela nodejs-tutorial será usada em vez da criada por `.ebextensions/create-dynamodb-table.config`.

3. Remova `.ebextensions/create-dynamodb-table.config`.

```
~/nodejs-tutorial$ rm .ebextensions/create-dynamodb-table.config
```

Na próxima vez que você implantar o aplicativo, a tabela criada por esse arquivo de configuração será excluída.

4. Implante o aplicativo atualizado em seu ambiente Elastic Beanstalk com o comando eb deploy.

```
~/nodejs-example-dynamo$ eb deploy
```

5. Quando a criação do ambiente for concluída, use o comando eb open para abrir o URL do ambiente no navegador padrão.

```
~/nodejs-example-dynamo$ eb open
```

Quando você implanta, o Elastic Beanstalk atualiza a configuração do tópico do Amazon SNS e exclui a tabela do DynamoDB que ele criou quando a primeira versão da aplicação foi implantada.

Agora, quando você encerrar o ambiente, a tabela nodejs-tutorial não será excluída. Isso permite que você realize implantações azul/verde, modifique os arquivos de configuração ou desative seu site sem o risco de perda de dados.

Abra seu site em um navegador e verifique se o formulário funciona como esperado. Crie alguma entradas e, depois, verifique o console do DynamoDB para verificar a tabela.

Para visualizar a tabela

1. Abra a página Tables (Tabelas) no console do DynamoDB.
2. Encontre a tabela nodejs-tutorial.
3. Selecione a tabela, escolha Items e Start search para visualizar todos os itens na tabela.

Você também pode ver que o Elastic Beanstalk excluiu a tabela que foi criada anteriormente.

Configurar seu ambiente para alta disponibilidade

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias o tempo todo para impedir que os servidores Web em seu ambiente sejam um ponto único de falha e para permitir que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Você também pode excluir as tabelas externas do DynamoDB que você criou.

Como excluir uma tabela do DynamoDB

1. Abra a página Tables (Tabelas) no console do DynamoDB.
2. Selecione uma tabela.
3. Escolha Actions e, em seguida, escolha Delete table.
4. Escolha Excluir.

Próximas etapas

O aplicativo de exemplo usa arquivos de configuração para definir as configurações do software e criar AWS recursos como parte do seu ambiente. Consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`) para obter mais informações sobre os arquivos de configuração e seu uso.

O aplicativo de exemplo para este tutorial usa a framework da Web Express para Node.js. Para obter mais informações sobre o Express, consulte a documentação oficial em expressjs.com.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite o HTTPS para conexões seguras.

Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente do Node.js Elastic Beanstalk

Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como

desacoplado e gerenciado externamente por outro serviço. Nestas instruções, o banco de dados é acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Seções

- Adicionar uma instância de banco de dados ao seu ambiente
- Download de um driver
- Conectar-se a um banco de dados

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão	Na guia Connectivity & security (Conectividade e

Nome da propriedade	Descrição	Valor da propriedade
	varia entre os mecanismos de banco de dados.	segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **`ebdb`**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
`RDS_PASSWORD`	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Download de um driver

Adicione o driver de banco de dados ao arquivo `package.json` do projeto em `dependencies`.

Example **`package.json`**: Express com 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"
  }
}
```

Pacotes de drivers comuns para Node.js

- MySQL – mysql
- PostgreSQL – node-postgres
- SQL Server – node-mssql
- Oracle – node-oracledb

Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `process.env.`*`VARIABLE`* para ler as propriedades e configurar uma conexão de banco de dados.

Example app.js: conexão de banco de dados 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();
```

Para obter mais informações sobre como construir uma string de conexão usando node-mysql, consulte npmjs. org/package/mysql.

Ferramentas e recursos Node.js

Há vários lugares que você pode acessar para obter ajuda adicional durante o desenvolvimento de seus aplicativos Node.js:

Recurso	Descrição
GitHub	Instale o AWS SDK para Node.js usando o. GitHub
AWS SDK para Node.js (prévia para desenvolvedores)	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.

Implantação de aplicações PHP com o Elastic Beanstalk

Você pode implantar um aplicativo PHP em alguns minutos usando a interface de linha de comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Para saber como, comece com o QuickStart tutorial e, em seguida, revise os exemplos avançados.

Recursos para implantar aplicativos PHP no Elastic Beanstalk

- QuickStart para PHP— Step-by-step instruções para implantar um aplicativo PHP Hello World usando a CLI do EB.
- Usar a plataforma PHP do Elastic Beanstalk— Como usar os recursos da plataforma especificamente para PHP.
- Exemplos avançados — Tutoriais de como fazer estruturas e aplicativos PHP comuns, além de como adicionar um banco de dados Amazon RDS ao seu ambiente.

Para obter mais informações sobre desenvolvimento com PHP em AWS, consulte os seguintes recursos:

- GitHub— Instale o AWS SDK for PHP GitHub usando.
- Centro de desenvolvedores de PHP — Ferramentas, documentos e exemplos de código para desenvolver aplicativos PHP em AWS.
- AWS SDK for FAQs PHP — Obtenha respostas para as perguntas mais frequentes.

QuickStart: Implante um aplicativo PHP no Elastic Beanstalk

No tutorial a seguir, você aprenderá a criar e implantar um aplicativo PHP de amostra em um AWS Elastic Beanstalk ambiente usando a CLI do EB.

⚠ Aviso - Não é para uso em produção!

Os exemplos são destinados apenas para demonstração. Não use aplicativos de exemplo na produção.

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta para usar o Elastic Beanstalk.

Criar uma conta da AWS

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

- Interface de linha de comando do Elastic Beanstalk — Para instalação, consulte. Instale o EB CLI com script de configuração (recomendado)
- PHP - Instale o PHP em sua máquina local seguindo as instruções de instalação e configuração no site do PHP.

Etapa 1: criar uma aplicação PHP

Para esse início rápido, você criará um aplicativo PHP Hello World.

Crie um diretório do projeto.

```
~$ mkdir eb-php
~$ cd eb-php
```

Em seguida, crie um `index.php` arquivo no diretório do projeto e adicione o código a seguir.

Example **`index.php`**

```
<?php
  echo "Hello from a PHP application running in Elastic Beanstalk!";
?>
```

Etapa 2: executar a aplicação localmente

Use o comando a seguir para executar seu aplicativo localmente.

```
php -S localhost:5000
```

Abra um navegador para http://localhost:5000.

Você deve ver sua mensagem de saudação no navegador e as mensagens de registro em seu terminal.

Pare o servidor local entrando`Control+c`, para que você possa implantar o Elastic Beanstalk.

Etapa 3: inicializar e implantar seu aplicativo PHP

Em seguida, você implantará seu aplicativo em um ambiente usando o console do Elastic Beanstalk ou o EB CLI. Neste tutorial, você usará o EB CLI com a opção interativa para inicializar um ambiente.

Para inicializar seu ambiente e criar um ambiente

1. Execute o seguinte comando init.

   ```
   eb init -i
   ```

   O comando init cria um aplicativo interativamente. O nome do aplicativo será padronizado para a pasta local, que é `eb-php`.

   Para todos os prompts, exceto o acesso SSH, aceite os padrões para criar um ambiente com a versão mais recente da plataforma PHP. Para solucionar problemas de instâncias, você pode configurar o acesso SSH executando novamente o `eb init -i` comando posteriormente ou conectar-se usando o Amazon EC2 Instance Connect ou o Session Manager.
2. Crie um ambiente e implante seu aplicativo

   Execute o comando a seguir para criar um ambiente chamado `blue-env`.

   ```
   eb create blue-env
   ```

   Quando você executa o eb create comando pela primeira vez, o Elastic Beanstalk cria automaticamente um arquivo zip do seu aplicativo, chamado de pacote de origem. Em seguida, o Elastic Beanstalk cria um ambiente com uma ou mais instâncias EC2 da Amazon e, em seguida, implanta o aplicativo no ambiente.

   A implantação do seu aplicativo no Elastic Beanstalk pode levar até cinco minutos.

Etapa 4: Navegue pelo seu aplicativo em nuvem

Quando o processo de criação do ambiente for concluído, seu aplicativo deverá estar em execução e escutando as solicitações na porta 5000. Conecte-se ao seu aplicativo com o seguinte comando:

```
eb open
```

O `eb open` comando abre uma guia do navegador para um subdomínio personalizado criado para seu aplicativo.

Etapa 5: atualizar e reimplantar seu aplicativo

Depois de criar um aplicativo e implantá-lo em um ambiente, você pode implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é mais rápida porque não exige provisionamento ou reinicialização de instâncias da Amazon. EC2

Atualize seu código PHP para incluir o valor REQUEST_TIME do ambiente do servidor:

```
<?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);
?>
```

Reimplante seu código PHP no Elastic Beanstalk com o seguinte comando:

```
eb deploy
```

Quando você executa eb deploy, a EB CLI empacota o conteúdo do diretório do projeto e implanta-o em seu ambiente.

Após a conclusão da implantação, atualize a página ou reconecte-se ao seu aplicativo com. `eb open` Você deve ver suas atualizações. Caso contrário, solucione o problema executando seu servidor local novamente para verificar suas alterações.

> Parabéns!
>
> Você criou, implantou e atualizou um aplicativo PHP com o Elastic Beanstalk!

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Como alternativa, você pode encerrar seu ambiente com a CLI do EB com o seguinte comando:

```
eb terminate
```

Próximas etapas

Você pode explorar seu ambiente de aplicativos usando o console do Elastic Beanstalk. Para obter mais informações, consulte Explore seu ambiente.

Para exemplos avançados usando PHP, consulteExemplos avançados de PHP no Elastic Beanstalk.

Usar a plataforma PHP do Elastic Beanstalk

AWS Elastic Beanstalk fornece e suporta várias ramificações de plataforma para diferentes versões do PHP. As plataformas suportam aplicativos web PHP que são executados de forma independente ou no Composer. Consulte PHP no documento Plataformas do AWS Elastic Beanstalk para uma lista completa de ramificações de plataforma compatíveis.

O Elastic Beanstalk fornece opções de configuração que você pode usar para personalizar o software executado nas instâncias da EC2 Amazon em seu ambiente do Elastic Beanstalk. É possível configurar as variáveis de ambiente necessárias à sua aplicação, habilitar a alternância de log para o Amazon S3, mapear pastas na origem da aplicação que contenham arquivos estáticos para caminhos servidos pelo servidor de proxy e definir configurações de inicialização comuns do PHP.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente

quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

Se você usa o Composer, pode incluir um arquivo `composer.json` em seu pacote de origem para instalar pacotes durante a implantação.

Para configuração avançada de PHP e definições de PHP que não são fornecidos como opções de configuração, é possível usar arquivos de configuração para fornecer um arquivo `INI`que pode estender e substituir as configurações padrão aplicadas pelo Elastic Beanstalk ou instalar extensões adicionais.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Tópicos da plataforma PHP

- Instalando o AWS SDK for PHP
- Considerações sobre o PHP 8.1 no Amazon Linux 2
- Configurar seu ambiente PHP
- Namespaces para configuração
- Instalar dependências da aplicação PHP do Elastic Beanstalk
- Atualizar o Composer no Elastic Beanstalk
- Estender php.ini na sua configuração do Elastic Beanstalk

Instalando o AWS SDK for PHP

Se você precisar gerenciar AWS recursos de dentro do seu aplicativo, instale AWS SDK para PHP o. Por exemplo, com o SDK for PHP, é possível usar o Amazon DynamoDB (DynamoDB) para armazenar informações de usuário e sessão sem criar um banco de dados relacional.

Para instalar o SDK para PHP com o Composer

```
$ composer require aws/aws-sdk-php
```

Para obter mais informações, consulte a página inicial do AWS SDK for PHP. Para obter instruções, consulte Instalar AWS SDK for PHP o.

Considerações sobre o PHP 8.1 no Amazon Linux 2

Leia esta seção se você estiver usando a ramificação da plataforma PHP 8.1.

Considerações sobre o PHP 8.1 no Amazon Linux 2

> **Note**
>
> As informações neste tópico somente são aplicáveis a ramificações de plataforma com base na Amazon Linux AMI (AL1). Ela não se aplica às ramificações da plataforma PHP com base em AL2 023. Também não se aplica aoPHP 8.0 Amazon Linux 2filial da plataforma.

O Elastic Beanstalk armazena os pacotes RPM relacionados ao PHP 8.1 para a ramificação da plataforma PHP 8.1 no Amazon Linux 2 nas instâncias EC2 em um diretório local, em vez do repositório Amazon Linux. Você pode usar rpm -i para instalar os pacotes necessários. A partir da versão 3.5.0 da plataforma PHP 8.1, o Elastic Beanstalk armazena os pacotes RPM relacionados ao PHP 8.1 no seguinte diretório local. EC2

`/opt/elasticbeanstalk/RPMS`

O exemplo a seguir instala o pacote php-debuginfo.

```
$rpm -i /opt/elasticbeanstalk/RPMS/php-debuginfo-8.1.8-1.amzn2.x86_64.rpm
```

A versão no nome do pacote variará de acordo com a versão real listada no diretório EC2 local/opt/elasticbeanstalk/RPMS. Use a mesma sintaxe para instalar outros pacotes RPM da plataforma PHP 8.1.

Expanda a seção a seguir para exibir uma lista dos pacotes RPM que fornecemos.

Pacotes RPM

A lista a seguir fornece os pacotes RMP que a plataforma PHP 8.1 do Elastic Beanstalk fornece no Amazon Linux 2. Eles ficam localizados no diretório local /opt/elasticbeanstalk/RPMS.

Os números de versão 8.1.8-1 e 3.7.0-1 nos nomes dos pacotes listados são apenas um exemplo.

- 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

Você pode usar os pacotes PEAR e PECL para instalar extensões comuns. Para obter mais informações sobre o PEAR, consulte o site PEAR PHP Extension and Application Repository (Repositório de extensão e aplicação do PEAR PHP). Para obter mais informações sobre o PECL, consulte o site de extensões PECL.

Os exemplos de comandos a seguir instalam as extensões Memcached.

```
$pecl install memcache
```

Ou você também pode usar o seguinte:

```
$pear install pecl/memcache
```

Os exemplos de comando a seguir instalam as extensões Redis.

```
$pecl install redis
```

Ou você também pode usar o seguinte:

```
$pear install pecl/redis
```

Configurar seu ambiente PHP

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3, configurar variáveis que a aplicação pode ler no ambiente e alterar as configurações do PHP.

Como configurar seu ambiente PHP no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Configurações de PHP

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, é usado nginx.
- Document root (Raiz do documento): a pasta que contém a página padrão do site. Se a sua página de boas-vindas não estiver na raiz do pacote de origem, especifique a pasta que a contém relativo ao caminho raiz. Por exemplo, `/public` se a página de boas-vindas estiver em uma pasta chamada `public`.
- Memory limite (Limite de memória): a quantidade máxima de memória que um script tem permissão para alocar. Por exemplo, `.512M`
- Zlib output compression (Compactação de saída Zlib): defina como `On` para compactar respostas.
- Allow URL fopen (Permitir URL fopen): defina como `Off` para impedir o download de arquivos de scripts de locais remotos.
- Display errors (Exibir erros): defina como `On` para mostrar mensagens de erro interno para depuração.
- Max execution time (runtime máximo): o tempo máximo em segundos que um script pode ser executado antes que o ambiente o encerre.

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Propriedades de ambiente

A seção Propriedades do ambiente permite que você especifique as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. Essas definições são passadas para o aplicativo como pares de chave-valor.

Seu código de aplicativo pode acessar propriedades de ambiente usando `$_SERVER` ou a função `get_cfg_var`.

```
$endpoint = $_SERVER['API_ENDPOINT'];
```

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespaces para configuração

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

Os namespaces a seguir configuram seu serviço de proxy e opções específicas do PHP:

- `aws:elasticbeanstalk:environment:proxy:staticfiles`: configure o proxy do ambiente para servir arquivos estáticos. Defina mapeamentos de caminhos virtuais para diretórios de aplicativos.
- `aws:elasticbeanstalk:environment:proxy`: especifique o servidor proxy do ambiente.

- `aws:elasticbeanstalk:container:php:phpini`: configure opções específicas do PHP. Este namespace inclui `composer_options`, que não está disponível no console do Elastic Beanstalk. Essa opção define as opções personalizadas a serem usadas ao instalar dependências usando o Composer por meio do `composer.phar install` comando. Para obter mais informações sobre esse comando, incluindo as opções disponíveis, consulte instalar no site getcomposer.org.

O exemplo de arquivo de configuração a seguir especifica uma opção de arquivos estáticos que mapeia um diretório chamado `staticimages` para o caminho `/images`, e mostra as configurações para cada uma das opções disponíveis no namespace `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**
>
> O namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` não está definido nas ramificações da plataforma PHP de AMI do Amazon Linux (anterior ao Amazon Linux 2).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Instalar dependências da aplicação PHP do Elastic Beanstalk

Este tópico descreve como configurar a aplicação para instalar outros pacotes PHP necessários. Seu aplicativo pode ter dependências em outros pacotes PHP. Você pode configurar seu aplicativo para instalar essas dependências nas instâncias do Amazon Elastic Compute Cloud EC2 (Amazon) do ambiente. Como alternativa, você pode incluir as dependências do aplicativo no pacote de origem e implantá-las com o aplicativo. A seção a seguir discute cada uma dessas maneiras.

Usar um arquivo do Composer para instalar dependências em instâncias

Use um `composer.json` arquivo na raiz da fonte do seu projeto para usar o composer para instalar os pacotes que seu aplicativo exige nas EC2 instâncias da Amazon do seu ambiente.

Example composer.json

```
{
    "require": {
        "monolog/monolog": "1.0.*"
    }
}
```

Quando um arquivo `composer.json` está presente, o Elastic Beanstalk executa `composer.phar install` para instalar dependências. Você pode adicionar opções para acrescentar ao comando definindo a opção `composer_options` no namespace `aws:elasticbeanstalk:container:php:phpini`.

Incluir dependências no pacote de origem

Se o seu aplicativo tiver um grande número de dependências, sua instalação poderá demorar muito tempo. Isso pode aumentar a implantação e as operações de escalabilidade, porque as dependências são instaladas em cada nova instância.

Para evitar o impacto negativo no momento da implantação, use o Composer em seu ambiente de desenvolvimento para resolver dependências e instalá-las na pasta `vendor`.

Para incluir dependências em seu pacote de origem do aplicativo

1. Execute o seguinte comando:

   ```
   % composer install
   ```

2. Inclua a pasta `vendor` gerada na raiz do seu pacote de origem do aplicativo.

Quando o Elastic Beanstalk encontra uma pasta `vendor` na instância, ele ignora o arquivo `composer.json` (mesmo se ele existir). O aplicativo usa as dependências da pasta `vendor`.

Atualizar o Composer no Elastic Beanstalk

Este tópico descreve como configurar o Elastic Beanstalk para manter o Composer atualizado. Talvez seja necessário atualizar o Composer se você vir um erro ao tentar instalar pacotes com um arquivo do Composer ou se não conseguir usar a versão mais recente da plataforma. Entre as atualizações da plataforma, você pode atualizar o Composer em suas instâncias de ambiente por meio do uso de arquivos de configuração na pasta `.ebextensions`.

É possível atualizar o Composer com a seguinte configuração.

```
commands:
  01updateComposer:
    command: /usr/bin/composer.phar self-update 2.7.0
```

A configuração de opção a seguir define a variável de ambiente `COMPOSER_HOME`, que configura o local do cache do Composer.

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environment
    option_name: COMPOSER_HOME
    value: /home/webapp/composer-home
```

É possível combinar ambos no mesmo arquivo de configuração da pasta `.ebextensions`.

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**
>
> Devido às atualizações na instalação do Composer na versão da plataforma de 22 de fevereiro de 2024 e na versão de 28 de fevereiro de 2024, AL2 a atualização automática do Composer pode falhar se for definida quando a atualização automática `COMPOSER_HOME` for executada. AL2
> Os seguintes comandos combinados não serão executados: `export COMPOSER_HOME=/home/webapp/composer-home && /usr/bin/composer.phar self-update 2.7.0` Entretanto, o exemplo anterior funcionará. No exemplo anterior, a configuração da opção para `COMPOSER_HOME` não será passado para a execução do `01updateComposer` e não será definido quando o comando de autoatualização for executado.

> **Important**
>
> Se você omitir o número da versão do comando `composer.phar self-update`, o Composer será atualizado para a versão mais recente disponível a cada vez que você implantar seu código-fonte e quando novas instâncias forem provisionadas pelo Auto Scaling. Isso pode fazer com que as operações de escalabilidade e as implantações falhem se for lançada uma versão do Composer que não seja compatível com o seu aplicativo.

Para obter mais informações sobre as Plataformas PHP do Elastic Beanstalk, incluindo a versão do Composer, consulte Versões da plataformas PHP no documento AWS Elastic Beanstalk Platforms (Plataformas do Elastic Beanstalk.

Estender php.ini na sua configuração do Elastic Beanstalk

Use um arquivo de configuração com um bloco `files` para adicionar um arquivo `.ini` em `/etc/php.d/` nas instâncias em seu ambiente. O arquivo de configuração principal, `php.ini`, obtém configurações de arquivos nesta pasta em ordem alfabética. Muitas extensões são habilitadas por padrão pelos arquivos nessa pasta.

Example .ebextensions/mongo.config

```
files:
  "/etc/php.d/99mongo.ini":
    mode: "000755"
```

```
    owner: root
    group: root
    content: |
      extension=mongo.so
```

Exemplos avançados de PHP no Elastic Beanstalk

Para começar a usar aplicativos PHP AWS Elastic Beanstalk, você precisa fazer o upload de um pacote de origem do aplicativo como sua primeira versão do aplicativo a ser implantada em um ambiente.

Recomendamos que você comece com um aplicativo PHP simples implantado com o EB CLI. QuickStart para PHP

Exemplos avançados de PHP

- Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente PHP Elastic Beanstalk
- Implantar uma aplicação Laravel no Elastic Beanstalk
- Implantar uma aplicação CakePHP no Elastic Beanstalk
- Implantar uma aplicação Symfony no Elastic Beanstalk
- Implantar uma aplicação PHP de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk
- Implantação de um WordPress site de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk
- Implantar um site Drupal de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente PHP Elastic Beanstalk

Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Nestas instruções, o banco de dados é

acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Seções

- Adicionar uma instância de banco de dados ao seu ambiente
- Download de um driver
- Conectando-se a um banco de dados com um PDO ou My SQLi
- Conexão a um banco de dados com Symfony

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão	Na guia Connectivity & security (Conectividade e

Nome da propriedade	Descrição	Valor da propriedade
	varia entre os mecanismos de banco de dados.	segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
`RDS_PASSWORD`	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Download de um driver

Para usar Objetos de dados PHP (PDO) para se conectar ao banco de dados, instale o driver que corresponde ao mecanismo de banco de dados que você escolheu.

- MySQL – `PDO_MYSQL`
- PostgreSQL – `PDO_PGSQL`
- Oracle: `PDO_OCI`
- SQL Server – `PDO_SQLSRV`

Para obter mais informações, consulte http://php.net/manual/en/pdo.installation.php.

Conectando-se a um banco de dados com um PDO ou My SQLi

Você pode usar `$_SERVER[`*VARIABLE*`]` para ler as informações de conexão do ambiente.

Para obter um PDO, crie um Nome da Fonte de Dados (DSN) do host, porta e nome. Passe o DSN para o construtor do PDO com o nome de usuário e senha do banco de dados.

Example Conexão com banco de dados RDS com 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);
?>
```

Para outros drivers, substitua `mysql` pelo nome do driver: `pgsql`, `oci` ou `sqlsrv`.

Para MySQLi, passe o nome do host, o nome do usuário, a senha, o nome do banco de dados e a porta para o `mysqli` construtor.

Example Conexão com banco de dados RDS com mysqli_connect()

```
$link = new mysqli($_SERVER['RDS_HOSTNAME'], $_SERVER['RDS_USERNAME'],
 $_SERVER['RDS_PASSWORD'], $_SERVER['RDS_DB_NAME'], $_SERVER['RDS_PORT']);
```

Conexão a um banco de dados com Symfony

Para o Symfony versão 3.2 e mais recente, é possível usar `%env(`*`PROPERTY_NAME`*`)%` para definir os parâmetros de banco de dados em um arquivo de configuração com base nas propriedades do ambiente definidas pelo 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)%'
```

Consulte Parâmetros externos (Symfony 3.4) para obter mais informações.

Para versões anteriores do Symfony, as variáveis de ambiente estão acessíveis somente se elas começarem com `SYMFONY__`. Isso significa que as propriedades do ambiente definidas pelo Elastic Beanstalk não podem ser acessadas, e você deve definir suas próprias propriedades do ambiente para passar as informações de conexão para o Symfony.

Para se conectar a um banco de dados com o Symfony 2, Crie uma propriedade de ambiente para cada parâmetro. Em seguida, use `%property.name%` para acessar a variável transformada por Symfony em um arquivo de configuração. Por exemplo, uma propriedade de ambiente chamada `SYMFONY__DATABASE__USER` é acessível como `database.user`.

```
    database_user:     "%database.user%"
```

Consulte Parâmetros externos (Symfony 2.8) para obter mais informações.

Implantar uma aplicação Laravel no Elastic Beanstalk

O Laravel é uma estrutura de código aberto model-view-controller (MVC) para PHP. Este tutorial mostra o processo de geração de um aplicativo Laravel, sua implantação em um AWS Elastic Beanstalk ambiente e sua configuração para se conectar a uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS).

Seções

- Pré-requisitos
- Iniciar um ambiente do Elastic Beanstalk
- Instalar o Laravel e gerar um site
- Implantar o aplicativo
- Definir as configurações do Composer
- Adicionar um banco de dados ao seu ambiente
- Limpeza
- Próximas etapas

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

O Laravel 6 requer PHP 7.2 ou posterior. Ele também requer as extensões PHP listadas no tópico requisitos do servidor na documentação oficial do Laravel. Siga as instruções para instalar o PHP e o Composer.

Para obter informações sobre suporte e manutenção do Laravel, consulte o tópico política de suporte na documentação oficial do Laravel.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.

5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Instalar o Laravel e gerar um site

O Composer pode instalar o Laravel e criar um projeto de trabalho com um comando:

```
~$ composer create-project --prefer-dist laravel/laravel eb-laravel
```

O Composer instala o Laravel e suas dependências, e gera um projeto padrão.

Se você tiver algum problema ao instalar o Laravel, acesse o tópico de instalação na documentação oficial: https://laravel.com/docs/ 6.x.

Implantar o aplicativo

Crie um pacote de origem com os arquivos criados pelo Composer. O comando a seguir cria um pacote de origem chamado `laravel-default.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
~/eb-laravel$ zip ../laravel-default.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Laravel no ambiente.

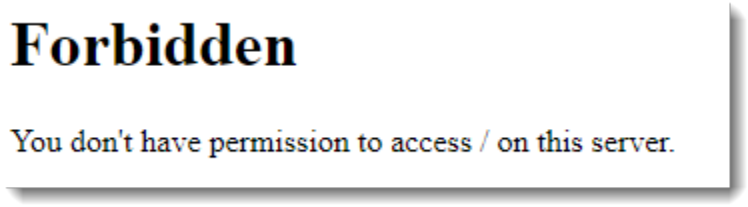
Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

> **Note**
>
> Para otimizar mais o pacote de origem, inicie um repositório Git e use o comando `git archive`para criar o pacote de origem. O projeto do Laravel padrão inclui um arquivo `.gitignore` que diz para o Git excluir a pasta `vendor` e outros arquivos que não são necessários para implantação.

Definir as configurações do Composer

Quando a implantação for concluída, clique no URL para abrir o aplicativo Laravel no navegador:



O que é isso? Por padrão, o Elastic Beanstalk fornece a raiz do seu projeto no caminho raiz do site. Neste caso, contudo, a página padrão (`index.php`) está em um nível abaixo na pasta `public`. Você pode verificar isso adicionando `/public` ao URL. Por exemplo, .`http://laravel.us-east-2.elasticbeanstalk.com/public`

Para fornecer a aplicação Laravel no caminho raiz, use o console do Elastic Beanstalk para configurar a raiz do documento para o site.

Para configurar a raiz do documento do seu site

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Para Document Root (Raiz do documento), insira **`/public`**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
7. Quando a atualização for concluída, clique no URL para reabrir seu site no navegador.



Até agora, tudo tranquilo. Em seguida, adicione um banco de dados ao ambiente e configure o Laravel para se conectar a ele.

Adicionar um banco de dados ao seu ambiente

Execute uma instância de banco de dados do RDS no ambiente do Elastic Beanstalk. Você pode usar bancos de dados MySQL ou PostgreSQL com o Laravel no Elastic Beanstalk. SQLServer Neste exemplo, usaremos MySQL.

Como adicionar uma instância de banco de dados do RDS ao ambiente do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Em Engine (Mecanismo), escolha mysql.
6. Digite um username (nome de usuário) e uma password (senha) principais. O Elastic Beanstalk fornecerá esses valores para a aplicação usando propriedades de ambiente.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Para obter mais informações sobre bancos de dados acoplados a um ambiente do Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Enquanto isso, atualize o código-fonte para ler as informações de conexão do ambiente. O Elastic Beanstalk fornece detalhes de conexão usando variáveis de ambiente, como `RDS_HOSTNAME`, que você pode acessar na aplicação.

A configuração de banco de dados do Laravel é armazenada em um arquivo chamado `database.php` na pasta `config` no código do projeto. Encontre a entrada `mysql` e modifique as variáveis de `host`, `database`, `username`, and `password` para ler os valores correspondentes do 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,
        ],
...
```

Para verificar se a conexão de banco de dados está configurada corretamente, adicione código a `index.php` para conectar ao banco de dados e adicionar código à resposta padrão:

Example ~/Eb- .php laravel/public/index

```
...
if(DB::connection()->getDatabaseName())
{
   echo "Connected to database ".DB::connection()->getDatabaseName();
}
$response->send();
...
```

Quando a instância de banco de dados tiver iniciado, empacote e implante o aplicativo atualizado em seu ambiente.

Como atualizar o ambiente do Elastic Beanstalk

1. Crie um novo pacote de origem:

   ```
   ~/eb-laravel$ zip ../laravel-v2-rds.zip -r * .[^.]* -x "vendor/*"
   ```

2. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
4. Escolha Upload and Deploy.
5. Escolha Browse e faça upload de `laravel-v2-rds.zip`.
6. Escolha Implantar.

Implantar uma nova versão do seu aplicativo leva menos de um minuto. Quando a implantação for concluída, atualize a página da web novamente para verificar se a conexão com o banco de dados foi bem-sucedida:



Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

Para obter mais informações sobre o Laravel, acesse o site oficial do Laravel em laravel.com.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Neste tutorial, você usou o console do Elastic Beanstalk para configurar as opções do Composer. Para tornar essa configuração parte de origem do aplicativo, você pode usar um arquivo de configuração, como o seguinte.

Example .ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Para obter mais informações, consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`).

A execução de uma instância de banco de dados do Amazon RDS no ambiente do Elastic Beanstalk é ideal para desenvolvimento e teste, mas vincula o ciclo de vida do banco de dados ao ambiente. Para obter instruções sobre como se conectar a um banco de dados em execução fora de seu ambiente, consulte Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente PHP Elastic Beanstalk.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantar uma aplicação CakePHP no Elastic Beanstalk

CakePHP é uma framework MVC de código aberto para PHP. Este tutorial orienta você no processo de gerar um projeto CakePHP, implantá-lo em um ambiente do Elastic Beanstalk e configurá-lo para se conectar a uma instância de banco de dados do Amazon RDS.

Seções

- Pré-requisitos

- Iniciar um ambiente do Elastic Beanstalk
- Instalar o CakePHP e gerar um site
- Implantar o aplicativo
- Adicionar um banco de dados ao seu ambiente
- Limpeza
- Próximas etapas

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

O CakePHP 4 requer PHP 7.4 ou posterior. Ele também requer as extensões PHP listadas na documentação oficial de instalação do CakePHP. Você deve instalar o PHP e o Composer.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced

2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.

- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Instalar o CakePHP e gerar um site

O Composer pode instalar o CakePHP e criar um projeto de trabalho com um comando:

```
~$ composer create-project --prefer-dist cakephp/app eb-cake
```

O Composer instala o CakePHP e cerca de 20 dependências e gera um projeto padrão.

Se tiver problemas ao instalar o CakePHP, visite o tópico de instalação na documentação oficial: http://book.cakephp.org/4.0/en/installation.html

Implantar o aplicativo

Crie um pacote de origem com os arquivos criados pelo Composer. O comando a seguir cria um pacote de origem chamado `cake-default.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
eb-cake zip ../cake-default.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o CakePHP no ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

> **Note**
>
> Para otimizar mais o pacote de origem, inicie um repositório Git e use o comando `git archive`para criar o pacote de origem. O projeto do Symfony padrão inclui um arquivo `.gitignore` que diz para o Git excluir a pasta `vendor` e outros arquivos que não são necessários para implantação.

Quando o processo for concluído, clique na URL para abrir a aplicação CakePHP no navegador:

Até agora, tudo tranquilo. Em seguida, adicione um banco de dados ao ambiente e configure o CakePHP para se conectar a ele.

Adicionar um banco de dados ao seu ambiente

Execute uma instância de banco de dados do Amazon RDS no ambiente do Elastic Beanstalk. Você pode usar bancos de dados MySQL ou PostgreSQL com o CakePHP no Elastic Beanstalk. SQLServer Neste exemplo, usaremos o PostgreSQL.

Como adicionar uma instância de banco de dados do Amazon RDS ao ambiente do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Em Database (Banco de dados), escolha Edit (Editar).
5. Em DB engine, escolha postgres.
6. Digite um username (nome de usuário) e uma password (senha) principais. O Elastic Beanstalk fornecerá esses valores para a aplicação usando propriedades de ambiente.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, atualize o código-fonte para ler as informações de conexão do ambiente. O Elastic Beanstalk fornece detalhes de conexão usando variáveis de ambiente, como `RDS_HOSTNAME`, que você pode acessar na aplicação.

A configuração de banco de dados do CakePHP é armazenada em um arquivo chamado `app.php` na pasta `config` no código do projeto. Abra este arquivo e adicione um código que leia as variáveis de ambiente do `$_SERVER` e as atribua às variáveis locais. Insira as linhas realçadas no exemplo abaixo após a primeira linha (`<?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 [
...
```

A conexão do banco de dados está configurada mais abaixo no arquivo `app.php`. Encontre a seção a seguir e modifique a configuração de fontes de dados padrão com o nome do driver que corresponde ao seu mecanismo de banco de dados (`Mysql`, `Sqlserver` ou `Postgres`) e defina as variáveis `host`, `username`, `password` e `database` para ler os valores correspondentes do Elastic Beanstalk:

Example ~/Eb- .php cake/config/app

```
...
     /**
     * 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 a instância de banco de dados tiver iniciado, crie um pacote e implante o aplicativo atualizado no ambiente:

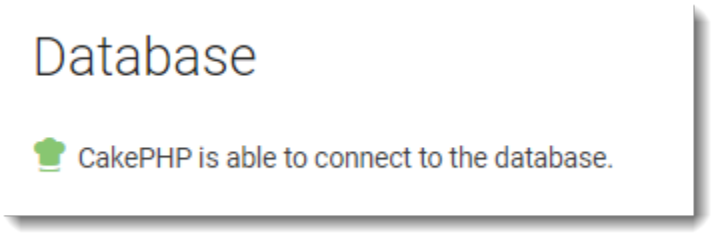
Como atualizar o ambiente do Elastic Beanstalk

1. Crie um novo pacote de origem:

```
~/eb-cake$ zip ../cake-v2-rds.zip -r * .[^.]* -x "vendor/*"
```

2. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
4. Escolha Upload and Deploy.
5. Escolha Browse e faça upload de `cake-v2-rds.zip`.
6. Escolha Implantar.

Implantar uma nova versão do seu aplicativo leva menos de um minuto. Quando a implantação for concluída, atualize a página da web novamente para verificar se a conexão com o banco de dados foi bem-sucedida:



Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

Para obter mais informações sobre o CakePHP, leia o livro em book.cakephp.org.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

A execução de uma instância de banco de dados do Amazon RDS no ambiente do Elastic Beanstalk é ideal para desenvolvimento e teste, mas vincula o ciclo de vida do banco de dados ao ambiente. Para obter instruções sobre como se conectar a um banco de dados em execução fora de seu ambiente, consulte Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente PHP Elastic Beanstalk.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantar uma aplicação Symfony no Elastic Beanstalk

O Symfony é uma framework de código aberto para o desenvolvimento de aplicações web de PHP dinâmico. Este tutorial mostra o processo de geração de um aplicativo Symfony e sua implantação em um ambiente. AWS Elastic Beanstalk

Seções

- Pré-requisitos
- Iniciar um ambiente do Elastic Beanstalk
- Instalar o Symfony e gerar um site
- Implantar o aplicativo
- Definir as configurações do Composer
- Limpeza
- Próximas etapas

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

O Symfony 4.4.9 requer PHP 7.1.3 ou posterior. Ele também requer as extensões PHP listadas no tópico requisitos técnicos na documentação de instalação oficial do Symfony. Neste tutorial, usamos o PHP 7.2 e a versão correspondente da plataforma do Elastic Beanstalk. Antes de continuar, você deve instalar o PHP e o Composer.

Para obter informações sobre suporte e manutenção do Symfony, consulte o tópico lançamentos do symfony no site do Symfony. Para obter mais informações sobre atualizações relacionadas ao suporte à versão do PHP para o Symfony 4.4.9, consulte o tópico notas de release do Symfony 4.4.9 no site do Symfony.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.

- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Instalar o Symfony e gerar um site

O Composer pode instalar o Symfony e criar um projeto de trabalho com um comando:

```
~$ composer create-project symfony/website-skeleton eb-symfony
```

O Composer instala o Symfony e suas dependências e gera um projeto padrão.

Se tiver problemas com a instalação do Symfony, consulte o tópico instalação na documentação oficial do Symfony.

Implantar o aplicativo

Acesse o diretório do projeto.

```
~$ cd eb-symfony
```

Crie um pacote de origem com os arquivos criados pelo Composer. O comando a seguir cria um pacote de origem chamado `symfony-default.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

```
eb-symfony$ zip ../symfony-default.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Symfony no ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

> **Note**
>
> Para otimizar mais o pacote de origem, inicie um repositório Git e use o comando `git archive`para criar o pacote de origem. O projeto do Symfony padrão inclui um arquivo `.gitignore` que diz para o Git excluir a pasta `vendor` e outros arquivos que não são necessários para implantação.

Definir as configurações do Composer

Quando a implantação for concluída, clique na URL para abrir a aplicação Symfony no navegador:

O que é isso? Por padrão, o Elastic Beanstalk fornece a raiz do seu projeto no caminho raiz do site. Neste caso, contudo, a página padrão (`app.php`) está em um nível abaixo na pasta `web`. Você pode verificar isso adicionando `/public` ao URL. Por exemplo, .`http://symfony.us-east-2.elasticbeanstalk.com/public`

Para fornecer a aplicação Symfony no caminho raiz, use o console do Elastic Beanstalk para configurar a raiz do documento para o site.

Para configurar a raiz do documento do seu site

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Para Document Root (Raiz do documento), insira **`/public`**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
7. Quando a atualização for concluída, clique no URL para reabrir seu site no navegador.

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Próximas etapas

Para obter mais informações sobre Symfony, consulte O que é o Symfony? no site symfony.com.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Neste tutorial, você usou o console do Elastic Beanstalk para configurar as opções do Composer. Para tornar essa configuração parte de origem do aplicativo, você pode usar um arquivo de configuração, como o seguinte.

Example .ebextensions/composer.config

```
option_settings:
  aws:elasticbeanstalk:container:php:phpini:
    document_root: /public
```

Para obter mais informações, consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`).

O Symfony usa seus próprios arquivos de configuração para configurar as conexões do banco de dados. Para obter instruções sobre como se conectar a um banco de dados com o Symfony, consulte Conexão a um banco de dados com Symfony.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantar uma aplicação PHP de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Este tutorial mostra o processo de iniciar uma instância de banco de dados RDS externa e configurar um ambiente de alta disponibilidade executando um aplicativo PHP para se conectar a ela. AWS Elastic Beanstalk A execução de uma instância de banco de dados externa ao Elastic Beanstalk desvincula o banco de dados do ciclo de vida do ambiente. Com isso, é possível se conectar ao mesmo banco de dados de vários ambientes, trocar um banco de dados por outro ou realizar uma implantação azul/verde sem afetar seu banco de dados.

O tutorial usa um exemplo de aplicativo PHP que usa um banco de dados MySQL para armazenar dados de texto fornecidos pelo usuário. O aplicativo de exemplo usa arquivos de configuração para definir as configurações de PHP e para criar uma tabela no banco de dados para o aplicativo usar. Ele também mostra como usar um arquivo do Composer para instalar pacotes durante a implantação.

Seções

- Pré-requisitos
- Executar uma instância de banco de dados no Amazon RDS
- Criar um ambiente do Elastic Beanstalk
- Configurar grupos de segurança, propriedades de ambiente e escalabilidade
- Implantar o aplicativo de exemplo
- Limpeza
- Próximas etapas

Pré-requisitos

Antes de começar, baixe o pacote de origem do aplicativo de amostra em GitHub: -app-1.3.zip eb-demo-php-simple

Os procedimentos neste tutorial para tarefas do Amazon Relational Database Service (Amazon RDS) assumem que você esteja iniciando recursos em uma Amazon Virtual Private Cloud (Amazon VPC) padrão. Todas as novas contas incluem uma VPC padrão em cada região. Se você não tem uma VPC padrão, os procedimentos podem variar. Consulte Usar o Elastic Beanstalk com o Amazon RDS para obter instruções sobre EC2 plataformas VPC clássicas e personalizadas.

Executar uma instância de banco de dados no Amazon RDS

Para usar um banco de dados externo com uma aplicação em execução no Elastic Beanstalk, primeiro execute uma instância de banco de dados com o Amazon RDS. Quando uma instância é executada com o Amazon RDS, ela é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk e não será encerrada nem monitorada pelo Elastic Beanstalk.

Use o console do Amazon RDS para executar uma instância de banco de dados MySQL Multi-AZ. Escolha uma implantação Multi-AZ para garantir que seu banco de dados execute o failover e continue disponível se a instância de banco de dados de origem ficar fora de serviço.

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o console do RDS.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

   > **Important**
   >
   > Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:

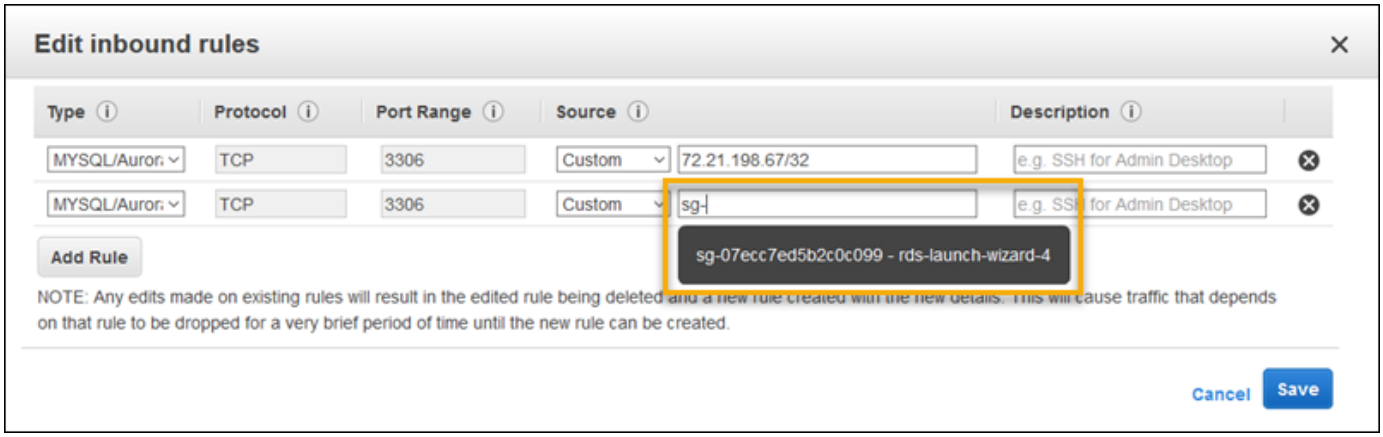
   - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
   - Implantação Multi-AZ — Para alta disponibilidade, defina isso como Criar um nó Replica/ Reader Aurora em uma AZ diferente.
   - Master username (Nome do usuário principal) e Master password (Senha principal):o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.

7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Em seguida, modifique o security group anexado à sua instância de banco de dados para permitir o tráfego de entrada na porta apropriada. Este é o mesmo grupo de segurança a ser anexado ao ambiente do Elastic Beanstalk mais tarde, para que a regra adicionada conceda permissão de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para ver o grupo de segurança no EC2 console da Amazon.
6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Escolha Editar.
8. Escolha Add Rule.
9. Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
10. Em Source (Origem), digite **`sg-`** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as EC2 instâncias da Amazon no ambiente possam ter acesso ao banco de dados.



11. Escolha Salvar.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, crie o ambiente do Elastic Beanstalk.

Criar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo. Depois de iniciar o ambiente, você pode configurar o ambiente para se conectar ao banco de dados e, em seguida, implantar o aplicativo de amostra do qual você baixou GitHub.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.

- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele. A instância de banco de dados RDS que você executou está fora do seu ambiente, portanto você é responsável por gerenciar o ciclo de vida dela.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Configurar grupos de segurança, propriedades de ambiente e escalabilidade

Adicione o security group da instância de banco de dados ao seu ambiente em execução. Esse procedimento faz com que o Elastic Beanstalk provisione novamente todas as instâncias no ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
  - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
    a. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
    b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
    c. No painel de navegação, escolha Configuration (Configuração).
    d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
    e. Em grupos EC2 de segurança, escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança de instâncias que o Elastic Beanstalk cria.
    f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
    g. Leia o aviso e escolha Confirm.
  - Para adicionar um grupo de segurança usando um arquivo de configuração, use o exemplo de arquivo `securitygroup-addexisting.config`.

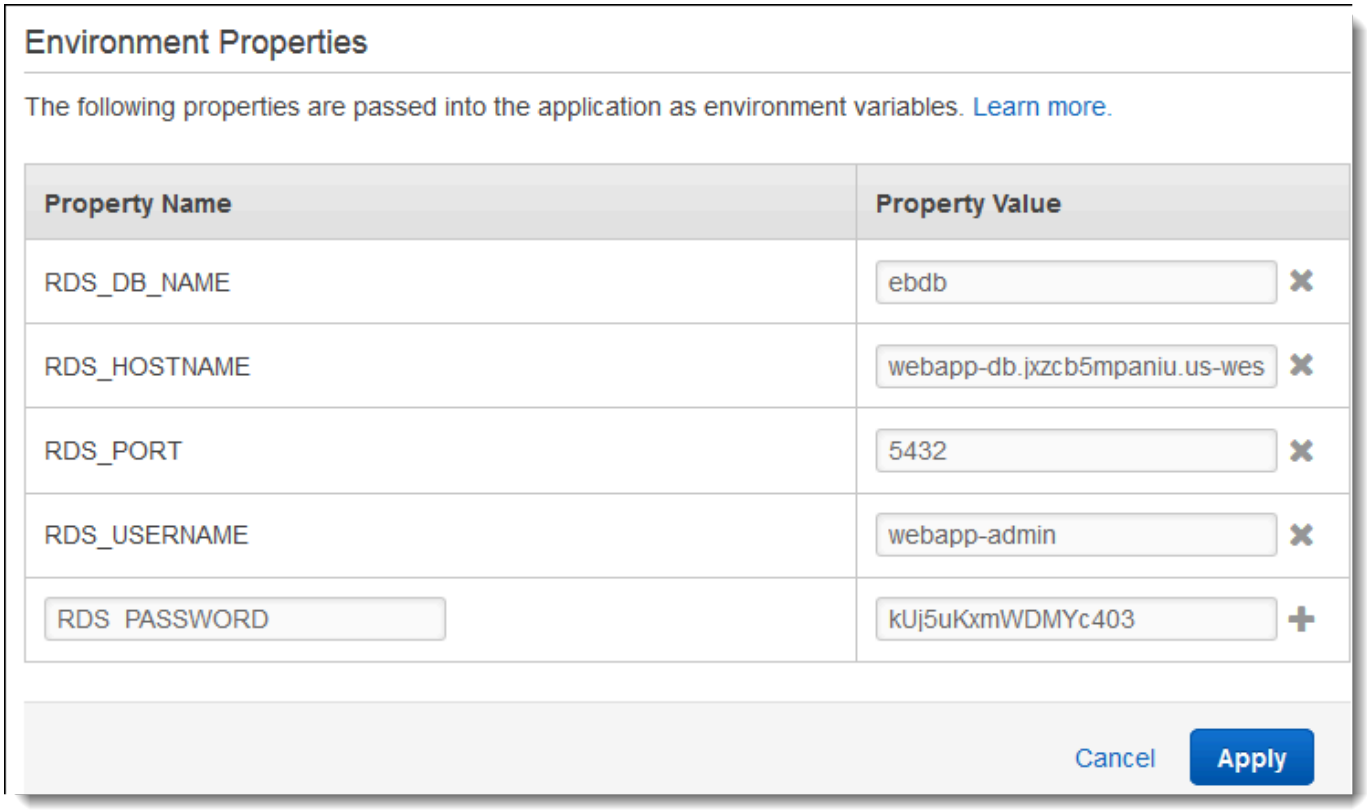
Em seguida, use as propriedades do ambiente para passar as informações de conexão para o ambiente. A aplicação demonstrativa usa um conjunto padrão de propriedades que correspondem àquelas que o Elastic Beanstalk configura quando você provisiona um banco de dados no ambiente.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no console do RDS.

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.

Nome da propriedade	Descrição	Valor da propriedade
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias o tempo todo para impedir que os servidores Web em seu ambiente sejam um ponto único de falha e para permitir que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).

4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Implantar o aplicativo de exemplo

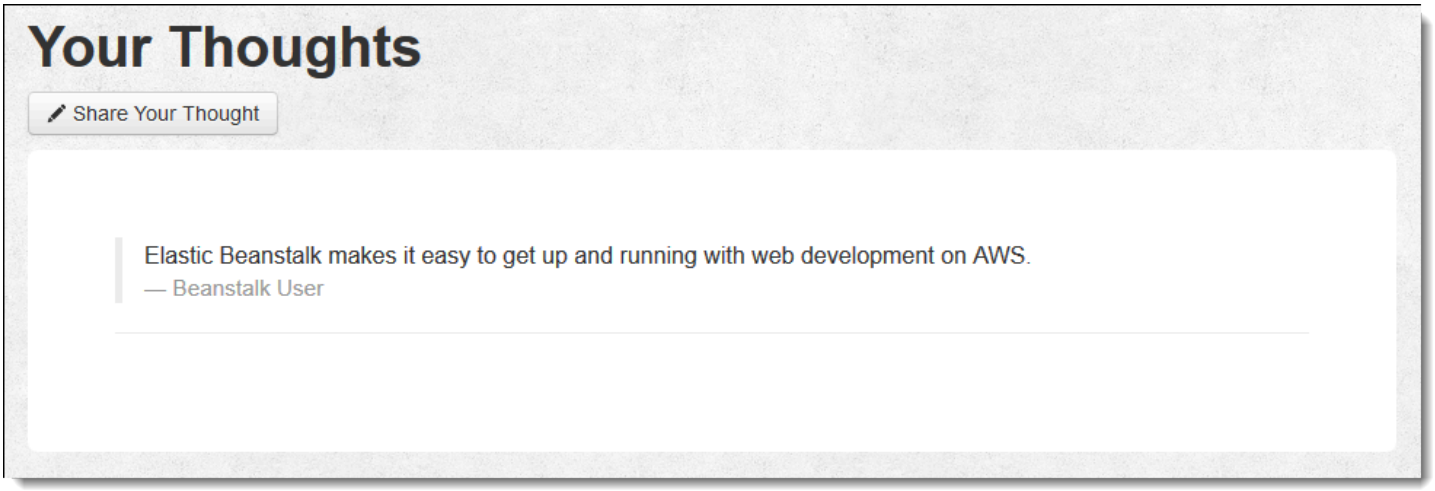
Agora, o ambiente está pronto para executar a aplicação demonstrativa e conectar-se ao Amazon RDS. Implante o aplicativo de exemplo em seu ambiente.

> **Note**
>
> Baixe o pacote de origem em GitHub, se ainda não o fez: eb-demo-php-simple -app-1.3.zip

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

O site coleta os comentários do usuário e usa um banco de dados MySQL para armazenar os dados. Para adicionar um comentário, escolha Share Your Thought (Compartilhar sua opinião), insira um comentário e escolha Submit Your Thought (Enviar sua opinião). O aplicativo Web grava o comentário no banco de dados para que qualquer instância do ambiente possa ler e ele não será perdido se as instâncias ficarem fora de serviço.



Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.

4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

O aplicativo de exemplo usa os arquivos de configuração para especificar as configurações de PHP e criar uma tabela no banco de dados, se ainda não existir. Você também pode usar um arquivo de configuração para definir as configurações de security group das instâncias durante a criação do ambiente para evitar atualizações de configuração demoradas. Consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Se você precisa de um banco de dados de alta performance, considere usar o Amazon Aurora. O Amazon Aurora é um mecanismo de banco de dados compatível com MySQL que oferece recursos de banco de dados comerciais a baixo custo. Para conectar seu aplicativo a outro banco de dados, repita as etapas de configuração de grupo de segurança e atualize as propriedades de ambiente relacionadas ao RDS.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantação de um WordPress site de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Este tutorial descreve como iniciar uma instância de banco de dados Amazon RDS externa e, em seguida AWS Elastic Beanstalk, como configurar um ambiente de alta disponibilidade executando um WordPress site para se conectar a ela. O site usa o Amazon Elastic File System (Amazon EFS) como o armazenamento compartilhado para arquivos carregados.

A execução de uma instância de banco de dados externa ao Elastic Beanstalk desvincula o banco de dados do ciclo de vida do ambiente. Com isso, é possível se conectar ao mesmo banco de dados de vários ambientes, trocar um banco de dados por outro ou realizar uma implantação azul/verde sem afetar seu banco de dados.

> **Note**
>
> Para obter informações atuais sobre a compatibilidade das versões do PHP com WordPress versões, consulte Compatibilidade e WordPress versões do PHP no WordPress site. Você deve consultar essas informações antes de atualizar para uma nova versão do PHP para suas WordPress implementações.

Tópicos

- Pré-requisitos
- Executar uma instância de banco de dados no Amazon RDS
- Baixar WordPress
- Iniciar um ambiente do Elastic Beanstalk
- Configurar grupos de segurança e propriedades de ambiente
- Configurar e implantar seu aplicativo
- Instalar WordPress
- Atualizar chaves e salts
- Remover restrições de acesso
- Configurar o grupo de Auto Scaling
- Atualizar WordPress
- Limpeza
- Próximas etapas

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

VPC padrão

Os procedimentos do Amazon Relational Database Service (Amazon RDS) neste tutorial assumem que você esteja iniciando recursos em uma Amazon Virtual Private Cloud (Amazon VPC) padrão. Todas as novas contas incluem uma VPC padrão em cada AWS região. Se você não tem uma VPC padrão, os procedimentos podem variar. Consulte Usar o Elastic Beanstalk com o Amazon RDS para obter instruções sobre EC2 plataformas VPC clássicas e personalizadas.

AWS Regiões

O aplicativo de amostra usa o Amazon EFS, que só funciona em AWS regiões que oferecem suporte ao Amazon EFS. Para saber mais sobre AWS as regiões suportadas, consulte Amazon Elastic File System Endpoints and Quotas no. Referência geral da AWS

Executar uma instância de banco de dados no Amazon RDS

Quando uma instância é executada com o Amazon RDS, ela é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk e não será encerrada nem monitorada pelo Elastic Beanstalk.

Nas etapas a seguir, você usará o console do Amazon RDS para:

- Iniciar um banco de dados com o mecanismo MySQL .
- Habilitar uma Multi-AZ deployment (Implantação Multi-AZ). Isso cria um modo de espera em uma zona de disponibilidade (AZ) diferente para fornecer redundância de dados, eliminar I/O congelamentos e minimizar picos de latência durante os backups do sistema.

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o console do RDS.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

> **Important**
>
> Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:
   - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
   - Implantação Multi-AZ — Para alta disponibilidade, defina isso como Criar um nó Replica/ Reader Aurora em uma AZ diferente.
   - Master username (Nome do usuário principal) e Master password (Senha principal):o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.
7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Depois que sua instância de banco de dados for criada, modifique o grupo de segurança anexado a ela para permitir o tráfego de entrada na porta apropriada.

> **Note**
>
> Este é o mesmo grupo de segurança a ser anexado ao ambiente do Elastic Beanstalk mais tarde, para que a regra adicionada agora conceda permissão de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para ver o grupo de segurança no EC2 console da Amazon.
6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Escolha Editar.
8. Escolha Add Rule.
9. Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
10. Em Source (Origem), digite **`sg-`** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as EC2 instâncias da Amazon no ambiente possam ter acesso ao banco de dados.



11. Escolha Salvar.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, baixe WordPress e crie seu ambiente do Elastic Beanstalk.

Baixar WordPress

Para se preparar para a implantação WordPress usando AWS Elastic Beanstalk, você deve copiar os WordPress arquivos para o seu computador e fornecer as informações de configuração corretas.

Para criar um WordPress projeto

1. Baixe WordPress em wordpress.org.

```
~$curl https://wordpress.org/wordpress-6.2.tar.gz -o wordpress.tar.gz
```

2. Faça download dos arquivos de configuração no repositório de exemplo.

```
~$ wget https://github.com/aws-samples/eb-php-wordpress/releases/download/v1.1/eb-
php-wordpress-v1.zip
```

3. Extraia WordPress e altere o nome da pasta.

```
~$ tar -xvf wordpress.tar.gz
~$ mv wordpress wordpress-beanstalk
~$ cd wordpress-beanstalk
```

4. Extraia os arquivos de configuração durante a WordPress instalação.

```
~/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
```

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Depois de iniciar o ambiente, você pode configurá-lo para se conectar ao banco de dados e, em seguida, implantar o WordPress código no ambiente.

Nas etapas a seguir, você usará o console do Elastic Beanstalk para:

- Criar uma aplicação do Elastic Beanstalk usando a plataforma PHP gerenciada.
- Aceitar as configurações padrão e o código de exemplo.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação do ambiente leva cerca de cinco minutos e cria os seguintes recursos.

Recursos criados do Elastic Beanstalk

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.

- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Como a instância de banco de dados do Amazon RDS executada está fora do ambiente, você é responsável por gerenciar o ciclo de vida dela.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Configurar grupos de segurança e propriedades de ambiente

Adicione o security group da instância de banco de dados ao seu ambiente em execução. Esse procedimento faz com que o Elastic Beanstalk provisione novamente todas as instâncias no ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
  - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
    a. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
    b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
    c. No painel de navegação, escolha Configuration (Configuração).
    d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
    e. Em grupos EC2 de segurança, escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança de instâncias criado pelo Elastic Beanstalk.
    f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
    g. Leia o aviso e escolha Confirm.
  - Para adicionar um grupo de segurança usando um arquivo de configuração, use o exemplo de arquivo `securitygroup-addexisting.config`.

Em seguida, use as propriedades do ambiente para passar as informações de conexão para o ambiente.

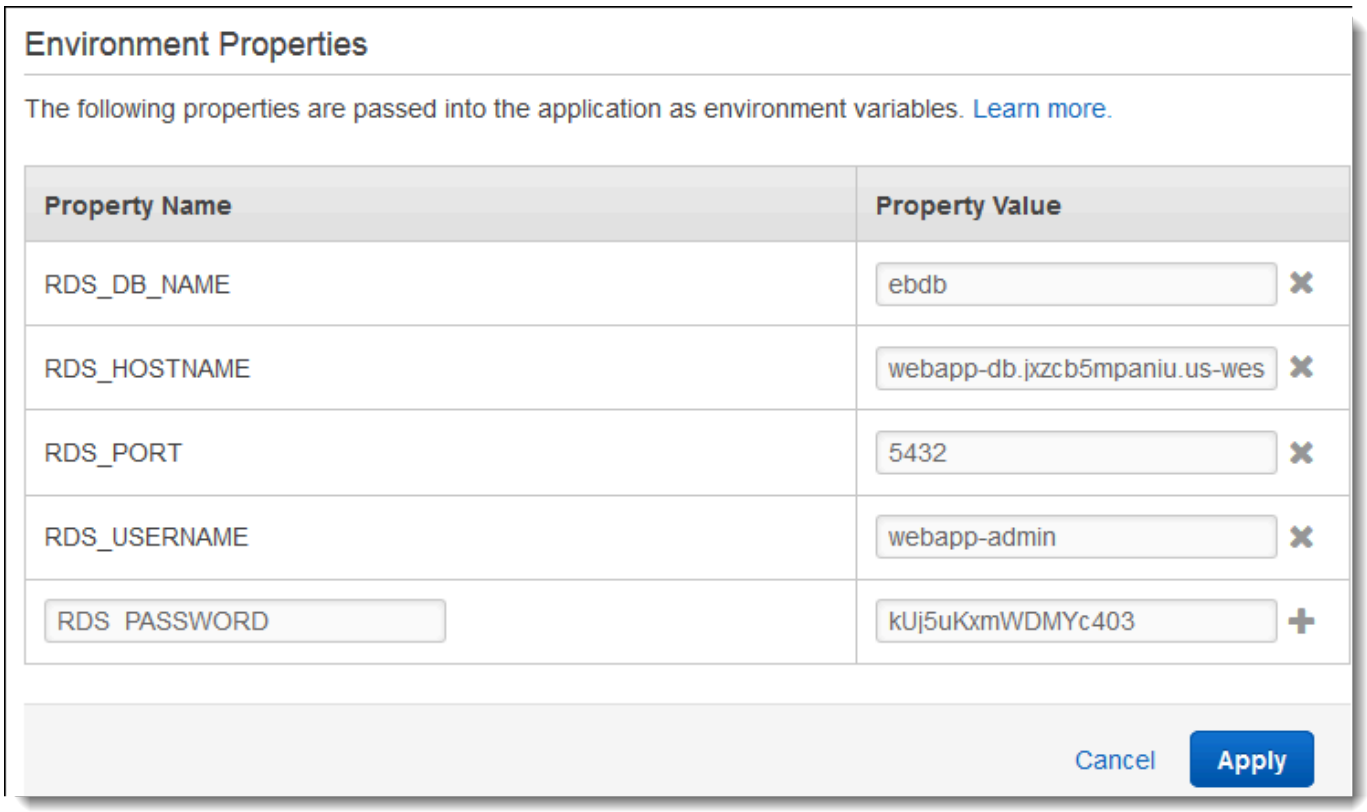
O WordPress aplicativo usa um conjunto padrão de propriedades que correspondem às que o Elastic Beanstalk configura quando você provisiona um banco de dados em seu ambiente.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no console do RDS.

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.

Nome da propriedade	Descrição	Valor da propriedade
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar e implantar seu aplicativo

Verifique se a estrutura da pasta `wordpress-beanstalk` está correta, conforme exibido.

```
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
```

O arquivo `wp-config.php` personalizado do repositório do projeto usa as variáveis de ambiente que você definiu na etapa anterior para configurar a conexão do banco de dados. A pasta `.ebextensions` contém arquivos de configuração que criam recursos adicionais no ambiente do Elastic Beanstalk.

Os arquivos de configuração devem ser modificados para funcionar com sua conta. Substitua os valores do espaço reservado nos arquivos pelos apropriados IDs e crie um pacote de origem.

Como atualizar os arquivos de configuração e criar um pacote de origem

1. Modifique os arquivos de configuração da seguinte forma.

   - `.ebextensions/dev.config`— Restringe o acesso ao seu ambiente para protegê-lo durante o processo WordPress de instalação. Substitua o endereço IP do espaço reservado próximo à parte superior do arquivo pelo endereço IP público do computador que você usará para acessar o site do seu ambiente e concluir a WordPress instalação.

     > **Note**
     >
     > Dependendo da sua rede, pode ser necessário usar um bloco de endereços IP.

- `.ebextensions/efs-create.config`— Cria um sistema de arquivos EFS e pontos de montagem em cada disponibilidade Zone/subnet em sua VPC. Identifique sua VPC e sub-rede padrão no console IDs da Amazon VPC.

2. Crie um pacote de origem com os arquivos em sua pasta de projeto. O comando a seguir cria um pacote de origem chamado `wordpress-beanstalk.zip`.

```
~/eb-wordpress$ zip ../wordpress-beanstalk.zip -r * .[^.]*
```

Faça o upload do pacote de origem para o Elastic WordPress Beanstalk para implantá-lo em seu ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Instalar WordPress

Para concluir sua WordPress instalação

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha o URL do ambiente para abrir seu site em um navegador. Você é redirecionado para um assistente WordPress de instalação porque ainda não configurou o site.
4. Execute uma instalação padrão. O arquivo `wp-config.php` já está presente no código-fonte e configurado para ler as informações de conexão do banco de dados do ambiente, portanto você não deve ser solicitado para configurar a conexão.

A instalação leva cerca de um minuto para ser concluída.

Atualizar chaves e salts

O arquivo WordPress de configuração `wp-config.php` também lê valores de chaves e sais das propriedades do ambiente. Atualmente, essas propriedades são todas definidas como `test` pelo arquivo `wordpress.config` na pasta `.ebextensions`.

O salt de hash pode ser qualquer valor que atenda aos requisitos de propriedade de ambiente, mas você não deve armazená-lo no controle de origem. Use o console do Elastic Beanstalk para definir essas propriedades diretamente no ambiente.

Para atualizar propriedades de ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuração.
4. Em Software (Software), escolha Edit (Editar).
5. Para `Environment properties`, modifique as seguintes propriedades:
   - `AUTH_KEY`: o valor escolhido para `AUTH_KEY`.
   - `SECURE_AUTH_KEY`: o valor escolhido para `SECURE_AUTH_KEY`.
   - `LOGGED_IN_KEY`: o valor escolhido para `LOGGED_IN_KEY`.
   - `NONCE_KEY`: o valor escolhido para `NONCE_KEY`.
   - `AUTH_SALT`: o valor escolhido para `AUTH_SALT`.
   - `SECURE_AUTH_SALT`: o valor escolhido para `SECURE_AUTH_SALT`.
   - `LOGGED_IN_SALT`: o valor escolhido para `LOGGED_IN_SALT`.
   - `NONCE_SALT`: o valor escolhido para `NONCE_SALT`.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

> **Note**
>
> A definição de propriedades no ambiente substitui os valores diretamente em `wordpress.config`.

Remover restrições de acesso

O projeto de exemplo inclui o arquivo de configuração `loadbalancer-sg.config`. Ele cria um grupo de segurança e o atribui ao load balancer do ambiente, usando o endereço IP que você configurou no `dev.config`. Ele restringe o acesso HTTP na porta 80 às conexões da sua rede. Caso contrário, uma pessoa externa poderia se conectar ao seu site antes de você instalar WordPress e configurar sua conta de administrador.

Agora que você instalou WordPress, remova o arquivo de configuração para abrir o site para o mundo.

Para remover a restrição e atualizar seu ambiente

1. Exclua o arquivo `.ebextensions/loadbalancer-sg.config` de seu diretório de projeto.

   ```
   ~/wordpress-beanstalk$ rm .ebextensions/loadbalancer-sg.config
   ```

2. Criar um pacote de origem.

   ```
   ~/eb-wordpress$ zip ../wordpress-beanstalk-v2.zip -r * .[^.]*
   ```

Faça o upload do pacote de origem para o Elastic WordPress Beanstalk para implantá-lo em seu ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Configurar o grupo de Auto Scaling

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias em todos os momentos para evitar que os servidores da web no seu ambiente sejam um único ponto de falha. Isso também permite que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para oferecer suporte a uploads de conteúdo em várias instâncias, o projeto demonstrativo usa o Amazon EFS para criar um sistema de arquivos compartilhados. Crie uma postagem no site e faça upload de conteúdo para armazená-lo no sistema de arquivos compartilhados. Visualize a publicação e atualize a página várias vezes para atingir ambas as instâncias e verificar se o sistema de arquivos compartilhado está funcionando.

Atualizar WordPress

Para atualizar para uma nova versão do WordPress, faça backup do seu site e implante-o em um novo ambiente.

> **Important**
>
> Não use a funcionalidade de atualização em seus arquivos de origem WordPress nem atualize seus arquivos de origem para usar uma nova versão. Ambas as ações podem fazer com que sua postagem URLs retorne erros 404, mesmo que eles ainda estejam no banco de dados e no sistema de arquivos.

Para atualizar WordPress

1. No console do WordPress administrador, use a ferramenta de exportação para exportar suas publicações para um arquivo XML.
2. Implante e instale a nova versão do WordPress Elastic Beanstalk com as mesmas etapas que você usou para instalar a versão anterior. Para evitar o tempo de inatividade, é possível criar um ambiente com a nova versão.
3. Na nova versão, instale a ferramenta WordPress Importador no console administrativo e use-a para importar o arquivo XML contendo suas postagens. Se as postagens foram criadas pelo usuário administrador na versão antiga, as atribua para o usuário administrador no novo site em vez de tentar importar o usuário administrador.
4. Se você implantou a nova versão em um ambiente separado, faça uma troca de CNAME para redirecionar os usuários de um local antigo para o novo site.

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o console do Amazon RDS.

2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

O aplicativo de exemplo usa os arquivos de configuração para especificar as configurações de PHP e criar uma tabela no banco de dados, se ainda não existir. Você também pode usar um arquivo de configuração para definir as configurações de security group das instâncias durante a criação do ambiente para evitar atualizações de configuração demoradas. Consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Se você precisa de um banco de dados de alta performance, considere usar o Amazon Aurora. O Amazon Aurora é um mecanismo de banco de dados compatível com MySQL que oferece recursos de banco de dados comerciais a baixo custo. Para conectar seu aplicativo a outro banco de dados, repita as etapas de configuração de grupo de segurança e atualize as propriedades de ambiente relacionadas ao RDS.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantar um site Drupal de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk

Este tutorial mostra o processo de execução de uma instância de banco de dados RDS externa a. AWS Elastic Beanstalk Em seguida, ele descreve como configurar um ambiente de alta

disponibilidade que executa um site do Drupal para se conectar a ele. O site usa o Amazon Elastic File System (Amazon EFS) como o armazenamento compartilhado para arquivos carregados. A execução de uma instância de banco de dados externa ao Elastic Beanstalk separa o banco de dados do ciclo de vida do seu ambiente e permite que você se conecte ao mesmo banco de dados a partir de vários ambientes, troque um banco de dados por outro ou execute uma implantação sem afetar seu banco de dados. blue/green

Seções

- Pré-requisitos
- Executar uma instância de banco de dados no Amazon RDS
- Iniciar um ambiente do Elastic Beanstalk
- Definir configurações de segurança e propriedades de ambiente
- Configurar e implantar seu aplicativo
- Instalar o Drupal
- Atualizar a configuração de Drupal e remover as restrições de acesso
- Configurar o grupo de Auto Scaling
- Limpeza
- Próximas etapas

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

Os procedimentos neste tutorial para tarefas do Amazon Relational Database Service (Amazon RDS) assumem que você esteja iniciando recursos em uma Amazon Virtual Private Cloud (Amazon VPC) padrão. Todas as novas contas incluem uma VPC padrão em cada região. Se você não tem uma VPC padrão, os procedimentos podem variar. Consulte Usar o Elastic Beanstalk com o Amazon RDS para obter instruções sobre EC2 plataformas VPC clássicas e personalizadas.

A aplicação demonstrativa usa o Amazon EFS. Ele só funciona em AWS regiões que oferecem suporte ao Amazon EFS. Para saber mais sobre o suporte a AWS regiões, consulte Amazon Elastic File System Endpoints and Quotas no. Referência geral da AWS

Se a plataforma do seu ambiente do Elastic Beanstalk usa PHP 7.4 ou anterior, recomendamos que você use o Drupal versão 8.9.13 para este tutorial. Para plataformas instaladas com PHP 8.0 ou posterior, recomendamos que você use o Drupal 9.1.5.

Para obter mais informações sobre versões do Drupal e as versões do PHP compatíveis, consulte Requisitos do PHP no site do Drupal. As versões principais recomendadas pelo Drupal estão listadas no site https://www.drupal. org/project/drupal.

Executar uma instância de banco de dados no Amazon RDS

Para usar um banco de dados externo com uma aplicação em execução no Elastic Beanstalk, primeiro execute uma instância de banco de dados com o Amazon RDS. Quando uma instância é executada com o Amazon RDS, ela é completamente independente do Elastic Beanstalk e dos seus ambientes do Elastic Beanstalk e não será encerrada nem monitorada pelo Elastic Beanstalk.

Use o console do Amazon RDS para executar uma instância de banco de dados MySQL Multi-AZ. Escolha uma implantação Multi-AZ para garantir que seu banco de dados execute o failover e continue disponível se a instância de banco de dados de origem ficar fora de serviço.

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o console do RDS.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

> **Important**
>
> Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:
   - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
   - Implantação Multi-AZ — Para alta disponibilidade, defina isso como Criar um nó Replica/ Reader Aurora em uma AZ diferente.
   - Master username (Nome do usuário principal) e Master password (Senha principal):o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.
7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Em seguida, modifique o security group anexado à sua instância de banco de dados para permitir o tráfego de entrada na porta apropriada. Este é o mesmo grupo de segurança a ser anexado ao ambiente do Elastic Beanstalk mais tarde, para que a regra adicionada conceda permissão de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para ver o grupo de segurança no EC2 console da Amazon.

6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Escolha Editar.
8. Escolha Add Rule.
9. Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
10. Em Source (Origem), digite **`sg-`** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as EC2 instâncias da Amazon no ambiente possam ter acesso ao banco de dados.



11. Escolha Salvar.

A criação de uma instância de banco de dados leva cerca de 10 minutos. Enquanto isso, inicie o ambiente do Elastic Beanstalk.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma PHP e aceite as configurações padrão e código de exemplo. Depois de iniciar o ambiente, você poderá configurá-lo para conectar-se ao banco de dados e implantar o código Drupal no ambiente.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.

3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.

- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele. A instância de banco de dados RDS que você executou está fora do seu ambiente, portanto você é responsável por gerenciar o ciclo de vida dela.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Definir configurações de segurança e propriedades de ambiente

Adicione o security group da instância de banco de dados ao seu ambiente em execução. Esse procedimento faz com que o Elastic Beanstalk provisione novamente todas as instâncias no ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
  - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
    a. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS

b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

c. No painel de navegação, escolha Configuration (Configuração).

d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).

e. Em grupos EC2 de segurança, escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança de instâncias que o Elastic Beanstalk cria.

f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

g. Leia o aviso e escolha Confirm.

- Para adicionar um grupo de segurança usando um arquivo de configuração, use o exemplo de arquivo `securitygroup-addexisting.config`.

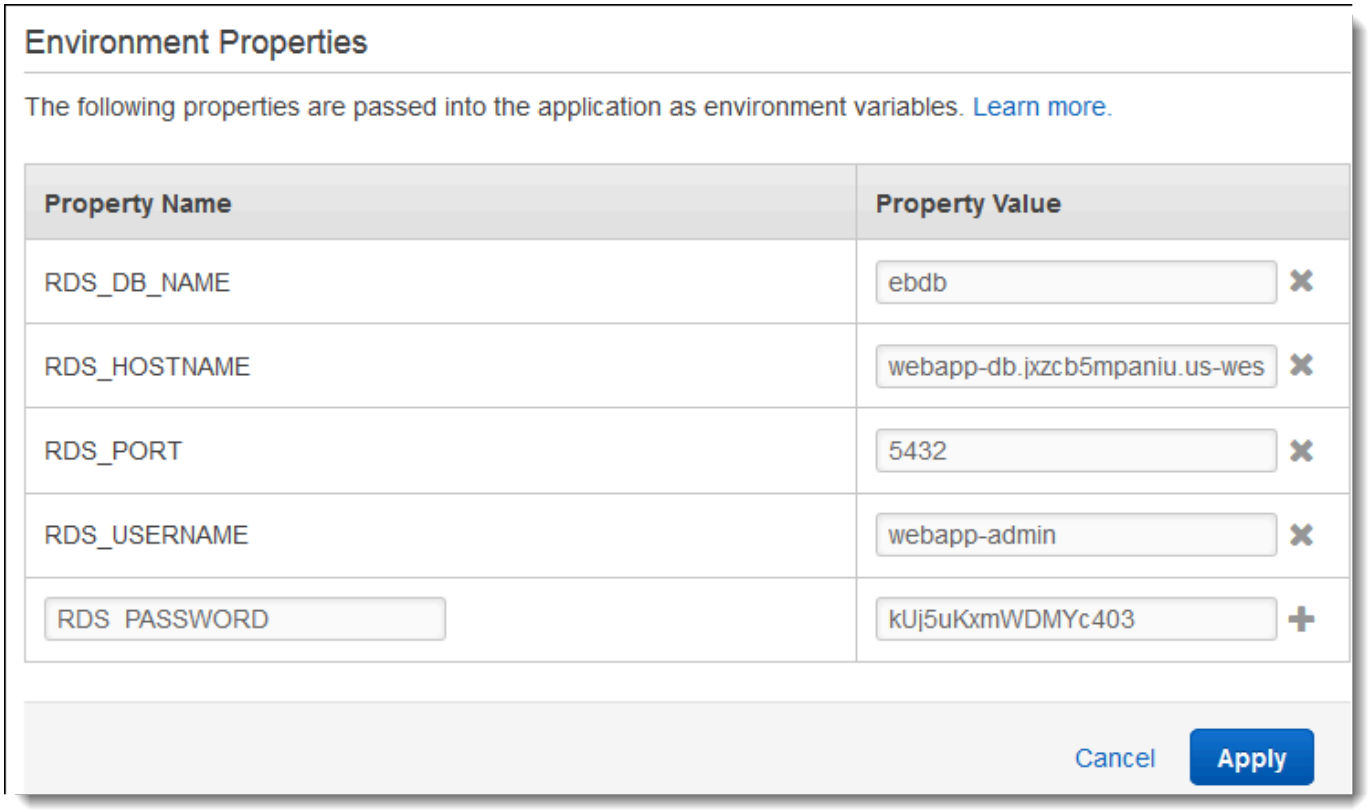
Em seguida, use as propriedades do ambiente para passar as informações de conexão para o ambiente. A aplicação demonstrativa usa um conjunto padrão de propriedades que correspondem àquelas que o Elastic Beanstalk configura quando você provisiona um banco de dados no ambiente.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no console do RDS.

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.

Nome da propriedade	Descrição	Valor da propriedade
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
`RDS_PASSWORD`	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Após a instalação do Drupal, você precisa se conectar à instância com o SSH para recuperar alguns detalhes de configuração. Atribua uma chave SSH às instâncias do ambiente.

Para configurar SSH

1. Se você ainda não criou um par de chaves, abra a página de pares de chaves do EC2 console da Amazon e siga as instruções para criar um.
2. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
3. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
4. No painel de navegação, escolha Configuration (Configuração).
5. Em Security (Segurança), escolha Edit (Editar).
6. Para par de EC2 chaves, escolha seu par de chaves.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar e implantar seu aplicativo

Para criar um projeto do Drupal para o Elastic Beanstalk, baixe o código-fonte do Drupal e combine-o com os arquivos no repositório aws-samples/ em. eb-php-drupal GitHub

Para criar um projeto Drupal

1. Execute o comando a seguir para baixar o Drupal em www.drupal.org/download. Para saber mais sobre downloads, consulte o site do Drupal.

   Se a plataforma do seu ambiente do Elastic Beanstalk usa PHP 7.4 ou anterior, recomendamos que você faça o download da versão 8.9.13 do Drupal para este tutorial. Execute o comando a seguir para fazer o download.

   ```
   ~$ curl https://ftp.drupal.org/files/projects/drupal-8.9.13.tar.gz -o drupal.tar.gz
   ```

   Se sua plataforma usa PHP 8.0 ou posterior, recomendamos que você faça o download do Drupal 9.1.5. Você pode usar este comando para fazer o download.

   ```
   ~$ curl https://ftp.drupal.org/files/projects/drupal-9.1.5.tar.gz -o drupal.tar.gz
   ```

   Para obter mais informações sobre versões do Drupal e as versões do PHP compatíveis, consulte Requisitos do PHP na documentação oficial do Drupal. As versões básicas que o Drupal recomenda estão listadas no site do Drupal.
2. Use o seguinte comando para fazer o download dos arquivos de configuração do repositório de exemplo:

   ```
   ~$ wget https://github.com/aws-samples/eb-php-drupal/releases/download/v1.1/eb-php-drupal-v1.zip
   ```
3. Extraia o Drupal e mude o nome da pasta.

   Se você baixou o Drupal 8.9.13:

   ```
   ~$ tar -xvf drupal.tar.gz
   ~$ mv drupal-8.9.13 drupal-beanstalk
   ~$ cd drupal-beanstalk
   ```

   Se você baixou o Drupal 9.1.5:

```
~$ tar -xvf drupal.tar.gz
~$ mv drupal-9.1.5 drupal-beanstalk
~$ cd drupal-beanstalk
```

4. Extraia os arquivos de configuração durante a instalação do 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
```

Verifique se a estrutura da pasta `drupal-beanstalk` está correta, conforme exibido.

```
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
```

O arquivo `beanstalk-settings.php` do repositório do projeto usa as variáveis de ambiente que você definiu na etapa anterior para configurar a conexão do banco de dados. A pasta `.ebextensions` contém arquivos de configuração que criam recursos adicionais no ambiente do Elastic Beanstalk.

Os arquivos de configuração devem ser modificados para funcionar com sua conta. Substitua os valores do espaço reservado nos arquivos pelos apropriados IDs e crie um pacote de origem.

Para atualizar os arquivos de configuração e criar um pacote de origem.

1. Modifique os arquivos de configuração da seguinte forma.

   - `.ebextensions/dev.config`: restringe o acesso ao seu ambiente para o seu endereço IP a fim de protegê-lo durante o processo de instalação do Drupal. Substitua o endereço IP de espaço reservado próximo à parte superior do arquivo pelo seu endereço IP público.
   - `.ebextensions/efs-create.config`: cria um sistema de arquivos EFS e pontos de montagem em cada zona de disponibilidade/sub-rede na VPC. Identifique sua VPC e sub-rede padrão no console IDs da Amazon VPC.

2. Crie um pacote de origem com os arquivos em sua pasta de projeto. O comando a seguir cria um pacote de origem chamado `drupal-beanstalk.zip`. Ele exclui arquivos na pasta `vendor`, que tomam muito espaço e não são necessários para implantação da aplicação no Elastic Beanstalk.

   ```
   ~/eb-drupal$ zip ../drupal-beanstalk.zip -r * .[^.]* -x "vendor/*"
   ```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Drupal no ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Instalar o Drupal

Para concluir a instalação do Drupal

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha o URL do ambiente para abrir seu site em um navegador. Você será redirecionado para um assistente de instalação do Drupal, pois o site ainda não foi configurado.
4. Realize a instalação padrão usando as seguintes configurações para o banco de dados:

   - Nome do banco de dados: o DB Name (Nome do banco de dados) exibido no console do Amazon RDS.
   - Nome de usuário e senha do banco de dados: os valores de Master username (Nome do usuário principal) e Master Password (Senha principal) inseridos ao criar o banco de dados.
   - Advanced Options (Opções avançadas) > Host: o Endpoint da instância de banco de dados exibida no console do Amazon RDS.

A instalação leva cerca de um minuto para ser concluída.

Atualizar a configuração de Drupal e remover as restrições de acesso

O processo de instalação do Drupal criou um arquivo chamado `settings.php` na pasta `sites/default` na instância. Você precisa desse arquivo em seu código-fonte para evitar redefinir seu site em implantações subsequentes, mas o arquivo atualmente contém segredos os quais você não deseja confirmar a origem. Conecte-se à instância do aplicativo para recuperar informações do arquivo de configurações.

Para conectar sua instância de aplicativo com SSH

1. Abra a página de instâncias do EC2 console da Amazon.
2. Escolha a instância do aplicativo. É a que recebeu o nome do seu ambiente do Elastic Beanstalk.
3. Selecione Conectar.
4. Siga as instruções para conectar a instância com SSH. O comando será semelhante à seguinte.

```
$ ssh -i ~/.ssh/mykey ec2-user@ec2-00-55-33-222.us-west-2.compute.amazonaws.com
```

Obtenha o id do diretório de sincronização da última linha do arquivo de configurações.

```
[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';
```

O arquivo também contém os sites da chave de hash atual, mas você pode ignorar o valor atual e usar seu próprio.

Atribua o caminho do diretório de sincronização e chave de hash para as propriedades de ambiente. O arquivo de configurações personalizadas do repositório do projeto lê essas propriedades para configurar o site durante a implantação, além das propriedades de conexão de banco de dados que você definiu anteriormente.

Propriedades de configuração do Drupal

- `SYNC_DIR`: o caminho para o diretório de sincronização.
- `HASH_SALT`: qualquer valor de string que atenda aos requisitos de propriedade de ambiente.

Para configurar variáveis de ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

5. Role para baixo até Variáveis de ambiente de tempo de execução.
6. Selecione Adicionar variável de ambiente.
7. Em Fonte, selecione Texto sem formatação.

> **Note**
>
> Os valores do Secrets Manager e do SSM Parameter Store no menu suspenso servem para configurar variáveis de ambiente como segredos para armazenar dados confidenciais, como credenciais e chaves de API. Para obter mais informações, consulte Usando o Elastic AWS Secrets ManagerAWS Systems Manager Beanstalk com um Parameter Store.

8. Insira o nome da variável de ambiente e os pares de valores da variável de ambiente.
9. Se precisar adicionar mais variáveis, repita a Etapa 6 até a Etapa 8.
10. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Finalmente, o projeto de amostra inclui um arquivo de configuração (`loadbalancer-sg.config`) que cria um security group e o atribui ao load balancer do ambiente, usando o endereço IP configurado em `dev.config` para restringir o acesso HTTP na porta 80 às conexões da sua rede. Caso contrário, uma parte externa pode se conectar em seu sites antes de você ter instalado o Drupal e configurado sua conta de admin.

Para atualizar a configuração do Drupal e remover restrições de acesso

1. Exclua o arquivo `.ebextensions/loadbalancer-sg.config` de seu diretório de projeto.

```
~/drupal-beanstalk$ rm .ebextensions/loadbalancer-sg.config
```

2. Copie o arquivo `settings.php` personalizado na pasta dos sites.

```
~/drupal-beanstalk$ cp beanstalk-settings.php sites/default/settings.php
```

3. Criar um pacote de origem.

```
~/eb-drupal$ zip ../drupal-beanstalk-v2.zip -r * .[^.]* -x "vendor/*"
```

Faça upload do pacote de origem para o Elastic Beanstalk para implantar o Drupal no ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Configurar o grupo de Auto Scaling

Por fim, configure o grupo de Auto Scaling do seu ambiente com uma contagem mínima de instâncias mais alta. Execute pelo menos duas instâncias o tempo todo para impedir que os servidores Web em seu ambiente sejam um ponto único de falha e para permitir que você implante alterações sem colocar seu site fora do serviço.

Para configurar o grupo de Auto Scaling de seu ambiente para alta disponibilidade

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Capacity (Capacidade), escolha Edit (Editar).
5. Na seção Auto Scaling group (Grupo de Auto Scaling) defina Min instances (Mínimo de instâncias) como **2**.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para oferecer suporte a uploads de conteúdo em várias instâncias, o projeto demonstrativo usa o Amazon Elastic File System para criar um sistema de arquivos compartilhados. Crie uma postagem no site e faça upload de conteúdo para armazená-lo no sistema de arquivos compartilhados. Visualize a publicação e atualize a página várias vezes para atingir ambas as instâncias e verificar se o sistema de arquivos compartilhado está funcionando.

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Além disso, é possível encerrar recursos do banco de dados que você criou fora do ambiente do Elastic Beanstalk. Ao encerrar uma instância de banco de dados do Amazon RDS, você poderá criar um snapshot e restaurar os dados em outra instância posteriormente.

Para encerrar sua instância de banco de dados do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha a instância de banco de dados.
4. Escolha Ações e, em seguida, escolha Excluir.
5. Escolha se deseja criar um snapshot. Depois, escolha Excluir.

Próximas etapas

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

O aplicativo de exemplo usa os arquivos de configuração para especificar as configurações de PHP e criar uma tabela no banco de dados, se ainda não existir. Você também pode usar um arquivo de configuração para definir as configurações de security group da instância durante a criação do ambiente para evitar atualizações de configuração demoradas. Consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`) para obter mais informações.

Para desenvolvimento e teste, convém usar a funcionalidade do Elastic Beanstalk para adicionar uma instância de banco de dados gerenciada diretamente ao seu ambiente. Para obter instruções sobre como configurar um banco de dados no ambiente, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Se você precisa de um banco de dados de alta performance, considere usar o Amazon Aurora. O Amazon Aurora é um mecanismo de banco de dados compatível com MySQL que oferece recursos de banco de dados comerciais a baixo custo. Para conectar seu aplicativo a outro banco de dados, repita as etapas de configuração de grupo de segurança e atualize as propriedades de ambiente relacionadas ao RDS.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantação de aplicações Python com o Elastic Beanstalk

Este capítulo fornece instruções para configurar e implantar seu aplicativo web Python no. AWS Elastic Beanstalk O Elastic Beanstalk facilita a implantação, o gerenciamento e o dimensionamento das suas aplicações Web Python usando a Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk, ou o. AWS CLI APIs

Siga as etapas em QuickStart para Python para step-by-step obter instruções sobre como criar e implantar um aplicativo web Python Hello World com a CLI do EB.

Tópicos

- QuickStart: implante um aplicativo Python no Elastic Beanstalk
- Configurar seu ambiente de desenvolvimento Python para o Elastic Beanstalk
- Usar a plataforma Python do Elastic Beanstalk
- Implantar uma aplicação Flask no Elastic Beanstalk
- Implantar uma aplicação Django no Elastic Beanstalk
- Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Python Elastic Beanstalk
- Ferramentas e recursos do Python

QuickStart: implante um aplicativo Python no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo Python e sua implantação em um ambiente. AWS Elastic Beanstalk

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação Python
- Etapa 2: executar a aplicação localmente
- Etapa 3: implantar sua aplicação Python com a EB CLI
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços necessários.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Framework Python e Flask

Confirme que você tem uma versão funcional do Python com o `pip` instalado, executando os seguintes comandos.

```
~$ python3 --version
Python 3.N.N
>~$ python3 -m pip --version
pip X.Y.Z from ... (python 3.N.N)
```

Se algum dos comandos anteriores retornar "Python não encontrado", execute os seguintes comandos que usam `python` em vez de `python3`. A configuração de aliases e links simbólicos pode variar de acordo com o sistema operacional e as personalizações individuais, portanto, o comando `python3` pode não funcionar no seu computador.

```
~$ python --version
Python 3.N.N
>~$ python -m pip --version
pip X.Y.Z from ... (python 3.N.N)
```

Se você não tiver o Python instalado no computador local, poderá baixá-lo na página de downloads do Python no site do Python. Para obter uma lista das versões da linguagem Python compatíveis com o Elastic Beanstalk, consulte Plataformas Python compatíveis no guia de Plataformas do AWS Elastic Beanstalk . O site de downloads do Python fornece um link para o Guia do desenvolvedor do Python, onde você encontrará instruções de instalação e configuração.

> **Note**
>
> O pacote Python `pip` é incluído por padrão no Python 3.4 ou versão posterior.

Se o resultado indicar que você tem uma versão compatível do Python, mas não `pip`, consulte a página de instalação no site pip.pypa.io. Ele fornece orientação para instalar o pip em um ambiente Python que não o tenha.

Confirme se o Flask está instalado executando o seguinte comando:

```
~$ pip list | grep Flask
```

Se o Flask não estiver instalado, você poderá instalá-lo com o seguinte comando:

```
~$ pip install Flask
```

Etapa 1: criar uma aplicação Python

Crie um diretório do projeto.

```
~$ mkdir eb-python
~$ cd eb-python
```

Crie um exemplo de "Hello Elastic Beanstalk!" Aplicação Python que você implantará usando o Elastic Beanstalk.

Crie um arquivo de texto `application.py` no diretório que você acabou de criar com o seguinte conteúdo.

Example **`~/eb-python/application.py`**

```
from flask import Flask
application = Flask(__name__)

@application.route('/')
def hello_elastic_beanstalk():
        return 'Hello Elastic Beanstalk!'
```

Crie um arquivo de texto `requirements.txt` com a seguinte linha. Esse arquivo contém os pacotes `pip` necessários para a execução da aplicação.

Example **`~/eb-python/requirements.txt`**

```
Flask
```

Etapa 2: executar a aplicação localmente

Execute o seguinte comando para executar sua aplicação localmente.

```
~/eb-python$ export FLASK_APP=application.py && flask run --port 5000
```

Você deve ver saída semelhante ao seguinte

```
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 -
```

Navegue até `http://localhost:5000` no seu navegador da Web. O navegador deve exibir “Hello Elastic Beanstalk!”.

Etapa 3: implantar sua aplicação Python com a EB CLI

Execute os seguintes comandos para criar um ambiente Elastic Beanstalk para essa aplicação.

Para criar um ambiente e implantar sua aplicação Python

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/eb-python$ eb init -p python-3.9 python-tutorial --region us-east-2
   ```

   Esse comando cria uma aplicação `python-tutorial` e configura seu repositório local para criar ambientes com a versão da plataforma Python fornecida.
2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

   ```
   ~/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 ]
   ```

   Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.
3. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

   ```
   ~/eb-python$ eb create python-env
   ```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~/eb-python$ eb open
```

Parabéns! Você implantou uma aplicação Python com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 5: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~/eb-python$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador

de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.

- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

> **Explore mais tutoriais**
>
> Se quiser experimentar outros tutoriais com diferentes aplicações de exemplo, consulte os tutoriais a seguir:
>
> - Implantar uma aplicação Flask no Elastic Beanstalk
> - Implantar uma aplicação Django no Elastic Beanstalk

Depois de implantar um ou dois aplicações de amostra e estiver pronto para começar a desenvolver e executar aplicações Python localmente, consulte Configurar seu ambiente de desenvolvimento Python para o Elastic Beanstalk.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Configurar seu ambiente de desenvolvimento Python para o Elastic Beanstalk

Este tópico fornece instruções para configurar um ambiente de desenvolvimento Python para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

Seções

- Pré-requisitos
- Usar um ambiente virtual
- Configurar um projeto Python para o Elastic Beanstalk

Pré-requisitos

A lista a seguir fornece os pré-requisitos comuns para trabalhar com o Elastic Beanstalk e suas aplicações Python:

- Linguagem Python: instale a versão da linguagem Python que está incluída na versão da plataforma Python do Elastic Beanstalk escolhida. Para obter uma lista de nossas versões de linguagem Python compatíveis, consulte Plataformas Python compatíveis, no guia de Plataformas

do AWS Elastic Beanstalk . Se você ainda não tiver o Python configurado na máquina de desenvolvimento, consulte a página de downloads do Python no site do Python.

- Utilitário **`pip`**: o utilitário `pip` é o instalador de pacotes do Python. Ele instala e lista as dependências do seu projeto, para que o Elastic Beanstalk saiba como configurar o ambiente da aplicação. Para obter mais informações sobre `pip`, consulte a página do pip no site pip.pypa.io.
- (Opcional) A interface de linha de comando do Elastic Beanstalk (EB CLI): a EB CLI pode empacotar sua aplicação com os arquivos de implantação necessários. Ela também pode criar um ambiente Elastic Beanstalk e implantar sua aplicação nele. Você também pode fazer implantações por meio do console do Elastic Beanstalk e, portanto, a EB CLI não é estritamente necessária.
- Uma instalação funcional do **`SSH`**: você pode se conectar a instâncias em execução com o protocolo SSH para examinar ou depurar uma implementação.
- Pacote **`virtualenv`**: essa ferramenta `virtualenv` cria um ambiente de desenvolvimento e teste para sua aplicação. O Elastic Beanstalk pode replicar esse ambiente sem instalar pacotes extras que não são necessários para a aplicação. Para obter mais informações, consulte o site virtualenv. Depois de instalar o Python, você pode instalar o pacote `virtualenv` com o seguinte comando:

```
$ pip install virtualenv
```

Usar um ambiente virtual

Quando você tiver os pré-requisitos instalados, configure um ambiente virtual com `virtualenv` para instalar as dependências de seu aplicativo. Ao usar um ambiente virtual, você pode discernir exatamente quais pacotes são necessários para seu aplicativo, de modo que os pacotes necessários sejam instalados nas EC2 instâncias que estão executando seu aplicativo.

Para configurar um ambiente virtual

1. Abra uma janela de linha de comando e digite:

   ```
   $ virtualenv /tmp/eb_python_app
   ```

   *`eb_python_app`*Substitua por um nome que faça sentido para seu aplicativo (usar o nome do seu aplicativo é uma boa ideia). O comando `virtualenv` cria um ambiente virtual para você no diretório especificado e imprime os resultados de suas ações:

   ```
   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 seu ambiente virtual estiver pronto, inicie-o executando o script `activate` localizado no diretório `bin` do ambiente. Por exemplo, para iniciar o ambiente eb_python_app criado na etapa anterior, você digita:

   ```
   $ source /tmp/eb_python_app/bin/activate
   ```

   O ambiente virtual imprime o nome dele (por exemplo: `(eb_python_app)`) no início de cada prompt de comando, lembrando que você está em um ambiente Python virtual.

3. Para parar de usar seu ambiente virtual e voltar ao interpretador Python padrão do sistema com todas as bibliotecas instaladas, execute o comando `deactivate`.

   ```
   (eb_python_app) $ deactivate
   ```

> **Note**
>
> Após a criação, você poderá reiniciar o ambiente virtual a qualquer momento executando o script `activate` novamente.

Configurar um projeto Python para o Elastic Beanstalk

É possível usar a CLI do Elastic Beanstalk para preparar as aplicações Python para implantação com o Elastic Beanstalk.

Como configurar uma aplicação Python para implantação com o Elastic Beanstalk

1. Em seu ambiente virtual, volte para a parte superior da árvore de diretórios do projeto (`python_eb_app`) e digite:

   ```
   pip freeze >requirements.txt
   ```

Este comando copia os nomes e versões dos pacotes que estão instalados em seu ambiente virtual para `requirements.txt`, Por exemplo, se o pacote PyYAML, versão 3.11 estiver instalado em seu ambiente virtual, o arquivo conterá a linha:

```
PyYAML==3.11
```

Dessa forma, o Elastic Beanstalk pode replicar o ambiente Python da aplicação usando os mesmos pacotes e as mesmas versões que você usou para desenvolver e testar a aplicação.

2. Configure o repositório da EB CLI com o comando eb init. Siga os prompts para escolher a região, a plataforma e outras opções.

Por padrão, o Elastic Beanstalk procura um arquivo chamado `application.py` para iniciar a aplicação. Se ele não existir no projeto Python que você criou, alguns ajustes no ambiente do seu aplicativo serão necessários. Você também precisará definir variáveis de ambiente para que os módulos do aplicativo possam ser carregados. Consulte Usar a plataforma Python do Elastic Beanstalk para obter mais informações.

Usar a plataforma Python do Elastic Beanstalk

Este tópico descreve como configurar, criar e executar suas aplicações Python no Elastic Beanstalk.

AWS Elastic Beanstalk suporta várias ramificações de plataforma para diferentes versões da linguagem de programação Python. Consulte Python no documento Plataformas do AWS Elastic Beanstalk para uma lista completa.

As aplicações Web Python podem ser executadas por trás de um servidor proxy com WSGI. O Elastic Beanstalk fornece o Gunicorn como o servidor WSGI padrão.

Você pode adicionar um `Procfile` ao seu pacote de origem para especificar e configurar o servidor WSGI para seu aplicativo. Para obter detalhes, consulte the section called “Procfile”.

Você pode usar os arquivos `Pipfile.lock` e `Pipfile` criados pelo Pipenv para especificar dependências de pacotes Python e outros requisitos. Para obter detalhes sobre como especificar dependências, consulte the section called “Especificação de dependências”.

O Elastic Beanstalk fornece opções de configuração que você pode usar para personalizar o software executado EC2 nas instâncias do seu ambiente do Elastic Beanstalk. Você pode configurar variáveis de ambiente necessárias à aplicação, habilitar a alternância de logs para o Amazon S3 e

mapear pastas da origem da aplicação que contenham arquivos estáticos para caminhos atendidos pelo servidor de proxy.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Para os pacotes do Python disponíveis no `pip`, você também pode incluir um arquivo de requisitos na raiz do código-fonte da aplicação. O Elastic Beanstalk instala todos os pacotes de dependência especificados em um arquivo de requisitos durante a implantação. Para obter detalhes, consulte the section called “Especificação de dependências”.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Configurar o ambiente Python

As configurações da plataforma Python permitem que você ajuste o comportamento de suas instâncias da Amazon. EC2 Você pode editar a configuração da instância Amazon do ambiente do Elastic Beanstalk usando o console EC2 do Elastic Beanstalk.

Use o console do Elastic Beanstalk para definir AWS X-Ray as configurações de processo do Python, habilitar, habilitar a rotação de logs para o Amazon S3 e configurar variáveis que seu aplicativo possa ler do ambiente.

Como configurar o ambiente Python no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS

2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Configurações do Python

- Proxy server (Servidor de proxy): o servidor de proxy a ser usado nas instâncias de ambiente. Por padrão, é usado nginx.
- WSGI Path (Caminho do WSGI): o nome ou o caminho para o arquivo de aplicação principal. Por exemplo, `application.py` ou `django/wsgi.py`.
- NumProcesses— O número de processos a serem executados em cada instância do aplicativo.
- NumThreads— O número de threads a serem executados em cada processo.

AWS X-Ray configurações

- Daemon X-Ray — Execute o AWS X-Ray daemon para processar dados de rastreamento do. AWS X-Ray SDK for Python

Opções de log

A seção Log Options tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo

no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Por padrão, o servidor de proxy em um ambiente Python atende todos os arquivos de uma pasta chamada `static` no caminho `/static`. Por exemplo, se a origem do aplicativo contiver um arquivo chamado `logo.png` em uma pasta chamada `static`, o servidor de proxy o servirá para os usuários em *`subdomain`*`.elasticbeanstalk.com/static/logo.png`. Você pode configurar mapeamentos adicionais conforme explicado nesta seção.

Propriedades de ambiente

Você pode usar as propriedades de ambiente para fornecer informações ao aplicativo e configurar variáveis de ambiente. Por exemplo, crie uma propriedade de ambiente chamada `CONNECTION_STRING` que especifica uma string de conexão que o aplicativo pode usar para se conectar a um banco de dados.

No ambiente do Python em execução no Elastic Beanstalk, esses valores podem ser acessados usando o dicionário do Python `os.environ`. Para obter mais informações, consulte http://docs.python. org/library/os.html.

Você pode usar um código semelhante ao mostrado abaixo para acessar as chaves e os parâmetros:

```
import os
endpoint = os.environ['API_ENDPOINT']
```

As propriedades de ambiente também podem fornecer informações para uma framework. Por exemplo, crie uma propriedade chamada `DJANGO_SETTINGS_MODULE` para configurar o Django a usar um módulo de configurações específicas. Dependendo do ambiente, o valor pode ser `development.settings`, `production.settings` etc.

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespaces de configuração do Python

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser

específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

A plataforma Python define opções nos namespaces `aws:elasticbeanstalk:environment:proxy`, `aws:elasticbeanstalk:environment:proxy:staticfiles` e `aws:elasticbeanstalk:container:python`.

O arquivo de configuração de exemplo a seguir especifica as definições da opção de configuração para criar uma propriedade de ambiente chamada `DJANGO_SETTINGS_MODULE`, selecionar o servidor de proxy Apache, especificar duas opções de arquivos estáticos que mapeiam um diretório chamado `statichtml` para o caminho `/html` e um diretório chamado `staticimages` para o caminho `/images` e especificar configurações adicionais no namespace `aws:elasticbeanstalk:container:python`. Esse namespace contém opções que permitem especificar a localização do script de WSGI no código-fonte e o número de threads e processos para executar no 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
```

Observações

- Se você estiver usando uma versão de plataforma Python da AMI do Amazon Linux (anterior ao Amazon Linux 2), substitua o valor `WSGIPath` por `ebdjango/wsgi.py`. O valor no exemplo funciona com o servidor WSGI do Gunicorn, que não é mais compatível nas versões da plataforma da AMI do Amazon Linux.
- Além disso, essas versões de plataforma mais antigas usam um namespace diferente para configurar arquivos estáticos:

`aws:elasticbeanstalk:container:python:staticfiles`. Ele tem os mesmos nomes de opção e semântica que o namespace de arquivo estático padrão.

Os arquivos de configuração também são compatíveis com várias chaves para modificar mais o software nas instâncias do ambiente. Este exemplo usa a chave dos pacotes para instalar o Memcached com `yum` e comandos do contêiner para executar comandos que configuram o servidor durante a implantação:

```
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"
```

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

O `python3` executável

A versão do `python3` executável disponível em EC2 instâncias nos ambientes Python do Elastic Beanstalk nem sempre corresponderá à mesma versão do Python usada pela plataforma. Por exemplo, na plataforma Python 3.12 AL2 023, aponta `/usr/bin/python3` para Python 3.9. Isso ocorre porque o Python 3.9 é o sistema Python em 023. AL2 Para obter mais informações, consulte Python em AL2 023 no Guia do usuário do Amazon Linux 2023. Você pode acessar um executável correspondente à versão do Python usada pela plataforma em um local versionado (por `/usr/bin/`

`python3.12` exemplo) ou no diretório do `bin` ambiente virtual do aplicativo (por exemplo). `/var/app/venv/staging-LQM1lest/bin/python3` A plataforma usa o executável correto do Python que corresponde à ramificação da plataforma.

Configuração do servidor WSGI com um Procfile no Elastic Beanstalk

Você pode adicionar um `Procfile` ao seu pacote de origem para especificar e configurar o servidor WSGI para seu aplicativo. É possível especificar comandos personalizados de inicialização e execução no `Procfile`.

Quando você usa um `Procfile`, ele substitui as opções de namespace `aws:elasticbeanstalk:container:python` definidas usando arquivos de configuração.

O exemplo a seguir usa um `Procfile` para especificar uWSGI como o servidor e configurá-lo.

Example Procfile

```
web: uwsgi --http :8000 --wsgi-file application.py --master --processes 4 --threads 2
```

O exemplo a seguir usa um `Procfile` para configurar o Gunicorn, o servidor WSGI padrão.

Example Procfile

```
web: gunicorn --bind :8000 --workers 3 --threads 2 project.wsgi:application
```

> **Observações**
>
> - Se você configurar qualquer servidor WSGI que não seja Gunicorn, certifique-se de também especificá-lo como uma dependência do seu aplicativo, para que ele seja instalado em suas instâncias de ambiente. Para obter detalhes sobre a especificação de dependência, consulte the section called “Especificação de dependências”.
> - A porta padrão para o servidor WSGI é 8000. Se você especificar um número de porta diferente no comando `Procfile`, defina a propriedade do ambiente `PORT` para esse número de porta também.

Especificação de dependências usando um arquivo de requisitos no Elastic Beanstalk

Este tópico descreve como configurar sua aplicação para instalar outros pacotes Python necessários. Um aplicativo em Python típico tem dependências em outros pacotes Python de terceiros. Com a plataforma Python do Elastic Beanstalk, você tem várias maneiras de especificar pacotes Python dos quais sua aplicação depende.

Utilizar **`pip`** e **`requirements.txt`**

A ferramenta padrão para instalar pacotes Python é `pip`. Ele tem um recurso que permite especificar todos os pacotes necessários (e suas versões) em um único arquivo de requisitos. Para obter mais informações sobre o arquivo de requisitos, consulte Formato do arquivo de requisitos no site de documentação do pip.

Crie um arquivo chamado `requirements.txt` e coloque-o no diretório de nível superior do pacote de origem. Veja a seguir um exemplo do arquivo `requirements.txt` para Django.

```
Django==2.2
mysqlclient==2.0.3
```

Em seu ambiente de desenvolvimento, use o comando `pip freeze` para gerar o arquivo de requisitos.

```
~/my-app$ pip freeze > requirements.txt
```

Para garantir que o arquivo de requisitos contenha apenas os pacotes realmente usados pelo aplicativo, use um ambiente virtual que tenha somente esses pacotes instalados. Fora de um ambiente virtual, a saída de `pip freeze` incluirá todos os pacotes do `pip` instalados na máquina de desenvolvimento, bem como aqueles que acompanham o sistema operacional.

> **Note**
>
> Nas versões da plataforma Python da AMI do Amazon Linux, o Elastic Beanstalk não é compatível com Pipenv nem Pipfiles. Se você usar Pipenv para gerenciar as dependências do aplicativo, execute o seguinte comando para gerar um arquivo `requirements.txt`.
>
> ```
> ~/my-app$ pipenv lock -r > requirements.txt
> ```

Para saber mais, consulte Gerar um requirements.txt na documentação do Pipenv.

Usar Pipenv e `Pipfile`

Pipenv é uma ferramenta moderna de empacotamento Python. Ele combina a instalação do pacote com a criação e gerenciamento de um arquivo de dependência e um virtualenv para o seu aplicativo. Para obter mais informações, consulte Pipenv: Python Dev Workflow for Humans.

O Pipenv mantém dois arquivos:

- `Pipfile` — Esse arquivo contém vários tipos de dependências e requisitos.
- `Pipfile.lock` — Esse arquivo contém um instantâneo da versão que permite compilações determinísticas.

Você pode criar esses arquivos em seu ambiente de desenvolvimento e incluí-los no diretório de nível superior do pacote de origem implantado no Elastic Beanstalk. Para obter mais informações sobre esses dois arquivos, consulte Exemplo de Pipfile e Pipfile.lock.

O exemplo a seguir usa o Pipenv para instalar o Django e a framework REST do Django. Esses comandos criam os arquivos `Pipfile` e `Pipfile.lock`.

```
~/my-app$ pipenv install django
~/my-app$ pipenv install djangorestframework
```

Precedência

Se você incluir mais de um dos arquivos de requisitos descritos neste tópico, o Elastic Beanstalk usará apenas um deles. A lista a seguir mostra a precedência, em ordem decrescente.

1. `requirements.txt`
2. `Pipfile.lock`
3. `Pipfile`

> **Note**
>
> A partir da versão da plataforma Amazon Linux 2 de 7 de março de 2023, se você fornecer mais de um desses arquivos, o Elastic Beanstalk emitirá uma mensagem do console informando qual dos arquivos de dependência foi usado durante uma implantação.

As etapas a seguir descrevem a lógica que o Elastic Beanstalk segue para instalar as dependências ao implantar uma instância.

- Se houver um arquivo `requirements.txt`, usamos o comando `pip install -r requirements.txt`.
- A partir do lançamento da plataforma Amazon Linux 2 em 7 de março de 2023, se não houver nenhum arquivo `requirements.txt`, mas houver um `Pipfile.lock`, usamos o comando `pipenv sync`. Antes desse lançamento, usamos `pipenv install --ignore-pipfile`.
- Se não houver um arquivo `requirements.txt` nem um `Pipfile.lock`, mas houver um `Pipfile`, usamos o comando `pipenv install --skip-lock`.
- Se nenhum dos três arquivos de requisitos for encontrado, não instalaremos nenhuma dependência do aplicativo.

Implantar uma aplicação Flask no Elastic Beanstalk

Este tutorial mostra o processo de geração de um aplicativo Flask e sua implantação em um AWS Elastic Beanstalk ambiente. O Flask é uma framework de aplicativo web de código aberto para Python.

Neste tutorial, você fará o seguinte:

- Configurar um ambiente Python virtual com Flask
- Criar uma aplicação Flask
- Implantar o site com a CLI do EB
- Limpeza

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

O Flask requer o Python 3.7 ou posterior. Neste tutorial, usamos Python 3.7 e a versão correspondente da plataforma do Elastic Beanstalk. Instale o Python seguindo as instruções em Configurar seu ambiente de desenvolvimento Python para o Elastic Beanstalk.

O framework Flask será instalado como parte do tutorial.

Este tutorial também usa a interface da linha de comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Configurar um ambiente Python virtual com Flask

Crie um diretório de projeto e um ambiente virtual para seu aplicativo e instale o Flask.

Para configurar o ambiente de projetos

1. Crie um diretório do projeto.

   ```
   ~$ mkdir eb-flask
   ~$ cd eb-flask
   ```

2. Crie e ative um ambiente virtual denominado `virt`:

   ```
   ~/eb-flask$ virtualenv virt
   ```

```
~$ source virt/bin/activate
(virt) ~/eb-flask$
```

Você verá `(virt)` acrescentado ao prompt de comando, indicando que você está em um ambiente virtual. Use o ambiente virtual para o restante deste tutorial.

3. Instale o Flask com `pip install`:

```
(virt)~/eb-flask$ pip install flask==2.0.3
```

4. Veja as bibliotecas instaladas com `pip freeze`:

```
(virt)~/eb-flask$ pip freeze
click==8.1.1
Flask==2.0.3
itsdangerous==2.1.2
Jinja2==3.1.1
MarkupSafe==2.1.1
Werkzeug==2.1.0
```

Esse comando relaciona todos os pacotes instalados em seu ambiente virtual. Como você está em um ambiente virtual, os pacotes instalados globalmente, como a EB CLI, não são mostrados.

5. Salve a saída de `pip freeze` como um arquivo chamado `requirements.txt`.

```
(virt)~/eb-flask$ pip freeze > requirements.txt
```

Esse arquivo informa o Elastic Beanstalk para instalar as bibliotecas durante a implantação. Para obter mais informações, consulte Especificação de dependências usando um arquivo de requisitos no Elastic Beanstalk.

Criar uma aplicação Flask

Depois, crie uma aplicação a ser implantada usando o Elastic Beanstalk. Criaremos um serviço RESTful web “Hello World”.

Crie um novo arquivo de texto nesse diretório chamado `application.py` com o seguinte conteúdo:

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()
```

Este exemplo imprime uma saudação personalizada que varia de acordo com o caminho usado para acessar o serviço.

> **Note**
>
> Com a adição de `application.debug = True` antes de executar o aplicativo, a saída da depuração é habilitada para o caso de algo der errado. É uma boa prática de

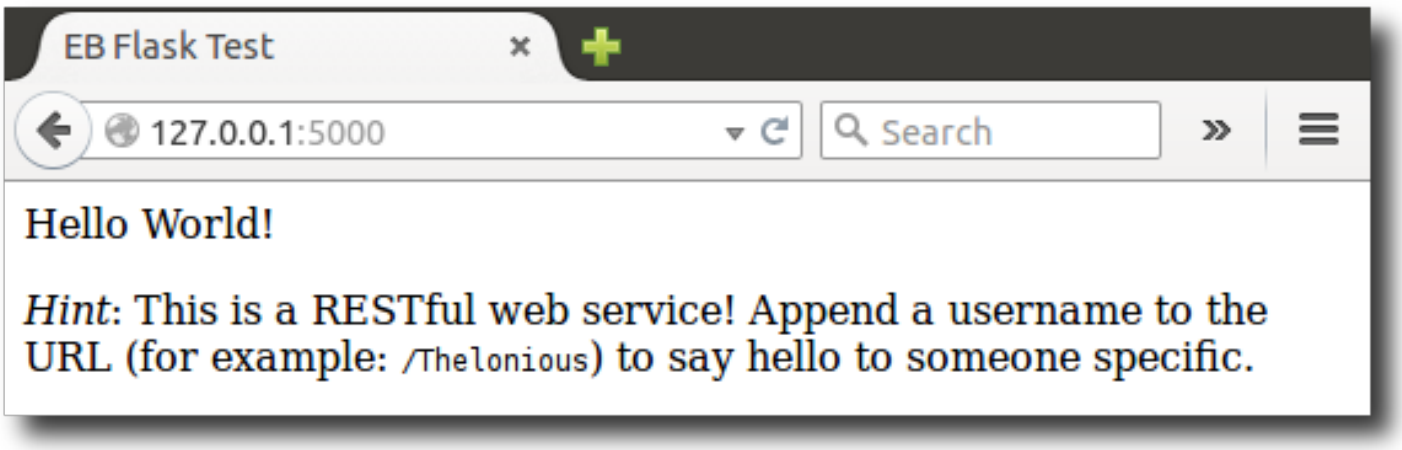
desenvolvimento, mas você deve remover instruções de depuração no código de produção, já que a saída da depuração pode revelar aspectos internos do seu aplicativo.

Use `application.py` como o nome do arquivo e forneça um objeto que pode ser chamado `application` (o objeto Flask, neste caso) para que o Elastic Beanstalk possa localizar facilmente o código da aplicação.

Execute `application.py` com 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
```

Abra `http://127.0.0.1:5000/` em seu navegador da web. Você deve ver o aplicativo em execução, mostrando a página de índice:



Verifique o log do servidor para ver a saída da sua solicitação. Você pode interromper o servidor web e retornar ao ambiente virtual pressionando Ctrl+C.

Se você obteve a saída da depuração em vez disso, corrija os erros e verifique se a aplicação é executada localmente antes de configurá-la para o Elastic Beanstalk.

Implantar o site com a CLI do EB

Você adicionou tudo o que precisa para implantar a aplicação no Elastic Beanstalk. O diretório do projeto deve ter esta aparência:

```
~/eb-flask/
|-- virt
|-- application.py
`-- requirements.txt
```

No entanto, a pasta `virt` não é necessária para que a aplicação seja executada no Elastic Beanstalk. Quando você faz a implantação, o Elastic Beanstalk cria um ambiente virtual nas instâncias do servidor e instala as bibliotecas listadas em `requirements.txt`. Para minimizar o tamanho do pacote de origem que você carregou durante a implantação, adicione um arquivo .ebignore que instrui a EB CLI a deixar a pasta `virt` de fora.

Example ~/eb-flask/.ebignore

```
virt
```

Depois, você criará o ambiente da aplicação e implantará a aplicação configurada com o Elastic Beanstalk.

Para criar um ambiente e implantar o aplicativo Flask

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/eb-flask$ eb init -p python-3.7 flask-tutorial --region us-east-2
   Application flask-tutorial has been created.
   ```

   Esse comando cria uma nova aplicação nomeada `flask-tutorial` e configura seu repositório local para criar ambientes com a versão da plataforma Python 3.7 mais recente.
2. (opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa se conectar à EC2 instância que executa seu aplicativo com 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 ]
```

Selecione um par de chaves se você já tiver um ou siga os prompts para criar um novo. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante seu aplicativo nele com eb create:

```
~/eb-flask$ eb create flask-env
```

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.

- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

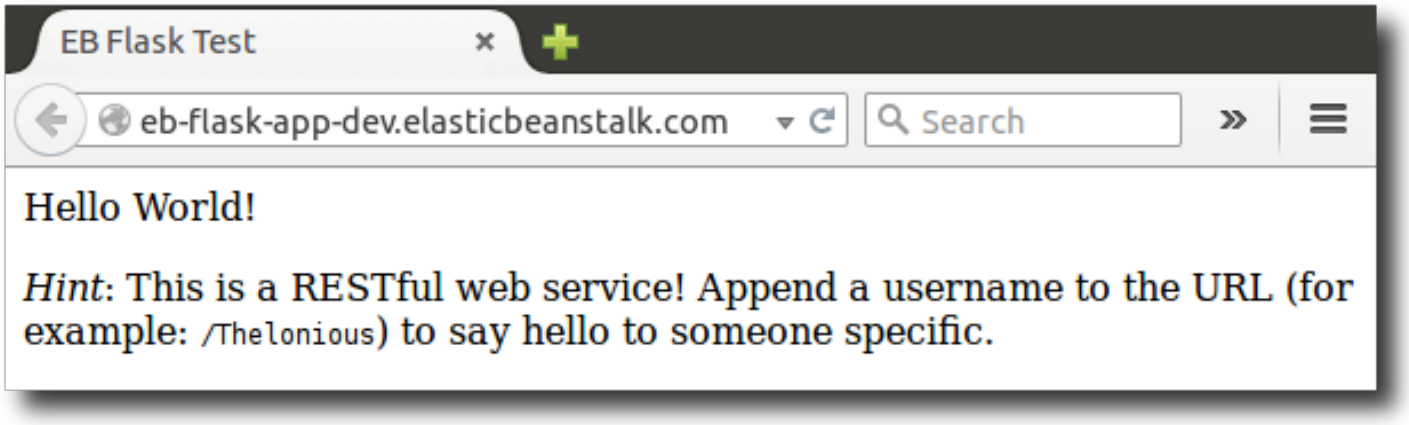
Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Quando o processo de criação do ambiente for concluído, abra o site com eb open:

```
~/eb-flask$ eb open
```

Isso abrirá uma janela do navegador usando o nome de domínio criado para o seu aplicativo. Você deve ver o mesmo site Flask que criou e testou localmente.



Se o aplicativo não aparecer em execução ou se houver uma mensagem de erro, consulte Solução de erros de implantação para obter ajuda sobre como determinar a causa do erro.

Caso veja a aplicação em execução, você implantou sua primeira aplicação Flask com o Elastic Beanstalk. Parabéns!

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

ou com a EB CLI:

```
~/eb-flask$ eb terminate flask-env
```

Próximas etapas

Para mais informações sobre o Flask, acesse flask.pocoo.org.

Se você deseja testar outra framework da Web para Python, consulte Implantar uma aplicação Django no Elastic Beanstalk.

Implantar uma aplicação Django no Elastic Beanstalk

Este tutorial explica o processo de implantação de um site Django padrão gerado automaticamente para ambientes do AWS Elastic Beanstalk que executam Python. Este tutorial mostra como hospedar uma aplicação Web Python na nuvem usando um ambiente do Elastic Beanstalk.

Neste tutorial, você fará o seguinte:

- Configurar um ambiente virtual Python e instalar o Django
- Criar um projeto Django
- Configurar a aplicação Django para o Elastic Beanstalk
- Implantar o site com a CLI do EB
- Atualizar seu aplicativo
- Limpeza

Pré-requisitos

Para seguir este tutorial, você deve ter todos os Pré-requisitos comuns para Python instalados, incluindo os seguintes pacotes:

- Python 3.7 ou posterior
- `pip`
- `virtualenv`
- `awsebcli`

A framework do Django será instalada como parte do tutorial.

> **Note**
>
> A criação de ambientes com a CLI do EB exige uma função de serviço. É possível criar uma função de serviço criando um ambiente no console do Elastic Beanstalk. Se você não tiver uma função de serviço, a CLI do EB tentará criar uma ao executar `eb create`.

Configurar um ambiente virtual Python e instalar o Django

Crie um ambiente virtual com `virtualenv` e use-o para instalar o Django e suas dependências. Ao usar um ambiente virtual, você pode saber exatamente quais pacotes seu aplicativo precisa, para que os pacotes necessários sejam instalados nas EC2 instâncias da Amazon que estão executando seu aplicativo.

As etapas a seguir demonstram os comandos que você deve inserir para sistemas baseados em Unix e Windows, mostrados em guias separadas.

Para configurar o ambiente virtual

1. Crie um ambiente virtual denominado `eb-virt`.

   Unix-based systems

   ```
   ~$ virtualenv ~/eb-virt
   ```

   Windows

   ```
   C:\> virtualenv %HOMEPATH%\eb-virt
   ```

2. Ative o ambiente virtual.

   Unix-based systems

   ```
   ~$ source ~/eb-virt/bin/activate
   (eb-virt) ~$
   ```

   Windows

   ```
   C:\>%HOMEPATH%\eb-virt\Scripts\activate
   ```

```
(eb-virt) C:\>
```

Você verá `(eb-virt)` acrescentado ao começo do prompt de comando, indicando que você está em um ambiente virtual.

> **Note**
>
> O restante das instruções mostra o prompt de comando do Linux no diretório inicial ~$. No Windows, ou seja`C:\Users\`*USERNAME*>, onde *USERNAME* está seu nome de login do Windows.

3. Use `pip` para instalar o Django.

```
(eb-virt)~$ pip install django==2.2
```

> **Note**
>
> A versão do Django que você instalar deve ser compatível com a versão do Python na configuração do Python do Elastic Beanstalk que você escolher para a implantação de seu aplicativo. Para obter informações sobre implantação, consulte ??? neste tópico.
> Para obter mais informações sobre as versões atuais da plataforma Python, consulte Python no documento Plataformas do AWS Elastic Beanstalk .
> Para obter informações sobre a compatibilidade de versão do Django com o Python, consulte Qual versão do Python posso usar com o Django?

4. Para verificar se o Django está instalado, insira o seguinte.

```
(eb-virt)~$ pip freeze
Django==2.2
...
```

Esse comando relaciona todos os pacotes instalados em seu ambiente virtual. Posteriormente, você usará a saída desse comando para configurar o projeto para uso com o Elastic Beanstalk.

Criar um projeto Django

Agora você está pronto para criar um projeto Django e executá-lo em sua máquina, usando o ambiente virtual.

> **Note**
>
> Este tutorial usa SQLite, que é um mecanismo de banco de dados incluído no Python. O banco de dados é implantado com seus arquivos de projeto. Para ambientes de produção, recomendamos que você use o Amazon Relational Database Service (Amazon RDS) e que você o separe do ambiente. Para obter mais informações, consulte Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Python Elastic Beanstalk.

Para gerar um aplicativo Django

1. Ative seu ambiente virtual.

   Unix-based systems

   ```
   ~$ source ~/eb-virt/bin/activate
   (eb-virt) ~$
   ```

   Windows

   ```
   C:\>%HOMEPATH%\eb-virt\Scripts\activate
   (eb-virt) C:\>
   ```

   Você verá o prefixo `(eb-virt)` acrescentado ao prompt de comando, indicando que você está em um ambiente virtual.

   > **Note**
   >
   > O restante das instruções mostra o prompt de comando do Linux ~$ em seu diretório inicial e no diretório inicial do Linux `~/`. No Windows`C:\Users\`*USERNAME*>, onde *USERNAME* está seu nome de login do Windows.

2. Use o comando `django-admin startproject` para criar um projeto Django denominado `ebdjango`:

```
(eb-virt)~$ django-admin startproject ebdjango
```

Este comando cria um site Django padrão denominado ebdjango com a seguinte estrutura de diretório.

```
~/ebdjango
  |-- ebdjango
  |   |-- __init__.py
  |   |-- settings.py
  |   |-- urls.py
  |   `-- wsgi.py
  `-- manage.py
```

3. Execute o site Django localmente com `manage.py runserver`:

```
(eb-virt) ~$ cd ebdjango
```

```
(eb-virt) ~/ebdjango$ python manage.py runserver
```

4. Em um navegador da web, abra `http://127.0.0.1:8000/` para visualizar o site.
5. Verifique o log do servidor para ver a saída da sua solicitação. Para interromper o servidor web e retornar ao seu ambiente virtual, pressione 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
```

Configurar a aplicação Django para o Elastic Beanstalk

Agora que você tem um site desenvolvido pelo Django em sua máquina local, você pode configurá-lo para implantação com o Elastic Beanstalk.

Por padrão, o Elastic Beanstalk procura um arquivo chamado `application.py` para iniciar sua aplicação. Como isso não existe no projeto Django que você criou, é necessário fazer alguns ajustes

no ambiente do aplicativo. Também é necessário definir variáveis de ambiente para que os módulos do aplicativo possam ser carregados.

Como configurar seu site para o Elastic Beanstalk

1. Ative seu ambiente virtual.

   Unix-based systems

   ```
   ~/ebdjango$ source ~/eb-virt/bin/activate
   ```

   Windows

   ```
   C:\Users\USERNAME\ebdjango>%HOMEPATH%\eb-virt\Scripts\activate
   ```

2. Execute `pip freeze` e salve a saída em um arquivo chamado `requirements.txt`.

   ```
   (eb-virt) ~/ebdjango$ pip freeze > requirements.txt
   ```

   O Elastic `requirements.txt` Beanstalk usa para determinar qual pacote instalar EC2 nas instâncias que executam seu aplicativo.

3. Crie um diretório chamado `.ebextensions`.

   ```
   (eb-virt) ~/ebdjango$ mkdir .ebextensions
   ```

4. No diretório `.ebextensions`, adicione um arquivo de configuração chamado `django.config` com o texto a seguir.

   Example ~/ebdjango/.ebextensions/django.config

   ```
   option_settings:
     aws:elasticbeanstalk:container:python:
       WSGIPath: ebdjango.wsgi:application
   ```

   Essa configuração, `WSGIPath`, especifica o local do script WSGI que o Elastic Beanstalk usa para iniciar a aplicação.

> **Note**
>
> Se você estiver usando uma versão de plataforma Python da AMI do Amazon Linux (anterior ao Amazon Linux 2), substitua o valor `WSGIPath` por `ebdjango/wsgi.py`. O valor no exemplo funciona com o servidor WSGI do Gunicorn, que não é mais compatível nas versões da plataforma da AMI do Amazon Linux.

5. Use o comando `deactivate` para desativar o ambiente virtual.

```
(eb-virt) ~/ebdjango$ deactivate
```

Reative o ambiente virtual sempre que for necessário adicionar pacotes ao aplicativo ou executá-lo localmente.

Implantar o site com a CLI do EB

Você adicionou tudo o que precisa para implantar a aplicação no Elastic Beanstalk. O diretório do projeto deve ter esta aparência:

```
~/ebdjango/
|-- .ebextensions
|   `-- django.config
|-- ebdjango
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
|-- db.sqlite3
|-- manage.py
`-- requirements.txt
```

Depois, você criará o ambiente da aplicação e implantará a aplicação configurada com o Elastic Beanstalk.

Imediatamente após a implantação, edite a configuração do Django para adicionar o nome de domínio que o Elastic Beanstalk atribuiu à aplicação para os `ALLOWED_HOSTS` do Django. Depois, reimplante o aplicativo. Este é um requisito de segurança do Django, projetado para impedir ataques

ao cabeçalho de `Host` HTTP. Para obter mais informações, consulte Validação do cabeçalho do host.

Para criar um ambiente e implantar o aplicativo Django

> **Note**
>
> O tutorial usa a CLI do EB como um mecanismo de implantação, mas também é possível usar o console do Elastic Beanstalk para implantar um arquivo zip com o conteúdo do seu projeto.

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/ebdjango$ eb init -p python-3.7 django-tutorial
   Application django-tutorial has been created.
   ```

   Esse comando cria um aplicativo chamado `django-tutorial`. Ele também configura o seu repositório local para criar ambientes com a versão mais recente da plataforma Python 3.7.

2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

   ```
   ~/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 ]
   ```

   Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante o aplicativo nele com eb create.

   ```
   ~/ebdjango$ eb create django-env
   ```

> **Note**
>
> Se você vir uma mensagem de erro "service role required", execute `eb create` interativamente (sem especificar um nome de ambiente) e a CLI do EB cria a função para você.

Esse comando cria um ambiente do Elastic Beanstalk com carga balanceada chamado `django-env`. A criação do ambiente leva cerca de 5 minutos. Como o Elastic Beanstalk cria os recursos necessários para executar a aplicação, ele gera mensagens informativas que a CLI do EB transmite ao terminal.

4. Quando o processo de criação do ambiente for concluído, localize o nome de domínio do seu novo ambiente executando eb status.

```
~/ebdjango$ eb status
Environment details for: django-env
  Application name: django-tutorial
  ...
  CNAME: eb-django-app-dev.elasticbeanstalk.com
  ...
```

Seu nome de domínio do ambiente é o valor da propriedade `CNAME`.

5. Abra o arquivo `settings.py` no diretório `ebdjango`. Localize a configuração `ALLOWED_HOSTS` e adicione o nome de domínio do aplicativo que você encontrou na etapa anterior ao valor da configuração. Se você não encontrar essa configuração no arquivo, adicione-a em uma nova linha.

```
...
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

6. Salve o arquivo e, em seguida, implante o aplicativo executando eb deploy. Quando você executa eb deploy, a EB CLI empacota o conteúdo do diretório do projeto e implanta-o em seu ambiente.

```
~/ebdjango$ eb deploy
```

> **Note**
>
> Se você estiver usando o Git com seu projeto, consulte Usar a EB CLI com o Git.

7. Quando o processo de atualização do ambiente for concluído, abra o site com eb open.

```
~/ebdjango$ eb open
```

Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo. Você deve ver o mesmo site Django que você criou e testou localmente.

Se o aplicativo não aparecer em execução ou se houver uma mensagem de erro, consulte Solução de erros de implantação para obter ajuda sobre como determinar a causa do erro.

Se a aplicação aparecer em execução, você implantou sua primeira aplicação Django com o Elastic Beanstalk. Parabéns!

Atualizar seu aplicativo

Agora que você tem uma aplicação em execução no Elastic Beanstalk, pode atualizar e reimplantar a aplicação ou sua configuração, e o Elastic Beanstalk fará o trabalho de atualizar suas instâncias e iniciar sua nova versão da aplicação.

Para este exemplo, vamos habilitar o console de administração do Django e definir algumas outras configurações.

Modificar as configurações do site

Por padrão, o site Django usa o fuso horário UTC para exibir a hora. Você pode alterá-lo especificando um fuso horário em `settings.py`.

Para alterar o fuso horário do site

1. Modifique a configuração `TIME_ZONE` em `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
```

Para obter uma lista de fusos horários, visite esta página.

2. Implante a aplicação no ambiente do Elastic Beanstalk.

```
~/ebdjango/$ eb deploy
```

Criar um administrador de site

É possível criar um administrador de site para o aplicativo Django para acessar o console de administração diretamente do site. Os detalhes de login do administrador são armazenados de forma segura na imagem do banco de dados local incluída no projeto padrão gerado pelo Django.

Para criar um administrador de site

1. Inicialize o banco de dados local do aplicativo 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. Execute `manage.py createsuperuser` para criar um administrador.

```
(eb-virt) ~/ebdjango$ python manage.py createsuperuser
Username: admin
Email address: me@mydomain.com
Password: ********
Password (again): ********
Superuser created successfully.
```

3. Para informar ao Django onde armazenar os arquivos estáticos, defina `STATIC_ROOT` em `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. Execute `manage.py collectstatic` para preencher o `static` diretório com ativos estáticos (JavaScriptCSS e imagens) para o site de administração.

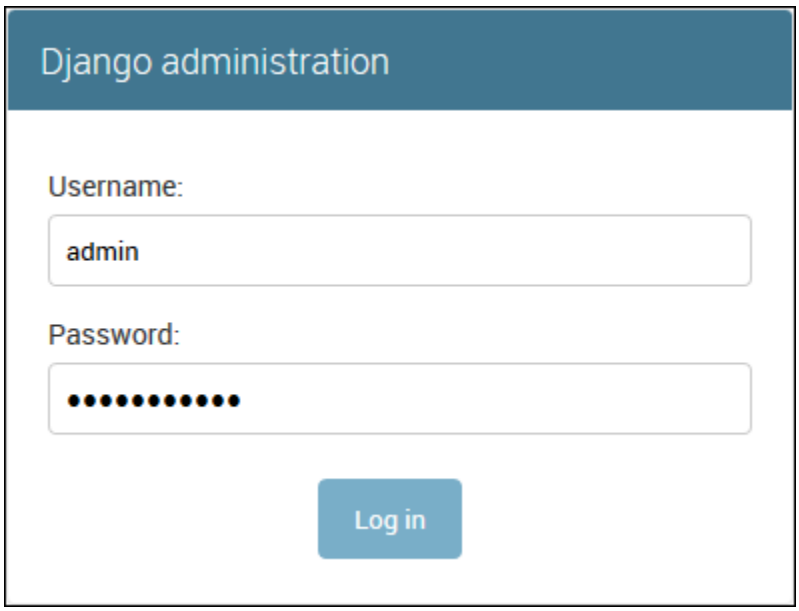
```
(eb-virt) ~/ebdjango$ python manage.py collectstatic
119 static files copied to ~/ebdjango/static
```

5. Implante o aplicativo.

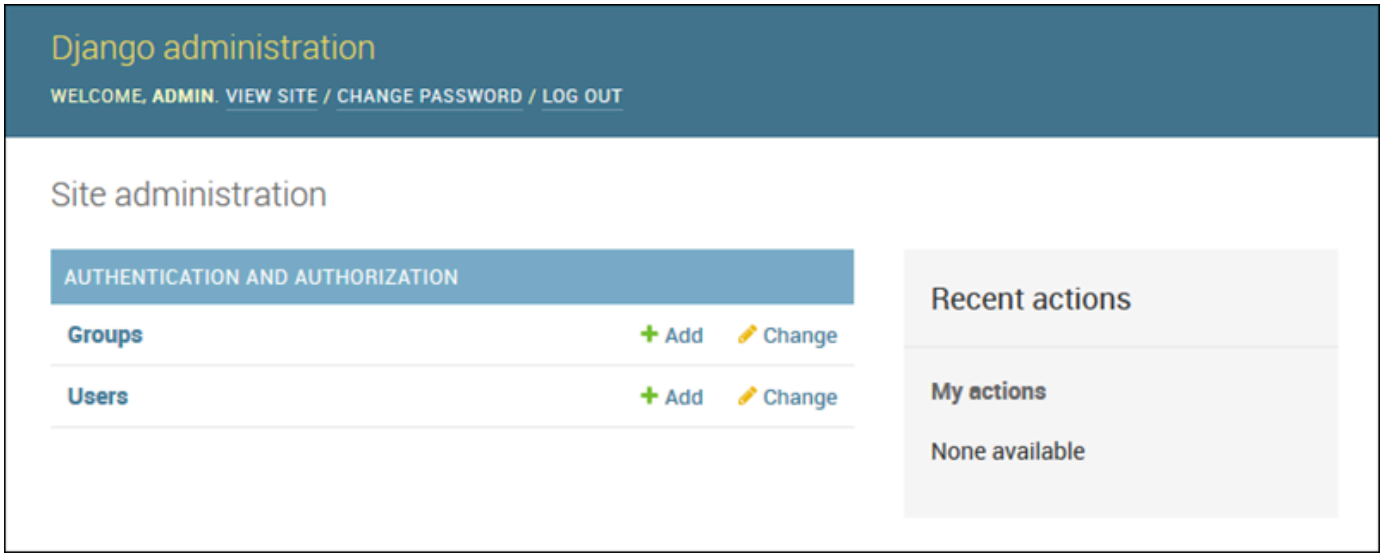
```
~/ebdjango$ eb deploy
```

6. Exiba o console de administração abrindo o site em seu navegador, anexando `/admin/` ao URL do site, como o seguinte:

```
http://djang-env.p33kq46sfh.us-west-2.elasticbeanstalk.com/admin/
```



7. Faça login com o nome de usuário e a senha que você configurou na etapa 2.



Você pode usar um procedimento semelhante ao local updating/testing seguido poreb deploy. O Elastic Beanstalk cuida do processo de atualizar seus servidores em tempo real para que você possa se dedicar ao desenvolvimento de aplicações, em vez da administração de servidor.

Adicionar um arquivo de configuração de migração de banco de dados

É possível adicionar comandos ao script `.ebextensions` que serão executados quando o site for atualizado. Isso permite gerar automaticamente migrações de banco de dados.

Para adicionar uma etapa de migração quando o aplicativo é implantado

1. Crie um arquivo de configuração chamado `db-migrate.config` com o conteúdo a seguir.

   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
   ```

   Esse arquivo de configuração ativa o ambiente virtual do servidor e executa o comando `manage.py migrate` durante o processo de implantação, antes de iniciar a sua aplicação. Como ele é executado antes da inicialização do aplicativo, também é necessário configurar a variável de ambiente `DJANGO_SETTINGS_MODULE` explicitamente (em geral, o `wsgi.py` cuida desse procedimento para você durante o startup). A especificação de `leader_only: true` no comando garante que ele seja executado somente uma vez quando você faz a implantação em várias instâncias.
2. Implante o aplicativo.

   ```
   ~/ebdjango$ eb deploy
   ```

Limpeza

Para economizar horas de instância e outros AWS recursos entre as sessões de desenvolvimento, encerre seu ambiente do Elastic Beanstalk com. eb terminate

```
~/ebdjango$ eb terminate django-env
```

Esse comando encerra o ambiente e todos os AWS recursos executados nele. Ele não exclui o aplicativo, mas você sempre pode criar mais ambientes com a mesma configuração executando eb create novamente.

Se você já concluiu o aplicativo de exemplo, também pode remover a pasta do projeto e o ambiente virtual.

```
~$ rm -rf ~/eb-virt
~$ rm -rf ~/ebdjango
```

Próximas etapas

Para obter mais informações sobre o Django, incluindo um tutorial detalhado, consulte a documentação oficial.

Se você deseja testar outra framework da Web para Python, consulte Implantar uma aplicação Flask no Elastic Beanstalk.

Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Python Elastic Beanstalk

Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Nestas instruções, o banco de dados é acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Seções

- Adicionar uma instância de banco de dados ao seu ambiente
- Download de um driver
- Conectar-se a um banco de dados

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).

5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
`RDS_PASSWORD`	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Download de um driver

Adicione o driver de banco de dados ao arquivo de requisitos de seu projeto.

Example requirements.txt: Django com MySQL

```
Django==2.2
mysqlclient==2.0.3
```

Pacotes de drivers comuns para Python

- MySQL – `mysqlclient`
- PostgreSQL – `psycopg2`
- Oracle: `cx_Oracle`
- SQL Server – `adodbapi`

Para obter mais informações, consulte Python DatabaseInterfaces e Django 2.2 - bancos de dados suportados.

Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `os.environ['VARIABLE']` para ler as propriedades e configurar uma conexão de banco de dados.

Example Arquivo de configurações Django: dicionário de BANCOS DE DADOS

```
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'],
        }
```

```
    }
```

Ferramentas e recursos do Python

Há vários lugares que você pode acessar para obter ajuda adicional durante o desenvolvimento de seus aplicativos Python:

Recurso	Descrição
AWS SDK para Python (Boto3) em GitHub	Instale o Boto3 proveniente de. GitHub
AWS SDK para Python (Boto3) página inicial	A AWS SDK para Python (Boto3) página inicial.
Centro de desenvolvedores Python	Um local central para acessar código de exemplo, documentação, ferramentas e outros recursos.

Implantação de aplicações Ruby com o Elastic Beanstalk

Este capítulo fornece instruções para configurar e implantar seu aplicativo web Ruby no. AWS Elastic Beanstalk O Elastic Beanstalk facilita a implantação, o gerenciamento e o dimensionamento das suas aplicações Web Ruby usando a Amazon Web Services.

Você pode implantar sua aplicação em apenas alguns minutos usando a Interface de Linha de Comando (EB CLI) do Elastic Beanstalk ou o console do Elastic Beanstalk. Depois de implantar seu aplicativo Elastic Beanstalk, você pode continuar usando o EB CLI para gerenciar seu aplicativo e ambiente, ou você pode usar o console do Elastic Beanstalk, ou o. AWS CLI APIs

Este capítulo também fornece step-by-step instruções para implantar um aplicativo de amostra no Elastic Beanstalk usando a CLI do EB e, em seguida, atualizar o aplicativo para usar as estruturas de aplicativos web Rails e Sinatra.

Os tópicos neste capítulo pressupõem que você tem um nível básico de conhecimento sobre ambientes do Elastic Beanstalk. Se você não tiver usado o Elastic Beanstalk antes, tente o tutorial de conceitos básicos para saber os princípios.

Tópicos

- Configurar seu ambiente de desenvolvimento Ruby para o Elastic Beanstalk
- Usar a plataforma Ruby do Elastic Beanstalk
- Implantar uma aplicação rails no Elastic Beanstalk
- Implantar uma aplicação sinatra no Elastic Beanstalk
- Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Ruby Elastic Beanstalk

Configurar seu ambiente de desenvolvimento Ruby para o Elastic Beanstalk

Este capítulo fornece instruções para configurar um ambiente de desenvolvimento Ruby para testar sua aplicação localmente antes de implantá-la no AWS Elastic Beanstalk. Também faz referência a sites com instruções de instalação para ferramentas úteis.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

Seções

- Instalar o Ruby
- Instalando o AWS SDK para Ruby
- Instalar um editor de texto ou IDE

Instalar o Ruby

Instale o GCC se você não tiver um compilador C. No Ubuntu, use `apt`.

```
~$ sudo apt install gcc
```

No Amazon Linux, use `yum`.

```
~$ sudo yum install gcc
```

Instale o RVM para gerenciar as instalações da linguagem Ruby em sua máquina. Use os comandos no rvm.io para obter as chaves do projeto e executar o script de instalação.

```
~$ gpg2 --recv-keys key1 key2
~$ curl -sSL https://get.rvm.io | bash -s stable
```

Esse script instala o RVM em uma pasta chamada `.rvm` no diretório do usuário e modifica seu perfil de shell para carregar um script de configuração sempre que você abrir um novo terminal. Carregue o script manualmente para começar.

```
~$ source ~/.rvm/scripts/rvm
```

Use `rvm get head` para obter a versão mais recente.

```
~$ rvm get head
```

Visualize as versões disponíveis do Ruby.

```
~$ rvm list known
```

Confira o Ruby no documento Plataformas da AWS Elastic Beanstalk para encontrar a versão mais recente do Ruby disponível em uma plataforma Elastic Beanstalk. Instale essa versão.

```
~$ rvm install 3.2
```

Teste a instalação do Ruby.

```
~$ ruby --version
```

Instalando o AWS SDK para Ruby

Se você precisar gerenciar AWS recursos de dentro do seu aplicativo, instale AWS SDK para Ruby o. Por exemplo, com o SDK for Ruby, é possível usar o Amazon DynamoDB (DynamoDB) para armazenar informações de usuário e sessão sem criar um banco de dados relacional.

Instale o SDK for Ruby e suas dependências com o comando `gem`.

```
$ gem install aws-sdk
```

Acesse a página inicial do AWS SDK para Ruby para obter mais informações e instruções de instalação.

Instalar um editor de texto ou IDE

Ambientes de desenvolvimento integrados (IDEs) fornecem uma ampla variedade de recursos que facilitam o desenvolvimento de aplicativos. Se você ainda não usou um IDE para desenvolvimento em Ruby, experimente o Aptana RubyMine e veja qual funciona melhor para você.

- Instalar o Aptana

- RubyMine

> **Note**
>
> Um IDE pode adicionar arquivos à pasta do seu projeto, que talvez você não queira confirmar no controle de código-fonte. Para evitar a confirmação desses arquivos no controle de código-fonte, use `.gitignore` ou a ferramenta de controle de código-fonte equivalente.

Se você deseja apenas começar a codificação e não precisa de todos os recursos de um IDE, considere instalar o Sublime Text.

Usar a plataforma Ruby do Elastic Beanstalk

Este tópico descreve como configurar, criar e executar aplicações Ruby no Elastic Beanstalk.

AWS Elastic Beanstalk suporta várias ramificações de plataforma para diferentes versões da linguagem de programação Ruby. Consulte Ruby no documento sobre Plataformas do AWS Elastic Beanstalk para obter uma lista completa.

A aplicação Web Ruby pode ser executada por trás de um servidor proxy NGINX em um servidor de aplicações Puma. Se você usar RubyGems, você pode incluir um `Gemfile`em seu pacote de origem para instalar pacotes durante a implantação.

Configuração do servidor de aplicações

O Elastic Beanstalk instala o servidor de aplicações Puma com base na ramificação da plataforma Ruby que você escolhe ao criar o ambiente. Para obter mais informações sobre os componentes fornecidos com as versões da plataforma Ruby, consulte Plataformas compatíveis no guia Plataformas do AWS Elastic Beanstalk .

Você pode configurar a aplicação com seu próprio servidor Puma. Isso permite usar uma versão do Puma diferente da que é pré-instalada com a ramificação da plataforma Ruby. Você também pode configurar a aplicação para usar um servidor de aplicações diferente, como o Passenger. Para isso, você deve incluir e personalizar um `Gemfile` em sua implantação. Você também precisa configurar um `Procfile` para iniciar o servidor de aplicações. Para obter mais informações, consulte Configurar o processo da aplicação com um Procfile.

Outras opções de configuração

O Elastic Beanstalk fornece opções de configuração que você pode usar para personalizar o software executado nas instâncias do Amazon Elastic Compute Cloud (EC2Amazon) em seu ambiente do Elastic Beanstalk. Você pode configurar variáveis de ambiente necessárias à aplicação, habilitar a alternância de logs para o Amazon S3 e mapear pastas da origem da aplicação que contenham arquivos estáticos para caminhos atendidos pelo servidor de proxy. A plataforma também predefine algumas variáveis de ambiente comuns relacionadas ao Rails e ao Rack para facilitar a descoberta e o uso.

As opções de configuração estão disponíveis no console do Elastic Beanstalk para modificar a configuração de um ambiente em execução. Para evitar a perda da configuração do ambiente quando você terminá-la, você pode usar configurações salvas para salvar as configurações e depois aplicá-las a outro ambiente.

Para salvar as configurações no código-fonte, você pode incluir arquivos de configuração. As definições em arquivos de configuração são aplicadas sempre que você cria um ambiente ou implementa a aplicação. Você também pode usar arquivos de configuração para instalar pacotes, executar scripts e realizar outras operações de personalização da instância em implantações.

As configurações aplicadas no console do Elastic Beanstalk substituem as mesmas configurações em arquivos de configuração, caso elas existam. Isso permite que você tenha configurações padrão em arquivos de configuração e as substitua por configurações específicas de ambiente no console. Para obter mais informações sobre precedência e outros métodos de alteração das configurações, consulte Opções de configuração.

Para obter detalhes sobre as várias maneiras de estender uma plataforma do Elastic Beanstalk baseada em Linux, consulte the section called “Estender plataformas Linux”.

Configurar o ambiente Ruby

É possível usar o console do Elastic Beanstalk para habilitar a alternância de logs para o Amazon S3 e configurar variáveis que sua aplicação pode ler no ambiente.

Para acessar as definições de configuração de software para seu ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).

Opções de log

A seção Log options (Opções de log) tem duas configurações:

- Instance profile (Perfil da instância): especifica o perfil da instância que tem permissão para acessar o bucket do Amazon S3 associado à sua aplicação.
- Habilitar a rotação de arquivos de log para o Amazon S3 — Especifica se os arquivos de log das EC2 instâncias da Amazon do seu aplicativo são copiados para o bucket do Amazon S3 associado ao seu aplicativo.

Arquivos estáticos

Para melhorar a performance, você pode usar a seção Static files (Arquivos estáticos) para configurar o servidor proxy para servir arquivos estáticos (por exemplo, HTML ou imagens) de um conjunto de diretórios dentro da aplicação Web. Defina, para cada diretório, o caminho virtual para mapeamento de diretórios. Quando o servidor de proxy recebe uma solicitação para um arquivo no caminho especificado, ele atende o arquivo diretamente em vez de rotear solicitação para seu aplicativo.

Para ver detalhes sobre como configurar arquivos estáticos usando arquivos de configuração ou o console do Elastic Beanstalk, consulte the section called “Arquivos estáticos”.

Por padrão, o servidor de proxy em um ambiente Ruby é configurado para atender arquivos estáticos da seguinte maneira:

- Arquivos na pasta `public` são atendidos a partir do caminho `/public` e a raiz do domínio (caminho `/`).
- Arquivos da subpasta `public/assets` são atendidos a partir do caminho `/assets`.

Os exemplos a seguir ilustram como funciona a configuração padrão:

- Por exemplo, se a origem da aplicação-fonte contiver um arquivo chamado `logo.png` em uma pasta chamada `public`, o servidor de proxy o servirá a usuários a partir de *`subdomain`*`.elasticbeanstalk.com/public/logo.png` e *`subdomain`*`.elasticbeanstalk.com/logo.png`.
- Se a origem da aplicação-fonte contiver um arquivo nomeado `logo.png` em uma pasta nomeada `assets`, dentro da pasta `public`, o servidor de proxy o servirá a partir de *`subdomain`*`.elasticbeanstalk.com/assets/logo.png`.

Você pode configurar mais mapeamentos para arquivos estáticos. Para obter mais informações, consulte Namespaces de configuração do Ruby mais adiante neste tópico.

> **Note**
>
> Para versões da plataforma anteriores à AL2 versão 3.3.7 do Ruby 2.7, a configuração padrão do servidor proxy nginx do Elastic Beanstalk não oferece suporte ao serviço de arquivos estáticos do domínio root (). `subdomain.elasticbeanstalk.com/` Esta versão da plataforma foi lançada em 21 de outubro de 2021. Para obter mais informações, consulte Novas versões da plataforma - Ruby nas Notas de Release do AWS Elastic Beanstalk .

Propriedades de ambiente

A seção Propriedades do ambiente permite que você especifique as configurações do ambiente nas EC2 instâncias da Amazon que estão executando seu aplicativo. As propriedades de ambiente são passadas para o aplicativo como pares de chave-valor.

A plataforma Ruby define as seguintes propriedades de configuração do ambiente:

- BUNDLE_WITHOUT: uma lista separada por dois pontos de grupos a ignorar ao instalar dependências de um Gemfile.
- BUNDLER_DEPLOYMENT_MODE: defina como `true` (padrão) para instalar as dependências no modo de implantação usando o Bundler. Defina como `false` para executar `bundle install` no modo de desenvolvimento.

  > **Note**
  >
  > Essa propriedade de ambiente não é definida nas ramificações da plataforma Ruby da AMI do Amazon Linux (anteriores ao Amazon Linux 2).

- RAILS_SKIP_ASSET_COMPILATION: definido como `true` para ignorar a execução de `rake assets:precompile` durante a implantação.
- RAILS_SKIP_MIGRATIONS: definido como `true` para ignorar a execução de `rake db:migrate` durante a implantação.
- RACK_ENV: especifique o estágio do ambiente para o Rack. Por exemplo, `development`, `production` ou `test`.

No ambiente Ruby executado no Elastic Beanstalk, as variáveis de ambiente podem ser acessadas usando o objeto `ENV`. Por exemplo, você pode ler uma propriedade denominada `API_ENDPOINT` em uma variável com o seguinte código:

```
endpoint = ENV['API_ENDPOINT']
```

Consulte Variáveis de ambiente e outras configurações de software para obter mais informações.

Namespaces de configuração do Ruby

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

É possível usar o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` para configurar o proxy de ambiente para servir arquivos estáticos. Defina mapeamentos de caminhos virtuais para diretórios de aplicativos.

A plataforma Ruby não define nenhum namespace específico da plataforma. Em vez disso, ela define as propriedades do ambiente para opções de Rails e Rack comuns.

O arquivo de configuração a seguir especifica uma opção de arquivos estáticos que mapeia um diretório chamado `staticimages` para o caminho `/images`, define cada uma das propriedades de ambiente definidas pela plataforma e define uma propriedade de ambiente adicional chamada `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**
>
> A propriedade de ambiente `BUNDLER_DEPLOYMENT_MODE` e o namespace `aws:elasticbeanstalk:environment:proxy:staticfiles` não são definidos nas ramificações da plataforma Ruby de AMI do Amazon Linux (anterior ao Amazon Linux 2).

O Elastic Beanstalk oferece muitas opções de configuração para personalizar o ambiente. Além de arquivos de configuração, você também pode definir opções de configuração usando o console, configurações salvas, a EB CLI ou a AWS CLI. Consulte Opções de configuração para obter mais informações.

Instalar pacotes com um Gemfile no Elastic Beanstalk

Para usar RubyGems para instalar pacotes que seu aplicativo exige, inclua um `Gemfile` arquivo na raiz da fonte do seu projeto.

Example Gemfile

```
source "https://rubygems.org"
gem 'sinatra'
gem 'json'
gem 'rack-parser'
```

Quando um arquivo `Gemfile` está presente, o Elastic Beanstalk executa `bundle install` para instalar dependências. Para mais informações, consulte as páginas Gemfiles e Bundle (Pacote) no site Bundler.io.

> **Note**
>
> Você pode usar uma versão diferente do Puma além da versão padrão que é pré-instalada com a plataforma Ruby. Para isso, inclua uma entrada em um `Gemfile` especificando a versão. Você também pode especificar um servidor de aplicações diferente, como o Passenger, usando um `Gemfile` personalizado.
> Em ambos desses casos, você precisa configurar um `Procfile` para iniciar o servidor de aplicações.
> Para obter mais informações, consulte Configurar o processo da aplicação com um Procfile.

Configurar o processo da aplicação com um Procfile no Elastic Beanstalk.

Para especificar o comando que inicia a aplicação em Ruby, inclua um arquivo chamado `Procfile` na raiz do pacote de origem.

> **Note**
>
> O Elastic Beanstalk não é compatível com esse recurso nas ramificações da plataforma Ruby da AMI do Amazon Linux (anteriores ao Amazon Linux 2). As ramificações da plataforma com os nomes que contêm Puma ou Passenger, independentemente de suas versões Ruby, precedem o Amazon Linux 2 e não são compatíveis com o recurso `Procfile`.

Para obter detalhes sobre como gravar e usar um `Procfile`, consulte Buildfile e Procfile.

Quando você não fornece um`Procfile`, o Elastic Beanstalk gera um padrão. `Procfile` Se você `Gemfile` incluir o Puma, o Elastic Beanstalk presume que você deseja usar a versão fornecida do Puma e gera o seguinte padrão. `Procfile`

```
web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

Se você `Gemfile` não incluir o Puma, o Elastic Beanstalk presume que você está usando o servidor de aplicativos Puma pré-instalado e gera o seguinte padrão. `Procfile` Nas ramificações da plataforma Amazon Linux 2 Ruby, o Elastic Beanstalk sempre gera o `Procfile` seguinte padrão se você não fornecer um. `Procfile`

```
web: puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
```

> **Note**
>
> Em 10 de outubro de 2024, as últimas ramificações da plataforma Ruby Amazon Linux 2 foram retiradas. Todas as ramificações da plataforma Ruby atualmente suportadas são baseadas no Amazon Linux 2023. Para obter informações sobre migração, consulteMigração do Amazon Linux 2 para o Amazon Linux 2023.

Se você quiser usar o servidor de aplicativos Passenger, use os seguintes exemplos de arquivos para configurar seu ambiente Ruby a ser instalado e usar o Passenger.

1. Use este exemplo de arquivo para instalar o Passenger.

   Example Gemfile

   ```
   source 'https://rubygems.org'
   gem 'passenger'
   ```

2. Use este arquivo demonstrativo para instruir o Elastic Beanstalk a iniciar o Passenger.

   Example Procfile

   ```
   web: bundle exec passenger start /var/app/current --socket /var/run/puma/my_app.sock
   ```

> **Note**
>
> Não é necessário alterar nada na configuração do servidor de proxy nginx para usar o Passenger. Para usar outros servidores de aplicativos, talvez seja necessário personalizar a configuração do nginx para encaminhar solicitações adequadamente para seu aplicativo.

Implantar uma aplicação rails no Elastic Beanstalk

Rails é uma estrutura de código aberto model-view-controller (MVC) para Ruby. Este tutorial mostra o processo de geração de um aplicativo Rails e sua implantação em um AWS Elastic Beanstalk ambiente.

Seções

- Pré-requisitos
- Conhecimento básico do Elastic Beanstalk
- Iniciar um ambiente do Elastic Beanstalk
- Instalar o rails e gerar um site
- Definir as configurações do rails
- Implantar o aplicativo
- Limpeza
- Próximas etapas

Pré-requisitos

Conhecimento básico do Elastic Beanstalk

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Linha de comando

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

Dependências do rails

O framework Rails 6.1.4.1 tem as dependências a seguir. Certifique-se de que você tem todas elas instaladas.

- Ruby 2.5.0 ou mais recente: para obter instruções de instalação, consulte Configurar seu ambiente de desenvolvimento Ruby para o Elastic Beanstalk.

  Neste tutorial, usamos Ruby 3.0.2 e a versão correspondente da plataforma do Elastic Beanstalk.
- Node.js: para obter instruções de instalação, consulte Instalar o Node.js como o gerenciador de pacotes.
- Yarn: para obter instruções de instalação, consulte Installation no site do Yarn.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma Ruby e aceite as configurações padrão e o código de exemplo.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.

- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Instalar o rails e gerar um site

Instale o Rails e suas dependências com o 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
...
```

Teste a instalação do Rails.

```
~$ rails --version
Rails 6.1.4.1
```

Use `rails new` com o nome do aplicativo para criar um novo projeto Rails.

```
~$ rails new ~/eb-rails
```

O Rails cria um diretório com o nome especificado, gera todos os arquivos necessários para executar um projeto de exemplo localmente e, em seguida, executa o bundler para instalar todas as dependências (Gems) definidas no Gemfile do projeto.

> **Note**
>
> Este processo instala a versão mais recente do Puma para o projeto. Talvez essa versão seja diferente da versão que o Elastic Beanstalk fornece na versão da plataforma Ruby do seu ambiente. Para ver as versões do Puma fornecidas pelo Elastic Beanstalk, consulte Histórico de plataforma Ruby no Guia de plataformas do AWS Elastic Beanstalk . Para mais informações sobre a versão mais recente do Puma, consulte o site Puma.io. Se houver uma incompatibilidade entre as duas versões do Puma, adote uma das seguintes opções:
>
> - Use a versão do Puma instalada pelo comando `rails new` anterior. Nesse caso, é necessário adicionar um `Procfile` para que a plataforma use a versão do servidor Puma que você fornece. Para obter mais informações, consulte Configurar o processo da aplicação com um Procfile no Elastic Beanstalk..
> - Atualize a versão do Puma para ser consistente com a versão pré-instalada na versão da plataforma Ruby do seu ambiente. Para fazer isso, modifique a versão do Puma no Gemfile localizado na raiz do diretório de origem do projeto. Em seguida, execute `bundle update`. Para mais informações, consulte a página bundle update (atualização do pacote) no site Bundler.io.

Teste a instalação do Rails executando o projeto padrão localmente.

```
~$ 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
...
```

Abra `http://localhost:3000` em um navegador da web para ver o projeto padrão em ação.



Essa página só está visível em modo de desenvolvimento. Adicione algum conteúdo à página inicial da aplicação para oferecer suporte à implantação de produção no Elastic Beanstalk. Use `rails generate` para criar um controlador, uma rota e uma visualização de sua página de boas-vindas.

```
~/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.
```

Isso fornece tudo o que você precisa para acessar a página na `/welcome_page/welcome`. No entanto, antes de publicar as alterações, altere o conteúdo na visualização e adicione uma rota para que essa página apareça no nível superior do site.

Use um editor de texto para editar o conteúdo em `app/views/welcome_page/welcome.html.erb`. Para este exemplo, você usará `cat` para simplesmente substituir o conteúdo do arquivo existente.

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>
```

Por fim, adicione a seguinte rota a `config/routes.rb`:

Example config/routes.rb

```
Rails.application.routes.draw do
  get 'welcome_page/welcome'
```

```
  root 'welcome_page#welcome'
```

Isso instrui o Rails a rotear as solicitações à raiz do site para o método de boas-vindas do controlador da página de boas-vindas, que renderiza o conteúdo na visualização de boas-vindas (`welcome.html.erb`).

Para que o Elastic Beanstalk implante com sucesso a aplicação na plataforma Ruby, precisamos atualizar o `Gemfile.lock`. Algumas dependências do `Gemfile.lock` talvez sejam específicas da plataforma. Portanto, precisamos adicionar **`platform ruby`** a `Gemfile.lock` para que todas as dependências necessárias sejam instaladas com a implantação.

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
```

Definir as configurações do rails

Use o console do Elastic Beanstalk para configurar o Rails com propriedades do ambiente. Defina a propriedade do ambiente `SECRET_KEY_BASE` para uma string de até 256 caracteres alfanuméricos.

O Rails usa essa propriedade para criar chaves. Portanto, você deve mantê-la em segredo e não armazená-la no controle de origem em texto sem formatação. Em vez disso, você a fornece ao código do Rails no ambiente por meio de uma propriedade de ambiente.

Para configurar variáveis de ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até Variáveis de ambiente de tempo de execução.
6. Selecione Adicionar variável de ambiente.
7. Em Fonte, selecione Texto sem formatação.

> **Note**
>
> Os valores do Secrets Manager e do SSM Parameter Store no menu suspenso servem para configurar variáveis de ambiente como segredos para armazenar dados confidenciais, como credenciais e chaves de API. Para obter mais informações, consulte Usando o Elastic AWS Secrets ManagerAWS Systems Manager Beanstalk com um Parameter Store.

8. Insira o nome da variável de ambiente e os pares de valores da variável de ambiente.
9. Se precisar adicionar mais variáveis, repita a Etapa 6 até a Etapa 8.
10. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Agora você está pronto para implantar o site no ambiente.

Implantar o aplicativo

Crie um pacote de origem com os arquivos criados pelo Rails. O comando a seguir cria um pacote de origem chamado `rails-default.zip`.

```
~/eb-rails$ zip ../rails-default.zip -r * .[^.]*
```

Faça upload do pacote de origem no Elastic Beanstalk para implantar o Rails no ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Próximas etapas

Para obter mais informações sobre o Rails, visite rubyonrails.org.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Implantar uma aplicação sinatra no Elastic Beanstalk

Este passo a passo mostra como implantar uma aplicação web Sinatra simples no AWS Elastic Beanstalk.

Pré-requisitos

Este tutorial pressupõe que você tenha conhecimento das operações básicas e do console do Elastic Beanstalk. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk.

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

O Sinatra 2.1.0 requer o Ruby 2.3.0 ou mais recente. Neste tutorial, usamos Ruby 3.0.2 e a versão correspondente da plataforma do Elastic Beanstalk. Instale o Ruby seguindo as instruções em Configurar seu ambiente de desenvolvimento Ruby para o Elastic Beanstalk.

Iniciar um ambiente do Elastic Beanstalk

Use o console do Elastic Beanstalk para criar um ambiente do Elastic Beanstalk. Escolha a configuração da Plataforma Ruby e aceite as configurações padrão e o código de exemplo.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk usando este link pré-configurado: console.aws.amazon.com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
2. Em Platform (Plataforma), selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo seu aplicativo.
3. Para Application code, escolha Sample application.
4. Selecione Review and launch.
5. Examine as opções disponíveis. Escolha a opção disponível que deseja usar e, quando estiver pronto, escolha Create app (Criar aplicativo).

A criação de ambiente leva cerca de 5 minutos e cria os seguintes recursos:

- EC2 instância — Uma máquina virtual Amazon Elastic Compute Cloud (Amazon EC2) configurada para executar aplicativos web na plataforma que você escolher.

Cada plataforma executa um conjunto específico de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou o NGINX como um proxy reverso que fica na frente da aplicação Web, encaminha solicitações para ela, atende ativos estáticos e gera logs de acesso e de erro.

- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Balanceador de carga: um balanceador de carga do Elastic Load Balancing configurado para distribuir solicitações para as instâncias que executam a aplicação. Um load balancer também elimina a necessidade de expor suas instâncias diretamente à Internet.
- Grupo de segurança do balanceador de carga — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP da Internet acesse o load balancer. Por padrão, o tráfego não é permitido em outras portas.
- Grupo de Auto Scaling: grupo de Auto Scaling configurado para substituir uma instância se ela for encerrada ou ficar indisponível.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e que são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *`subdomain`*.*`region`*.elasticbeanstalk.com.

> **Segurança de domínios**
>
> Para aumentar a segurança de suas aplicações do Elastic Beanstalk, o domínio elasticbeanstalk.com é registrado na Lista Pública de Sufixos (PSL).
> Se precisar definir cookies confidenciais no nome de domínio padrão para suas aplicações Elastic Beanstalk, recomendamos usar cookies com um prefixo `__Host-` para maior segurança. Essa prática defende seu domínio contra tentativas de falsificação de

solicitação entre sites (CSRF). Para obter mais informações, consulte a página Set-Cookie na Mozilla Developer Network.

Todos esses recursos são gerenciados pelo Elastic Beanstalk. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

> **Note**
>
> O bucket do Amazon S3 que o Elastic Beanstalk cria é compartilhado entre ambientes e não é excluído durante o encerramento do ambiente. Para obter mais informações, consulte Usar o Elastic Beanstalk com o Amazon S3.

Escrever um site sinatra básico

Como criar e implantar um aplicativo sinatra

1. Crie um arquivo de configuração chamado config.ru com o conteúdo a seguir.

   Example config.ru

   ```
   require './helloworld'
   run Sinatra::Application
   ```

2. Crie um arquivo de código Ruby chamado helloworld.rb com o conteúdo a seguir.

   Example helloworld.rb

   ```
   require 'sinatra'
   get '/' do
     "Hello World!"
   end
   ```

3. Crie um Gemfile com o conteúdo a seguir.

   Example Gemfile

   ```
   source 'https://rubygems.org'
   gem 'sinatra'
   ```

```
gem 'puma'
```

4. Execute a instalação do pacote para gerar o `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. Para que o Elastic Beanstalk implante com sucesso a aplicação na plataforma Ruby, precisamos atualizar o `Gemfile.lock`. Algumas dependências do `Gemfile.lock` talvez sejam específicas da plataforma. Portanto, precisamos adicionar **`platform ruby`** a `Gemfile.lock` para que todas as dependências necessárias sejam instaladas com a implantação.

   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. Crie um arquivo Procfile com o seguinte conteúdo:

   Example Procfile

   ```
   web: bundle exec puma -C /opt/elasticbeanstalk/config/private/pumaconf.rb
   ```

Implantar o aplicativo

Crie um pacote de origem com os arquivos de origem. O comando a seguir cria um pacote de origem chamado `sinatra-default.zip`.

```
~/eb-sinatra$ zip ../sinatra-default.zip -r * .[^.]*
```

Faça upload do pacote de origem no Elastic Beanstalk para implantar o Sinatra no ambiente.

Para implantar um pacote de origem

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Na página de visão geral do ambiente, escolha Upload and deploy (Fazer upload e implantar).
4. Use a caixa de diálogo na tela para carregar o pacote de origem.
5. Escolha Deploy (Implantar).
6. Quando a implantação for concluída, é possível escolher o URL do site para abri-lo em uma nova guia.

Limpeza

Depois de terminar de trabalhar com o código de demonstração, você pode encerrar seu ambiente. O Elastic Beanstalk exclui todos os recursos AWS relacionados, como instâncias da EC2 Amazon, instâncias de banco de dados, balanceadores de carga, grupos de segurança e alarmes.

A remoção de recursos não exclui o aplicativo Elastic Beanstalk, então você pode criar novos ambientes para seu aplicativo a qualquer momento.

Para encerrar seu ambiente Elastic Beanstalk do console

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Próximas etapas

Para mais informações sobre o Sinatra, acesse sinatrarb.com.

À medida que você desenvolva a aplicação, provavelmente vai precisar de uma maneira de gerenciar ambientes e implantá-la sem criar manualmente um arquivo .zip e carregá-la no console do Elastic Beanstalk. A interface de linha de comando do Elastic Beanstalk (EB CLI easy-to-use) fornece comandos para criar, configurar e implantar aplicativos nos ambientes do Elastic Beanstalk a partir da linha de comando.

Por fim, se você planeja usar seu aplicativo em um ambiente de produção, configure um nome de domínio personalizado para seu ambiente e habilite HTTPS para conexões seguras.

Adicionar uma instância de banco de dados do Amazon RDS ao seu ambiente Ruby Elastic Beanstalk

Este tópico fornece instruções para criar o Amazon RDS usando o console do Elastic Beanstalk. É possível usar uma instância de banco de dados do Amazon Relational Database Service (Amazon RDS) para armazenar dados coletados e modificados pela aplicação. O banco de dados pode ser acoplado ao seu ambiente e gerenciado pelo Elastic Beanstalk ou pode ser criado como desacoplado e gerenciado externamente por outro serviço. Nestas instruções, o banco de dados é acoplado ao ambiente e gerenciado pelo Elastic Beanstalk. Para obter mais informações sobre como integrar um Amazon RDS com o Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Seções

- Adicionar uma instância de banco de dados ao seu ambiente
- Download de adaptador
- Conectar-se a um banco de dados

Adicionar uma instância de banco de dados ao seu ambiente

Para adicionar uma instância de banco de dados ao ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Database (Banco de dados), escolha Edit (Editar).
5. Escolha um mecanismo de banco de dados e insira um nome de usuário e senha.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A adição de uma instância de banco de dados leva cerca de 10 minutos. Quando a atualização do ambiente for concluída, o nome de host da instância de banco de dados e outras informações de conexão estarão disponíveis para o seu aplicativo por meio das seguintes propriedades de ambiente:

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).
`RDS_USERNAME`	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
`RDS_PASSWORD`	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.

Para obter mais informações sobre como configurar uma instância de banco de dados acoplada a um ambiente Elastic Beanstalk, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Download de adaptador

Adicione o adaptador de banco de dados ao arquivo de gem de seu projeto.

Example Gemfile: Rails com o MySQL

```
source 'https://rubygems.org'
gem 'puma'
```

```
gem 'rails', '~> 6.1.4', '>= 6.1.4.1'
gem 'mysql2'
```

Gems de adaptadores comuns para Ruby

- MySQL – mysql2
- PostgreSQL – pg
- Oracle: activerecord-oracle_enhanced-adapter
- SQL Server – activerecord-sqlserver-adapter

Conectar-se a um banco de dados

O Elastic Beanstalk fornece informações de conexão referentes às instâncias de banco de dados associadas nas propriedades de ambiente. Use `ENV['VARIABLE']` para ler as propriedades e configurar uma conexão de banco de dados.

Example config/database.yml: configuração do banco de dados de 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'] %>
```

Implantação com contêineres Docker no Elastic Beanstalk

Este capítulo explica como usar o Elastic Beanstalk para implantar aplicações Web a partir de contêineres do Docker. Os contêineres do Docker são independentes e incluem todas as informações de configuração e software necessários à execução da aplicação Web. Com os contêineres do Docker, você pode definir seu próprio ambiente de runtime. Você também pode escolher sua própria linguagem de programação e dependências de aplicações, como gerenciadores de pacotes ou ferramentas, que normalmente não são compatíveis por outras plataformas Elastic Beanstalk.

Siga as etapas em QuickStart para Docker para criar uma aplicação do Docker "Hello World" e implantá-la em um ambiente Elastic Beanstalk usando a EB CLI.

Tópicos

- Ramificações da plataforma do Docker do Elastic Beanstalk
- Usar a ramificação da plataforma do Docker do Elastic Beanstalk
- Usando a ramificação da plataforma Docker gerenciada pelo ECS no Elastic Beanstalk
- Usar imagens de um repositório privado no Elastic Beanstalk
- Configuração de ambientes do Docker do Elastic Beanstalk
- Plataformas herdadas

Ramificações da plataforma do Docker do Elastic Beanstalk

A plataforma Docker do Elastic Beanstalk oferece suporte às seguintes ramificações da plataforma:

Docker executando Amazon Linux 2 e Docker executando 023 AL2

O Elastic Beanstalk implanta contêineres e código-fonte do Docker nas instâncias e as gerencia. EC2 Essas ramificações da plataforma oferecem suporte a vários contêineres. Você pode usar a ferramenta Docker Compose para simplificar a configuração, o teste e a implantação do seu aplicativo. Para obter mais informações sobre essa plataforma, consulte the section called “Ramificação da plataforma do Docker”.

ECS em execução no Amazon Linux 2 e ECS em execução em 023 AL2

Fornecemos essa filial para clientes que precisam de um caminho de migração para AL2 023/ AL2 da filial aposentada da plataforma Docker de vários contêineres em execução (Amazon Linux

AMI). As ramificações de plataforma mais recentes oferecem suporte a todos os recursos da ramificação da plataforma descontinuada. Não é necessária nenhuma alteração no código-fonte. Para obter mais informações, consulte Migrando seu aplicativo Elastic Beanstalk do Docker de vários contêineres gerenciado pelo ECS para o ECS no Amazon Linux 2023 AL1 . Se você não tiver um ambiente do Elastic Beanstalk em execução em uma ramificação de plataforma baseada em ECS, recomendamos que você use a ramificação da plataforma, Docker Running on 64bit 023. AL2 Isso oferece uma abordagem mais simples e requer menos recursos.

Para obter uma lista das versões dos componentes de software associadas a cada um dessas ramificações da plataforma, consulte Docker no documento Plataformas do AWS Elastic Beanstalk .

Filiais de plataformas descontinuadas em execução no Amazon Linux AMI (AL1)

Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Expanda cada seção a seguir para ler mais sobre cada ramificação da plataforma descontinuada e seu caminho de migração para a ramificação da plataforma mais recente em execução no Amazon Linux 2 ou Amazon Linux 2023 (recomendado).

Docker (AMI do Amazon Linux)

Esta ramificação da plataforma pode implantar uma imagem do Docker, descrita em um Dockerfile ou em uma definição do `Dockerrun.aws.json` v1. Essa ramificação da plataforma executa apenas um contêiner para cada instância. Suas sucessivas ramificações de plataforma, o Docker em execução em AL2 023 de 64 bits e o Docker em execução no Amazon Linux 2 de 64 bits, oferecem suporte a vários contêineres Docker por instância.

Recomendamos que você crie seus ambientes com a ramificação de plataforma mais recente e compatível, Docker, executada em 64 bits AL2 023. Você pode então migrar seu aplicativo para o ambiente recém-criado. Para obter mais informações sobre esses ambientes, consulte the section called “Ramificação da plataforma do Docker”. Para obter mais informações sobre a migração, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

Docker de vários contêineres (Amazon Linux AMI)

Essa ramificação da plataforma usa o Amazon ECS para coordenar uma implantação de vários contêineres do Docker em um cluster do Amazon ECS em um ambiente do Elastic Beanstalk. Se

você estiver usando esta ramificação da plataforma descontinuada, recomendamos migrar para a ramificação da plataforma ECS em execução no Amazon Linux 2023 mais recente. A ramificação mais recente da plataforma é compatível com todos os recursos da ramificação da plataforma que foi retirada. Não é necessária nenhuma alteração no código-fonte. Para obter mais informações, consulte Migrando seu aplicativo Elastic Beanstalk do Docker de vários contêineres gerenciado pelo ECS para o ECS no Amazon Linux 2023 AL1 .

Contêineres do Docker pré-configurados

Além das plataformas Docker mencionadas anteriormente, há também a ramificação da GlassFish plataforma Docker pré-configurada que é executada no sistema operacional Amazon Linux AMI (). AL1

Essa ramificação da plataforma foi substituída pelas ramificações da plataforma Docker em execução em AL2 023 de 64 bits e Docker em execução no Amazon Linux 2 de 64 bits. Para obter mais informações, consulte Implantação de um GlassFish aplicativo na plataforma Docker.

Usar a ramificação da plataforma do Docker do Elastic Beanstalk

Esta seção descreve como preparar sua imagem do Docker para lançamento com qualquer uma das ramificações da plataforma Elastic Beanstalk em execução no Docker ou 023. AL2 AL2

Siga as etapas em QuickStart para Docker para criar uma aplicação do Docker "Hello World" e implantá-la em um ambiente Elastic Beanstalk usando a EB CLI.

Tópicos

- QuickStart: Implante um aplicativo Docker no Elastic Beanstalk
- Preparar a imagem do Docker para implantação no Elastic Beanstalk

QuickStart: Implante um aplicativo Docker no Elastic Beanstalk

Este QuickStart tutorial mostra o processo de criação de um aplicativo Docker e sua implantação em um AWS Elastic Beanstalk ambiente.

> **Note**
>
> Os exemplos de tutoriais são destinados à demonstração. Não use o aplicativo para tráfego de produção.

Seções

- Sua AWS conta
- Pré-requisitos
- Etapa 1: criar uma aplicação do Docker e um contêiner
- Etapa 2: executar a aplicação localmente
- Etapa 3: implantar a aplicação do Docker com a EB CLI
- Etapa 4: executar a aplicação no Elastic Beanstalk
- Etapa 5: limpar
- AWS recursos para seu aplicativo
- Próximas etapas
- Implantar com o console do Elastic Beanstalk

Sua AWS conta

Se você ainda não é AWS cliente, precisa criar uma AWS conta. A inscrição permite que você acesse o Elastic Beanstalk AWS e outros serviços de que você precisa.

Se você já tem uma AWS conta, você pode passar paraPré-requisitos.

Crie uma AWS conta

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Pré-requisitos

Para seguir os procedimentos neste manual, você precisa de um terminal de linha de comando ou de um shell para executar os comandos. Nas listagens, os comandos são mostrados precedidos por um símbolo de prompt ($) e pelo nome do diretório atual, quando apropriado.

```
~/eb-project$ this is a command
this is output
```

No Linux e no macOS, você pode usar seu gerenciador de pacotes e de shell preferido. No Windows, você pode instalar o subsistema Windows para Linux para obter uma versão do Ubuntu e Bash integrada ao Windows.

EB CLI

Este tutorial também usa a Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para obter detalhes sobre a instalação e configuração da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado) e Configurar a EB CLI.

Docker

Para seguir este tutorial, você precisará de uma instalação local funcional do Docker. Para obter mais informações, consulte Obter o Docker no site de documentação do Docker.

Verifique se o daemon do Docker está ativo e em execução executando o seguinte comando.

```
~$ docker info
```

Etapa 1: criar uma aplicação do Docker e um contêiner

Para este exemplo, criamos uma imagem do Docker da aplicação Flask de exemplo que também é referenciada em Implantar uma aplicação Flask no Elastic Beanstalk.

A aplicação consiste em dois arquivos:

- `app.py`: o arquivo Python que contém o código que será executado no contêiner.
- `Dockerfile`: o Dockerfile para construir sua imagem.

Coloque ambos os arquivos na raiz de um diretório.

```
~/eb-docker-flask/
|-- Dockerfile
|-- app.py
```

Adicione o seguinte conteúdo ao seu `Dockerfile`.

Example **`~/eb-docker-flask/Dockerfile`**

```
FROM python:3.12
COPY . /app
WORKDIR /app
RUN pip install Flask==3.0.2
EXPOSE 5000
CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

Adicione o seguinte conteúdo ao seu arquivo `app.py`.

Example **`~/eb-docker-flask/app.py`**

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
```

```
    return 'Hello Elastic Beanstalk! This is a Docker application'
```

Crie seu contêiner do Docker, marcando a imagem com `eb-docker-flask`.

```
~/eb-docker-flask$ docker build -t eb-docker-flask
```

Etapa 2: executar a aplicação localmente

Use o comando docker build para construir sua imagem de contêiner localmente, marcando a imagem com `eb-docker-flask`. O período (`.`) no final do comando especifica que o caminho é um diretório local.

```
~/eb-docker-flask$ docker run -dp 127.0.0.1:5000:5000 eb-docker-flask .
```

Execute o contêiner com o comando docker run. O comando imprimirá o ID do contêiner em execução. A opção -d executa o docker no modo em segundo plano. A opção -p expõe a aplicação na porta 5000. O Elastic Beanstalk envia tráfego para a porta 5000 na plataforma do Docker por padrão.

```
~/eb-docker-flask$ docker run -dp 127.0.0.1:5000:5000 eb-docker-flask container-id
```

Navegue para `http://127.0.0.1:5000/` noseu navegador. Você verá o texto "Hello Elastic Beanstalk! Este é uma aplicação do Docker".

Execute o comando docker kill para encerrar o contêiner.

```
~/eb-docker-flask$ docker kill container-id
```

Etapa 3: implantar a aplicação do Docker com a EB CLI

Execute os seguintes comandos para criar um ambiente Elastic Beanstalk para essa aplicação.

Para criar um ambiente e implantar a aplicação do Docker

1. Inicialize o repositório da EB CLI com o comando eb init:

   ```
   ~/eb-docker-flask$ eb init -p docker docker-tutorial us-east-2
   ```

```
Application docker-tutorial has been created.
```

Este comando cria uma aplicação `docker-tutorial` e configura seu repositório local para criar ambientes com a versão mais recente da plataforma do Docker.

2. (Opcional) Execute eb init novamente para configurar um par de chaves padrão para que você possa usar o SSH para se conectar à EC2 instância que executa seu aplicativo.

```
~/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 ]
```

Selecione um par de chaves se você já tiver um, ou siga os as instruções na tela para criar um. Se nenhum prompt aparecer ou se for necessário alterar as configurações posteriormente, execute eb init -i.

3. Crie um ambiente e implante o aplicativo nele com eb create. O Elastic Beanstalk cria automaticamente um arquivo zip para a aplicação e o inicia na porta 5000.

```
~/eb-docker-flask$ eb create docker-tutorial
```

O Elastic Beanstalk leva cerca de cinco minutos para criar seu ambiente.

Etapa 4: executar a aplicação no Elastic Beanstalk

Quando o processo de criação de seu ambiente for concluído, abra seu site com eb open.

```
~/eb-docker-flask$ eb open
```

Parabéns! Você implantou uma aplicação do Docker com o Elastic Beanstalk! Isso abre uma janela do navegador usando o nome de domínio criado para o seu aplicativo.

Etapa 5: limpar

Você pode encerrar o ambiente quando terminar de trabalhar com a aplicação. O Elastic Beanstalk AWS encerra todos os recursos associados ao seu ambiente.

Para encerrar seu ambiente do Elastic Beanstalk com a CLI do EB, execute o seguinte comando.

```
~/eb-docker-flask$ eb terminate
```

AWS recursos para seu aplicativo

Você acabou de criar uma aplicação de instância única. Ele serve como um aplicativo de amostra simples com uma única EC2 instância, portanto, não requer balanceamento de carga ou escalonamento automático. Para aplicativos de instância única, o Elastic Beanstalk cria os seguintes recursos: AWS

- EC2 instância — Uma máquina EC2 virtual da Amazon configurada para executar aplicativos web na plataforma que você escolher.

  Cada plataforma executa um conjunto diferente de software, arquivos de configuração e scripts para oferecer suporte a uma versão de linguagem, framework, contêiner da Web específicos ou a uma combinação deles. A maioria das plataformas usa o Apache ou nginx como um proxy reverso que processa tráfego da web na frente do aplicativo web, encaminha solicitações para ele, atende ativos estáticos e gera logs de acesso e de erro.
- Grupo de segurança da instância — Um grupo EC2 de segurança da Amazon configurado para permitir tráfego de entrada na porta 80. Esse recurso permite que o tráfego HTTP do balanceador de carga chegue à EC2 instância que executa seu aplicativo web. Por padrão, o tráfego não é permitido em outras portas.
- Bucket do Amazon S3: um local de armazenamento do seu código-fonte, logs e outros artefatos criados quando o Elastic Beanstalk é utilizado.
- CloudWatch Alarmes da Amazon — Dois CloudWatch alarmes que monitoram a carga nas instâncias em seu ambiente e são acionados se a carga for muito alta ou muito baixa. Quando um alarme é acionado, o seu grupo de Auto Scaling pode aumentar ou diminuir.
- AWS CloudFormation stack — O Elastic AWS CloudFormation Beanstalk usa para lançar os recursos em seu ambiente e propagar as alterações de configuração. Os recursos são definidos em um modelo que você pode visualizar no console do AWS CloudFormation.
- Nome de domínio — Um nome de domínio que é direcionado para seu aplicativo da web no formulário *subdomain*.*region*.elasticbeanstalk.com.

O Elastic Beanstalk gerencia todos esses recursos. Quando você encerra o ambiente, o Elastic Beanstalk encerra todos os recursos dele.

Próximas etapas

Depois que um ambiente estiver executando um aplicativo, você poderá implantar uma nova versão do aplicativo ou um aplicativo diferente a qualquer momento. A implantação de uma nova versão do aplicativo é muito rápida porque não exige provisionamento ou reinicialização de instâncias. EC2 Também é possível explorar o novo ambiente usando o console do Elastic Beanstalk. Para conhecer as etapas detalhadas, consulte Explore seu ambiente, no capítulo Primeiros passos deste guia.

Depois de implantar uma ou duas aplicações demonstrativas, quando você estiver pronto para começar a desenvolver e executar aplicações Docker localmente, consulte Preparar a imagem do Docker para implantação no Elastic Beanstalk.

Implantar com o console do Elastic Beanstalk

Você também pode usar o console do Elastic Beanstalk para iniciar a aplicação de exemplo. Para conhecer as etapas detalhadas, consulte Criar uma aplicação de exemplo, no capítulo Primeiros passos deste guia.

Preparar a imagem do Docker para implantação no Elastic Beanstalk

Esta seção descreve como preparar sua imagem do Docker para implantação no Elastic Beanstalk com qualquer uma das ramificações do Docker em execução ou da AL2 plataforma 023. AL2 Os arquivos de configuração necessários dependem de as suas imagens serem locais, remotas e se você está usando o Docker Compose.

> **Note**
>
> Para um exemplo de um procedimento que inicia um ambiente do Docker, consulte o tópico QuickStart para Docker.

Tópicos

- Gerenciar suas imagens com o Docker Compose no Elastic Beanstalk
- Gerenciar imagens sem o Docker Compose no Elastic Beanstalk
- Criar imagens personalizadas com um Dockerfile

Gerenciar suas imagens com o Docker Compose no Elastic Beanstalk

É possível optar por usar o Docker Compose para gerenciar vários serviços em um arquivo YAML. Para saber mais sobre o Docker Compose, consulte Por que usar o Compose?, no site do Docker.

- Crie um `docker-compose.yml`. Este arquivo será necessário se você estiver usando o Docker Compose para gerenciar sua aplicação com o Elastic Beanstalk. Se todas as suas implantações forem originadas de imagens em repositórios públicos, nenhum outro arquivo de configuração será necessário. Se as imagens de origem da sua implantação estiverem em um repositório privado, será necessário fazer uma configuração adicional. Para obter mais informações, consulte Uso de imagens de um repositório privado. Para obter mais informações sobre o arquivo `docker-compose.yml`, consulte Referência de arquivo do Compose no site do Docker.
- A `Dockerfile` é opcional. Crie uma se precisar que o Elastic Beanstalk crie e execute uma imagem personalizada local. Para obter mais informações sobre `Dockerfile`, consulte Referência do arquivo do Docker no site do Docker.
- Talvez seja necessário criar um arquivo `.zip`. Se você usar apenas um arquivo `Dockerfile` para implantar a aplicação, não precisará criar um. Se você usar arquivos de configuração adicionais, o arquivo .zip deverá incluir o `Dockerfile`, o arquivo `docker-compose.yml`, os arquivos de aplicação e quaisquer dependências de arquivo de aplicação. O `Dockerfile` e o `docker-compose.yml` devem estar na raiz, ou nível superior, do arquivo .zip. Se você usar a EB CLI para implantar a aplicação, ela criará um arquivo .zip para você.

Para saber mais sobre o Docker Compose e como instalá-lo, consulte Visão geral do Docker Compose e Instalar o Docker Compose no site do Docker.

Gerenciar imagens sem o Docker Compose no Elastic Beanstalk

Se não estiver usando o Docker Compose para gerenciar suas imagens do Docker, será necessário configurar um `Dockerfile`, um arquivo `Dockerrun.aws.json` ou ambos.

- Criar um `Dockerfile` para que o Elastic Beanstalk crie e execute uma imagem personalizada localmente.
- Crie um arquivo `Dockerrun.aws.json v1` para implantar uma imagem do Docker a partir de um repositório hospedado do Elastic Beanstalk.
- Talvez seja necessário criar um arquivo `.zip`. Se você usar apenas um de qualquer arquivo, o `Dockerfile` ou o `Dockerrun.aws.json`, não será necessário criar um arquivo .zip. Se você usar ambos os arquivos, precisará de um arquivo .zip. O arquivo .zip deve incluir `Dockerfile`

e o `Dockerrun.aws.json`, juntamente com o arquivo que contém os arquivos da aplicação e quaisquer dependências do arquivo da aplicação. Se você usar a CLI do EB para implantar a aplicação, ela criará um arquivo `.zip` para você.

Arquivo de configuração do **`Dockerrun.aws.json`** v1

Um arquivo `Dockerrun.aws.json` descreve como implantar uma imagem do Docker remota como uma aplicação do Elastic Beanstalk. Este arquivo JSON é específico do Elastic Beanstalk. Se o seu aplicativo é executado em uma imagem disponível em um repositório hospedado, você pode especificar a imagem em um arquivo `Dockerrun.aws.json v1` e omitir o `Dockerfile`.

> **Versões do `Dockerrun.aws.json`**
>
> O parâmetro `AWSEBDockerrunVersion` indica a versão do arquivo `Dockerrun.aws.json`.
>
> - As plataformas Docker AL2 e AL2 023 usam as seguintes versões do arquivo.
>   - `Dockerrun.aws.json v3`: ambientes que usam o Docker Compose.
>   - `Dockerrun.aws.json v1`: ambientes que não usam o Docker Compose.
> - O ECS executado no Amazon Linux 2 e o ECS executado no AL2 023 usam o arquivo. `Dockerrun.aws.json v2` A plataforma descontinuada ECS-The Multicontainer Docker Amazon Linux AMI (AL1) também usou essa mesma versão.

Dockerrun.aws.json v1

Chaves e valores válidos para o arquivo `Dockerrun.aws.json v1` incluem as seguintes operações:

AWSEBDockerrunVersão

(Obrigatório) Especifique o número da versão 1 se não estiver usando o Docker Compose para gerenciar a imagem.

Autenticação

(Obrigatório apenas para repositórios privados) Especifica o objeto do Amazon S3 que armazena o arquivo `.dockercfg`.

Consulte Usar imagens de um repositório privado no Elastic Beanstalk em Usar imagens de um repositório privado mais adiante neste capítulo.

Imagem

Especifica a imagem de base do Docker em um repositório Docker existente do qual você está criando um contêiner do Docker. Especifique o valor da chave Name no formato *<organization>/<image name>* para imagens no Docker Hub ou *<site>/<organization name>/<image name>* para outros sites.

Quando você especifica uma imagem no arquivo `Dockerrun.aws.json`, cada instância no ambiente do Elastic Beanstalk executa `docker pull` para executar a imagem. Opcionalmente, inclua a chave Update. O valor padrão é `true` e orienta o Elastic Beanstalk a verificar o repositório, enviar todas as atualizações para a imagem e substituir todas as imagens armazenadas em cache.

Ao usar um `Dockerfile`, não especifique a chave Image no arquivo `Dockerrun.aws.json`. O Elastic Beanstalk sempre cria e usa a imagem descrita no `Dockerfile` quando um está presente.

Portas

(Obrigatório quando você especifica a chave Image) Lista as portas para expor no contêiner do Docker. O Elastic Beanstalk ContainerPortusa o valor para conectar o contêiner Docker ao proxy reverso em execução no host.

É possível especificar várias portas de contêiner, mas o Elastic Beanstalk usa apenas a primeira porta. Ele usa essa porta para conectar seu contêiner ao proxy reverso do host e rotear solicitações da Internet pública. Se você estiver usando a`Dockerfile`, o primeiro ContainerPortvalor deve corresponder à primeira entrada na lista EXPOSE **`Dockerfile`** do.

Opcionalmente, você pode especificar uma lista de portas em HostPort. HostPortas entradas especificam as portas do host para as quais ContainerPortos valores são mapeados. Se você não especificar um HostPortvalor, o valor padrão será o ContainerPortvalor.

```
{
  "Image": {
    "Name": "image-name"
  },
  "Ports": [
    {
      "ContainerPort": 8080,
```

```
      "HostPort": 8000
    }
  ]
}
```

Volumes

Mapeie volumes de uma EC2 instância para seu contêiner do Docker. Especifique um ou mais conjuntos de volumes para mapear.

```
{
  "Volumes": [
    {
      "HostDirectory": "/path/inside/host",
      "ContainerDirectory": "/path/inside/container"
    }
  ]
...
```

Registro em log

Especifique o diretório dentro do contêiner no qual a aplicação grava logs. O Elastic Beanstalk faz upload de todos os logs neste diretório para o Amazon S3 quando você solicitar logs finais ou de pacote. Se você alternar logs para uma pasta chamada `rotated` nesse diretório, também poderá configurar o Elastic Beanstalk para fazer upload de logs alternados para o Amazon S3 para armazenamento permanente. Para obter mais informações, consulte Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

Comando

Especifique um comando a ser executado no contêiner. Se você especificar um Entrypoint, Command será adicionado como um argumento para Entrypoint. Para obter mais informações, consulte CMD na documentação do Docker.

Entrypoint

Especifique um comando padrão a ser executado quando o contêiner for iniciado. Para obter mais informações, consulte ENTRYPOINT na documentação do Docker.

O seguinte snippet é um exemplo que mostra a sintaxe do arquivo `Dockerrun.aws.json` para um único contêiner.

```
{
  "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"
}>
```

Você pode fornecer ao Elastic Beanstalk apenas com o arquivo `Dockerrun.aws.json` ou com um arquivamento `.zip` que contém os arquivos `Dockerrun.aws.json` e `Dockerfile`. Quando você fornece os dois arquivos, `Dockerfile` descreverá a imagem de Docker, e o arquivo `Dockerrun.aws.json` fornecerá informações adicionais para a implantação conforme descritas posteriormente nesta seção.

> **Note**
>
> Os dois arquivos devem estar na raiz, ou no nível superior do arquivamento `.zip`. Não crie o arquivamento em um diretório que contém os arquivos. Em vez disso, navegue para esse diretório e crie o arquivamento.
>
> Quando você fornecer os dois arquivos, não especifique uma imagem no arquivo `Dockerrun.aws.json`. O Elastic Beanstalk cria e usa a imagem descrita no `Dockerfile` e ignora a imagem especificada no arquivo `Dockerrun.aws.json`.

Criar imagens personalizadas com um Dockerfile

Será necessário criar um `Dockerfile` se você ainda não tiver uma imagem existente hospedada em um repositório.

O seguinte snippet é um exemplo do `Dockerfile`. Quando você segue as instruções em QuickStart para Docker, pode fazer upload deste `Dockerfile` como gravado. O Elastic Beanstalk executa o jogo 2048 quando você usa este `Dockerfile`.

Para obter mais informações sobre instruções que podem ser incluídas no `Dockerfile`, consulte a referência do Dockerfile no site do 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**
>
> De um único Dockerfile, você pode executar compilações em vários estágios para produzir imagens menores com uma redução significativa de complexidade. Para obter mais informações, consulte Use multi-stage builds (Usar compilações em vários estágios) no site de documentação do Docker.

Usando a ramificação da plataforma Docker gerenciada pelo ECS no Elastic Beanstalk

Este tópico fornece uma visão geral das ramificações da plataforma Docker gerenciada pelo Elastic Beanstalk ECS para Amazon Linux 2 e Amazon Linux 2023. Ele também fornece informações de configuração específicas para a plataforma gerenciada Docker ECS.

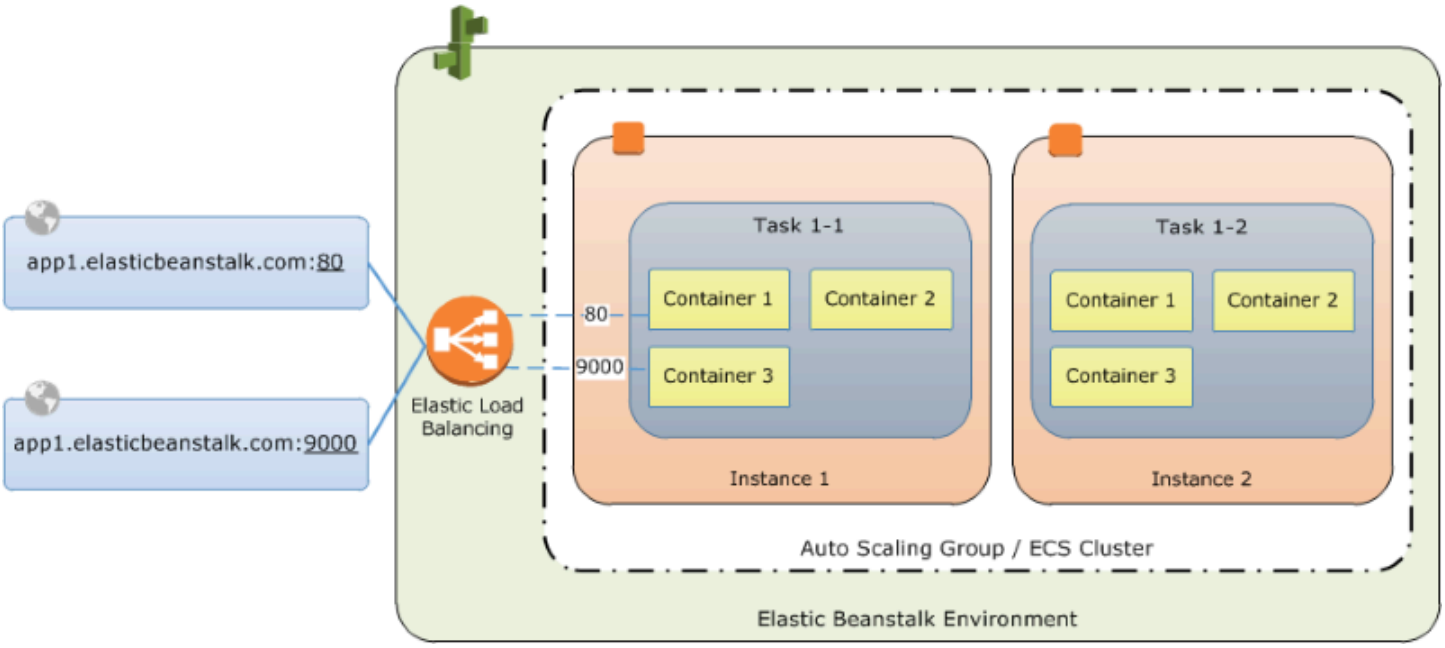
Migração do Docker de vários contêineres em AL1

Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Embora este capítulo forneça informações de configuração para essa plataforma descontinuada, é altamente recomendável que você migre para a ramificação mais recente da plataforma compatível. Se você está usando atualmente o Docker de vários contêineres aposentado em execução na ramificação da AL1 plataforma, você pode migrar para a ramificação mais recente da plataforma ECS Running on 023. AL2 A ramificação mais recente da plataforma é compatível com todos os recursos da ramificação da plataforma que foi retirada. Não é necessária nenhuma alteração no código-fonte. Para obter mais informações, consulte Migrando seu aplicativo Elastic Beanstalk do Docker de vários contêineres gerenciado pelo ECS para o ECS no Amazon Linux 2023 AL1 .

Visão geral da plataforma Docker gerenciada pelo ECS

O Elastic Beanstalk usa o Amazon Elastic Container Service (Amazon ECS) para coordenar implantações de contêiner em ambientes do Docker gerenciado pelo ECS. O Amazon ECS fornece ferramentas para gerenciar um cluster de instâncias executando contêineres do Docker. O Elastic Beanstalk cuida das tarefas do Amazon ECS, incluindo criação de cluster, definição de tarefas e execução. Cada uma das instâncias do ambiente executa o mesmo conjunto de contêineres que são definidos em um arquivo `Dockerrun.aws.json` v2. Para aproveitar ao máximo o Docker, o Elastic Beanstalk permite criar um ambiente em que suas instâncias da EC2 Amazon executam vários contêineres do Docker lado a lado.

O diagrama a seguir mostra um exemplo de ambiente do Elastic Beanstalk configurado com três contêineres Docker em execução em EC2 cada instância da Amazon em um grupo de Auto Scaling:



Recursos do Amazon ECS criados pelo Elastic Beanstalk

Quando você cria um ambiente usando a plataforma Docker gerenciada pelo ECS, o Elastic Beanstalk cria e configura automaticamente vários recursos do Amazon Elastic Container Service enquanto cria o ambiente. Ao fazer isso, ele cria os contêineres necessários em cada EC2 instância da Amazon.

- Cluster do Amazon ECS: as instâncias de contêiner no Amazon ECS são organizadas em clusters. Quando é ele usado com o Elastic Beanstalk, sempre é criado um cluster para cada ambiente do Docker gerenciado pelo ECS. Um cluster ECS também contém provedores de capacidade de grupo do Auto Scaling e outros recursos.
- Definição de tarefas do Amazon ECS: o Elastic Beanstalk usa o arquivo `Dockerrun.aws.json` v2 no projeto para gerar a definição de tarefas do Amazon ECS que é usada para configurar instâncias de contêiner no ambiente.
- Tarefa do Amazon ECS: o Elastic Beanstalk se comunica com o Amazon ECS para executar uma tarefa em todas as instâncias do ambiente para coordenar a implantação de contêineres. Em um ambiente escalável, o Elastic Beanstalk inicia uma nova tarefa sempre que uma instância é adicionada ao cluster.
- Agente de contêiner do Amazon ECS: o agente é executado em um contêiner do Docker nas instâncias do ambiente. O agente consulta o serviço do Amazon ECS e aguarda uma tarefa para executar.

- Volumes de dados do Amazon ECS — Além dos volumes que você define na `Dockerrun.aws.json` v2, o Elastic Beanstalk insere definições de volume na definição da tarefa para facilitar a coleta de registros.

  O Elastic Beanstalk cria volumes de log na instância de contêiner, um para cada contêiner, em `/var/log/containers/`*`containername`*. Esses volumes são denominados `awseb-logs-`*`containername`* e são fornecidos para os contêineres que devem ser montados. Consulte Formato de definição de contêiner para detalhes sobre como montá-los.

Para obter mais informações sobre os recursos do Amazon ECS, consulte o Guia do desenvolvedor do Amazon Elastic Container Service.

Arquivo **`Dockerrun.aws.json`** v2

As instâncias do contêiner exigem um arquivo de configuração chamado `Dockerrun.aws.json`. Instâncias de contêiner se referem às EC2 instâncias da Amazon executando o Docker gerenciado pelo ECS em um ambiente do Elastic Beanstalk. Esse arquivo é específico do Elastic Beanstalk e pode ser usado individualmente ou combinado com código-fonte e conteúdo em um pacote de fonte para criar um ambiente na plataforma Docker.

> **Note**
>
> O formato da versão 2 `Dockerrun.aws.json` adiciona suporte para vários contêineres por EC2 instância da Amazon e só pode ser usado com uma plataforma Docker gerenciada pelo ECS. O formato difere significativamente das outras versões do arquivo de configuração compatíveis com as ramificações da plataforma Docker que não são gerenciadas pelo ECS.

Consulte o `Dockerrun.aws.json` v2 para obter detalhes sobre o formato atualizado e um arquivo de exemplo.

Imagens de Docker

A plataforma Docker gerenciada pelo ECS para o Elastic Beanstalk exige que as imagens sejam pré-criadas e armazenadas em um repositório de imagens online público ou privado antes de criar um ambiente do Elastic Beanstalk.

> **Note**
>
> A criação de imagens personalizadas durante a implantação com um `Dockerfile` não é compatível com a plataforma Docker gerenciada pelo ECS no Elastic Beanstalk. Crie suas imagens e implante-as em um repositório online antes de criar um ambiente do Elastic Beanstalk.

Especifique as imagens por nome em `Dockerrun.aws.json` v2.

Para configurar o Elastic Beanstalk para autenticar em um repositório privado, inclua o parâmetro `authentication` no arquivo `Dockerrun.aws.json` v2.

Falha nas implantações de contêiner

Se uma tarefa do Amazon ECS falhar, um ou mais contêineres em seu ambiente Elastic Beanstalk não serão iniciados. O Elastic Beanstalk não reverte ambientes de vários contêineres devido a falha em uma tarefa do Amazon ECS. Se um contêiner não for iniciado no ambiente, implante novamente a versão atual ou uma versão funcional anterior do console do Elastic Beanstalk.

Para implantar uma versão existente

1. Abra o console do Elastic Beanstalk na região do seu ambiente.
2. Clique em Actions, à direita do nome do aplicativo, e depois clique em View application versions.
3. Selecione uma versão do aplicativo e clique em Deploy.

Estendendo as plataformas Docker baseadas em ECS para o Elastic Beanstalk

O Elastic Beanstalk oferece recursos de extensibilidade que permitem que você aplique seus próprios comandos, scripts, software e configurações às implantações de seus aplicativos. O fluxo de trabalho de implantação para as ramificações da plataforma ECS AL2 e AL2 023 varia um pouco das outras plataformas baseadas em Linux. Para obter mais informações, consulte Fluxo de trabalho de implantação de instâncias para o ECS em execução no Amazon Linux 2 e versões posteriores.

Configuração do Docker gerenciado pelo ECS para o Elastic Beanstalk

Este capítulo explica como configurar um ambiente Docker gerenciado pelo ECS. A lista a seguir resume os itens de configuração explicados neste capítulo.

- `Dockerrun.aws.jsonv2` — Esse arquivo de configuração especifica seu repositório de imagens e o nome de suas imagens do Docker, entre outros componentes.
- EC2 Função do perfil da instância — Se você tem um perfil de instância personalizado, explicamos como configurá-lo para que as permissões necessárias para o ECS gerenciar seus contêineres permaneçam atualizadas.
- Ouvintes do Elastic Load Balancing — Você precisará configurar vários ouvintes do Elastic Load Balancing se precisar que seu ambiente ofereça suporte ao tráfego de entrada para proxies ou outros serviços que não são executados na porta HTTP padrão.

Tópicos

- Configuração do arquivo Dockerrun.aws.json v2
- Política gerenciada de contêineres e função da EC2 instância
- Usar vários listeners do Elastic Load Balancing

Configuração do arquivo Dockerrun.aws.json v2

`Dockerrun.aws.json v2` é um arquivo de configuração do Elastic Beanstalk que descreve como implantar um conjunto de contêineres Docker hospedados em um cluster ECS em um ambiente Elastic Beanstalk. A plataforma Elastic Beanstalk cria uma definição de tarefa do ECS, que inclui uma definição de contêiner do ECS. Essas definições são descritas no arquivo de configuração do `Dockerrun.aws.json`.

A definição do contêiner no `Dockerrun.aws.json` arquivo descreve os contêineres a serem implantados em cada EC2 instância da Amazon no cluster do ECS. Nesse caso, uma EC2 instância da Amazon também é chamada de instância de contêiner host, porque hospeda os contêineres do Docker. O arquivo de configuração também descreve os volumes de dados a serem criados na instância de contêiner de host para os contêineres do Docker montarem. Para obter mais informações e um diagrama dos componentes em um ambiente Docker gerenciado pelo ECS no Elastic Beanstalk, consulte o Visão geral da plataforma Docker gerenciada pelo ECS anterior neste capítulo.

Um arquivo `Dockerrun.aws.json` pode ser usado sozinho ou zipado com o código-fonte adicional em um único arquivo. O código-fonte arquivado com um `Dockerrun.aws.json` é implantado nas instâncias de EC2 contêiner da Amazon e acessível no `/var/app/current/` diretório.

Tópicos

- Dockerrun.aws.json v2
- Formato do volume
- Formato ARN da função de execução
- Formato de definição de contêiner
- Formato de autenticação — Usar imagens de um repositório privado
- Exemplo Dockerrun.aws.json v2

`Dockerrun.aws.json` v2

O arquivo `Dockerrun.aws.json` inclui as seguintes seções:

AWSEBDockerrunVersão

Especifica o número da versão como o valor 2 para ambientes do Docker gerenciado pelo ECS.

executionRoleArn

Especifica as funções do IAM de execução de tarefas para diferentes propósitos e serviços associados à sua conta. Para que seu aplicativo use variáveis de ambiente do Elastic Beanstalk armazenadas como segredos, você precisará especificar o ARN de uma função de execução de tarefas que conceda as permissões necessárias. Outros casos de uso comuns também podem exigir esse parâmetro. Para obter mais informações, consulte Formato ARN da função de execução.

volumes

Cria volumes a partir de pastas na instância de EC2 contêiner da Amazon ou do seu pacote de origem (implantado em). `/var/app/current` Monte esses volumes em caminhos dentro de seus contêineres do Docker usando `mountPoints` na seção `containerDefinitions`.

containerDefinitions

Uma gama de definições de contêiner.

autenticação (opcional)

A localização no Amazon S3 de um arquivo `.dockercfg` que contém dados de autenticação para um repositório privado.

As seções de definição de contêiner e de volumes do `Dockerrun.aws.json` usam a mesma formatação que as seções correspondentes de um arquivo de definição de tarefas do Amazon ECS. Para obter mais informações sobre o formato de definição de tarefa e uma lista completa de parâmetros de definição de tarefa, consulte Definições de tarefa do Amazon ECS no Guia do desenvolvedor do Amazon Elastic Container Service.

Formato do volume

O parâmetro volume cria volumes de qualquer pasta na instância de EC2 contêiner da Amazon ou do seu pacote de origem (implantado em). `/var/app/current`

Os volumes são especificados no seguinte formato:

```
"volumes": [
    {
      "name": "volumename",
      "host": {
        "sourcePath": "/path/on/host/instance"
      }
    }
  ],
```

Monte esses volumes em caminhos dentro dos seus contêineres do Docker usando `mountPoints` na definição de contêiner.

O Elastic Beanstalk configura volumes adicionais para logs, um para cada contêiner. Eles devem ser montados pelos seus contêineres do Docker para gravar logs na instância do host.

Para obter mais detalhes, consulte o campo `mountPoints` na seção Formato de definição de contêiner a seguir.

Formato ARN da função de execução

Para que seu aplicativo use variáveis de ambiente do Elastic Beanstalk armazenadas como segredos, você precisará especificar uma função do IAM de execução de tarefas. A função deve

conceder ao contêiner do Amazon ECS permissão para fazer chamadas de AWS API em seu nome usando AWS Secrets Manager segredos ou AWS Systems Manager parâmetros do Parameter Store para referenciar dados confidenciais. Para obter instruções sobre como criar uma função IAM de execução de tarefas com as permissões necessárias para sua conta, consulte a função IAM de execução de tarefas do Amazon ECS no Amazon Elastic Container Service Developer Guide.

```
{
"AWSEBDockerrunVersion": 2,
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
```

Permissões adicionais necessárias para a plataforma Docker gerenciada pelo Amazon ECS

EC2 concessões de perfil de instância **`iam:PassRole`** ao ECS

Para que seu perfil de EC2 instância possa conceder essa função ao contêiner do ECS, você deve incluir a `iam:PassRole` permissão demonstrada no exemplo a seguir. `iam:PassRole`Isso permite que as EC2 instâncias passem a função de execução da tarefa para o contêiner do ECS.

Neste exemplo, limitamos a EC2 instância para passar a função somente para o serviço ECS. Embora essa condição não seja obrigatória, nós a adicionamos para seguir as melhores práticas para reduzir o escopo da permissão compartilhada. Conseguimos isso com o `Condition` elemento.

> **Note**
>
> Qualquer uso da função de execução de tarefas do ECS IAM requer a `iam:PassRole` permissão. Há outros casos de uso comuns que exigem a função de serviço gerenciado de execução de tarefas do ECS. Para obter mais informações, consulte a função IAM de execução de tarefas do Amazon ECS no Amazon Elastic Container Service Developer Guide.

Example política com **`iam:PassRole`** permissão

```
{
    "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"
                }
            }
        }
    ]
}
```

Concedendo acesso a segredos e parâmetros ao agente de contêineres do Amazon ECS

A função IAM de execução de tarefas do Amazon ECS também precisa de permissões para acessar os segredos e os armazenamentos de parâmetros. Semelhante aos requisitos da função de perfil da EC2 instância, o agente de contêiner do ECS exige permissão para extrair os recursos necessários do Secrets Manager ou do Systems Manager. Para obter mais informações, consulte as permissões do Secrets Manager ou do Systems Manager no Amazon Elastic Container Service Developer Guide

Concedendo acesso a segredos e parâmetros às instâncias do Elastic Beanstalk EC2

Para oferecer suporte a segredos configurados como variáveis de ambiente, você também precisará adicionar permissões ao seu perfil de EC2 instância. Para obter mais informações, consulte Buscando segredos e parâmetros para variáveis de ambiente do Elastic Beanstalk e Permissões do IAM necessárias para o Secrets Manager.

Os exemplos a seguir combinam o `iam:PassRole` exemplo anterior com os exemplos fornecidos na referênciaPermissões do IAM necessárias para o Secrets Manager. Eles adicionam as permissões que as EC2 instâncias exigem para acessar AWS Secrets Manager e AWS Systems Manager armazenar para recuperar os segredos e os dados de parâmetros para inicializar as variáveis de ambiente do Elastic Beanstalk que foram configuradas para segredos.

Example Política do Secrets Manager combinada com **`iam:PassRole`** permissão

```
{
    "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 Política do Systems Manager combinada com **`iam:PassRole`** permissão

```
{
    "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 de definição de contêiner

Os exemplos a seguir mostram um subconjunto de parâmetros comumente usados na seção Definições do contêiner. Há outros parâmetros opcionais disponíveis.

A plataforma Beanstalk cria uma definição de tarefa do ECS, que inclui uma definição de contêiner do ECS. O Beanstalk suporta um subconjunto de parâmetros para a definição do contêiner ECS. Para obter mais informações, consulte Definições de contêiner no Guia do desenvolvedor do Amazon Elastic Container Service.

Um arquivo `Dockerrun.aws.json` contém um ou mais objetos de definição de contêiner com os seguintes campos:

name

O nome do contêiner. Consulte Parâmetros de definição de contêiner padrão para obter informações sobre o tamanho máximo e os caracteres permitidos.

imagem

O nome de uma imagem do Docker em um repositório do Docker online a partir do qual você está criando um contêiner do Docker. Observe as seguintes convenções:

- As imagens em repositórios oficiais no Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
- As imagens em outros repositórios no Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`.
- As imagens em outros repositórios online são ainda mais qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).

ambiente

Uma gama de variáveis de ambiente a serem passadas para o contêiner.

Por exemplo, a seguinte entrada define uma variável de ambiente com o nome **`Container`** e o valor **`PHP`**:

```
"environment": [
  {
    "name": "Container",
    "value": "PHP"
  }
],
```

essential

Verdadeiro caso a tarefa deva ser interrompida se o contêiner falhar. Os contêineres não essenciais podem concluir ou falhar sem afetar o resto dos contêineres na instância.

memory

Quantidade de memória na instância de contêiner para reservar para o contêiner. Especifique um inteiro diferente de zero para um ou ambos os parâmetros `memory` ou `memoryReservation` em definições de contêiner.

memoryReservation

O limite flexível (em MiB) de memória a ser reservado para o contêiner. Especifique um inteiro diferente de zero para um ou ambos os parâmetros `memory` ou `memoryReservation` em definições de contêiner.

mountPoints

Volumes da instância de EC2 contêiner da Amazon a serem montados e o local no sistema de arquivos de contêiner Docker no qual montá-los. Quando você montar volumes que contêm conteúdo do aplicativo, o contêiner pode ler os dados carregados em seu pacote de origem. Quando você montar volumes de log para gravar dados de log, o Elastic Beanstalk pode reunir dados de log a partir desses volumes.

O Elastic Beanstalk cria volumes de log na instância de contêiner, um para cada contêiner do Docker, em `/var/log/containers/`*`containername`*. Esses volumes são chamados `awseb-logs-`*`containername`* e devem ser montados no local dentro da estrutura de arquivos do contêiner onde os logs são gravados.

Por exemplo, o ponto de montagem a seguir mapeia a localização de log nginx no contêiner para o volume gerado pelo Elastic Beanstalk para o contêiner `nginx-proxy`.

```
{
  "sourceVolume": "awseb-logs-nginx-proxy",
  "containerPath": "/var/log/nginx"
}
```

portMappings

Mapeia portas de rede no contêiner para portas no host.

links

Lista de contêineres aos quais vincular. Os contêineres vinculados podem descobrir uns aos outros e se comunicar com segurança.

volumesFrom

Monte todos os volumes de um contêiner diferente. Por exemplo, para montar volumes de um contêiner chamado `web`:

```
"volumesFrom": [
  {
    "sourceContainer": "web"
  }
],
```

Formato de autenticação — Usar imagens de um repositório privado

A seção `authentication` contém dados de autenticação para um repositório privado. Essa entrada é opcional.

Adicione as informações sobre o bucket do Amazon S3 que contém o arquivo de autenticação no parâmetro `authentication` do arquivo `Dockerrun.aws.json`. Certifique-se de que o parâmetro `authentication` contenha uma chave e um bucket válidos do Amazon S3. O bucket do Amazon S3 deve ser hospedado na mesma região do ambiente que o está usando. O Elastic Beanstalk não fará download dos arquivos de buckets do Amazon S3 hospedados em outras regiões.

Usa o seguinte formato:

```
"authentication": {
```

```
    "bucket": "amzn-s3-demo-bucket",
    "key": "mydockercfg"
  },
```

Para obter informações sobre a geração e o upload do arquivo de autenticação, consulte Usar imagens de um repositório privado no Elastic Beanstalk.

Exemplo Dockerrun.aws.json v2

O snippet a seguir é um exemplo que ilustra a sintaxe do arquivo `Dockerrun.aws.json` para uma instância com dois contêineres.

```
{
  "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"
        }
      ]
    }
  ]
}
```

Política gerenciada de contêineres e função da EC2 instância

Quando você cria um ambiente no console do Elastic Beanstalk, ele solicita que você crie um perfil de instância padrão que inclua a política gerenciada.

`AWSElasticBeanstalkMulticontainerDocker` Portanto, inicialmente, seu perfil de EC2 instância padrão deve incluir essa política gerenciada. Se seu ambiente usa uma função de perfil de EC2 instância personalizada em vez da padrão, certifique-se de que a política gerenciada `AWSElasticBeanstalkMulticontainerDocker` esteja anexada para que as permissões necessárias para o gerenciamento de contêineres permaneçam up-to-date.

O Elastic Beanstalk usa uma AMI otimizada para Amazon ECS com um agente de contêiner do Amazon ECS que é executado em um contêiner do Docker. O agente se comunica com o Amazon ECS para coordenar as implantações de contêiner. Para se comunicar com o Amazon ECS, cada EC2 instância da Amazon deve ter as permissões do IAM correspondentes, que são especificadas nessa política gerenciada. Consulte o Guia AWSElasticBeanstalkMulticontainerDockerde referência de políticas AWS gerenciadas para ver essas permissões.

Se você usa variáveis de ambiente do Elastic Beanstalk configuradas para acessar segredos ou parâmetros AWS Secrets Manager armazenados AWS Systems Manager no Parameter Store, você deve EC2 personalizar seu perfil de instância com permissões adicionais. Para obter mais informações, consulte Formato ARN da função de execução.

Usar vários listeners do Elastic Load Balancing

Você pode configurar vários listeners do Elastic Load Balancing em um ambiente do Docker gerenciado pelo ECS para acomodar o tráfego de entrada para proxies ou outros serviços que não são executados na porta HTTP padrão.

Crie uma pasta `.ebextensions` no seu pacote de origem e adicione um arquivo com a extensão `.config`. O exemplo a seguir mostra um arquivo de configuração que cria um listener do Elastic Load Balancing na porta 8080.

`.ebextensions/elb-listener.config`

```
option_settings:
  aws:elb:listener:8080:
    ListenerProtocol: HTTP
    InstanceProtocol: HTTP
    InstancePort: 8080
```

Se o seu ambiente estiver sendo executado em uma Amazon Virtual Private Cloud (Amazon VPC) personalizada criada por você, o Elastic Beanstalk cuidará do restante. Em uma VPC padrão, é

preciso configurar o grupo de segurança da instância para permitir a entrada do balanceador de carga. Adicione um segundo arquivo de configuração que inclui uma regra de entrada no security group:

`.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"] }
```

Para obter mais informações sobre o formato de arquivo de configuração, consulte Adicionar e personalizar recursos do ambiente do Elastic Beanstalk e Configurações de opção.

Além de adicionar um listener à configuração do Elastic Load Balancing e abrir uma porta no grupo de segurança, é preciso mapear a porta na instância do host para uma porta no contêiner do Docker, na seção `containerDefinitions` do arquivo `Dockerrun.aws.json` v2. O trecho a seguir mostra um exemplo:

```
"portMappings": [
  {
    "hostPort": 8080,
    "containerPort": 8080
  }
]
```

Consulte `Dockerrun.aws.json` v2 para obter detalhes sobre o formato de arquivo `Dockerrun.aws.json` v2.

Criação de um ambiente Docker gerenciado pelo ECS com o console do Elastic Beanstalk

Este tutorial detalha a configuração do contêiner e a preparação do código-fonte para um ambiente Docker gerenciado pelo ECS que usa dois contêineres.

Os contêineres, um aplicativo PHP e um proxy nginx são executados lado a lado em cada uma das instâncias do Amazon Elastic Compute Cloud (EC2Amazon) em um ambiente do Elastic Beanstalk. Depois de criar o ambiente e verificar se os aplicativos estão funcionando, conecte-se a uma instância de contêiner para ver como tudo isso se encaixa.

Seções

- Definir contêineres do Docker gerenciado pelo ECS
- Adicionar conteúdo
- Implantar no Elastic Beanstalk
- Conectar-se a uma instância de contêiner
- Inspecionar o agente de contêiner do Amazon ECS

Definir contêineres do Docker gerenciado pelo ECS

O primeiro passo na criação de um novo ambiente do Docker é criar um diretório para os dados do aplicativo. Essa pasta pode estar localizada em qualquer lugar da sua máquina local e ter qualquer nome que você escolher. Além de um arquivo de configuração de contêiner, essa pasta terá o conteúdo do qual o upload será feito para o Elastic Beanstalk e implantado no ambiente.

> **Note**
>
> Todo o código deste tutorial está disponível no repositório awslabs em. GitHub https://github.com/awslabs/eb-docker-nginx-proxy

O arquivo que o Elastic Beanstalk usa para configurar os contêineres em uma instância da EC2 Amazon é um arquivo de texto em formato JSON chamado v2. `Dockerrun.aws.json` As versões da plataforma Docker gerenciada pelo ECS usam um formato de versão 2 desse arquivo. Esse formato apenas pode ser usado com a plataforma Docker gerenciada pelo ECS, pois difere significativamente das outras versões de arquivo de configuração compatíveis com as ramificações da plataforma Docker que não são gerenciadas pelo ECS.

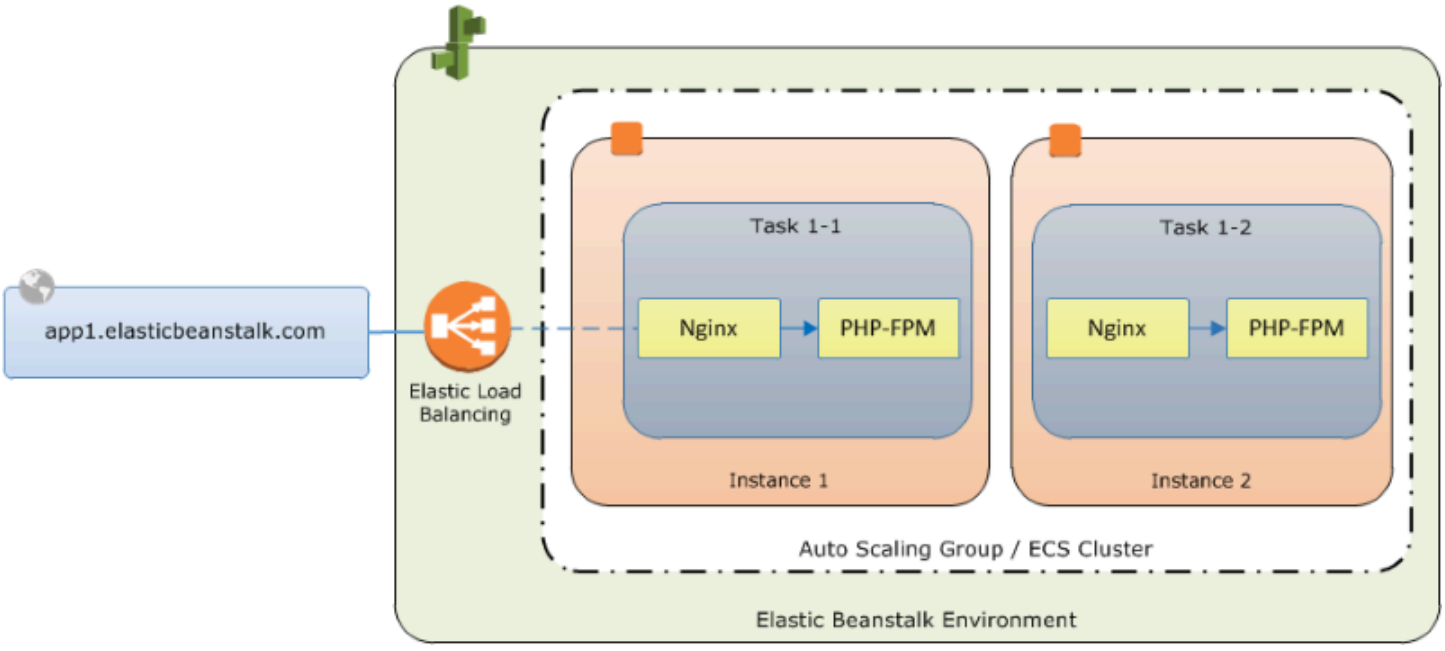
Crie um arquivo de texto `Dockerrun.aws.json` v2 com esse nome na raiz da sua aplicação e adicione o seguinte texto:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Este exemplo de configuração define dois contêineres, um site em PHP com proxy nginx na frente dele. Esses dois contêineres serão executados lado a lado nos contêineres do Docker em cada instância do ambiente do Elastic Beanstalk, acessando conteúdo compartilhado (conteúdo do site) dos volumes na instância do host, que também são definidos nesse arquivo. Os contêineres são criados a partir de imagens hospedadas em repositórios oficiais no Docker Hub. O ambiente resultante se parece com este:



Os volumes definidos na configuração correspondem ao conteúdo que você vai criar em seguida e fazer upload como parte do pacote de origem do aplicativo. Os contêineres acessam o conteúdo no host montando volumes na seção `mountPoints` das definições de contêiner.

Para obter mais informações sobre o formato `Dockerrun.aws.json` v2 e seus parâmetros, consulte Formato de definição de contêiner.

Adicionar conteúdo

Em seguida, você vai adicionar conteúdo para seu site PHP exibir para os visitantes, e um arquivo de configuração para o 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;
  }
}
```

Implantar no Elastic Beanstalk

Agora, a sua pasta do aplicativo contém os seguintes arquivos:

```
### Dockerrun.aws.json
### php-app
#   ### index.php
#   ### static.html
### proxy
    ### conf.d
        ### default.conf
```

Isso é tudo de que você precisa para criar o ambiente do Elastic Beanstalk. Crie um arquivamento `.zip` dos arquivos e pastas acima (não incluindo a pasta do projeto de nível superior). Para criar o arquivamento no Windows Explorer, selecione o conteúdo da pasta do projeto, clique com o botão direito do mouse, selecione Enviar para e clique em Pasta compactada (zipada)

> **Note**
>
> Para obter informações sobre a estrutura de arquivos e instruções necessárias para criar arquivamentos em outros ambientes, consulte Crie um pacote de origem da aplicação Elastic Beanstalk

Depois, faça upload do pacote de origem para o Elastic Beanstalk e crie o ambiente. Em Platform (Plataforma), selecione Docker. Em Ramificação da plataforma, selecione Docker de vários contêineres em execução no Amazon Linux 2023 de 64 bits.

Para iniciar um ambiente (console)

1. Abra o console do Elastic Beanstalk com este link pré-configurado: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced

2. Em Plataforma, selecione a plataforma e a ramificação da plataforma que correspondem à linguagem usada pelo aplicativo ou a plataforma do Docker para aplicativos baseados em contêiner.
3. Em Application code (Código do aplicativo), escolha Upload your code (Fazer upload do código).
4. Escolha Local file (Arquivo local) e Choose file (Escolher arquivo) e abra o pacote de origem.
5. Selecione Review and launch.
6. Revise as configurações disponíveis e, em seguida, escolha Create app.

O console do Elastic Beanstalk redireciona você para o painel de gerenciamento do novo ambiente. Essa tela mostra o status de integridade do ambiente e saída de eventos pelo serviço do Elastic Beanstalk. Quando o status estiver verde, clique no URL ao lado do nome do ambiente para ver seu novo site.

Conectar-se a uma instância de contêiner

Em seguida, você se conectará a uma EC2 instância da Amazon em seu ambiente do Elastic Beanstalk para ver algumas das partes móveis em ação.

A maneira mais fácil de se conectar a uma instância em seu ambiente é usar a CLI do EB. Para usá-la, instale a CLI do EB, se você ainda não tiver feito isso. Você também precisará configurar seu ambiente com um par de chaves Amazon EC2 SSH. Para fazer isso, use a página de configuração de segurança do console ou o comando eb init da CLI do EB. Use o comando eb ssh da CLI do EB para se conectar a uma instância do ambiente.

Agora que você está conectado a uma EC2 instância da Amazon que hospeda seus contêineres docker, você pode ver como as coisas estão configuradas. Execute `ls` em `/var/app/current`:

```
[ec2-user@ip-10-0-0-117 ~]$ ls /var/app/current
Dockerrun.aws.json  php-app  proxy
```

Esse diretório contém os arquivos do pacote de origem do qual você fez upload para o Elastic Beanstalk durante a criação do 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
```

Este é o local em que os logs são criados na instância de contêiner e coletados pelo Elastic Beanstalk. O Elastic Beanstalk cria um volume nesse diretório para cada contêiner, que é montado no local do contêiner onde os logs são gravados.

Você também pode dar uma olhada no Docker para ver os contêineres em execução com `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
```

Isso mostra os dois contêineres implantados em execução, bem como o agente de contêiner do Amazon ECS que coordenou a implantação.

Inspecionar o agente de contêiner do Amazon ECS

EC2 As instâncias da Amazon em um ambiente Docker gerenciado pelo ECS no Elastic Beanstalk executam um processo de agente em um contêiner Docker. Esse agente se conecta ao Amazon ECS Service para coordenar implantações de contêiner. Essas implantações são executadas como tarefas no Amazon ECS, que são configuradas nos arquivos de definição de tarefas. O Elastic Beanstalk cria esses arquivos de definição de tarefas com base no `Dockerrun.aws.json` cujo upload será feito em um pacote de origem.

Verifique o status do agente de contêiner com uma solicitação HTTP get para `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)"
```

```
}
```

Essa estrutura mostra o nome do cluster do Amazon ECS e o ARN (Amazon Resource Name) da instância do cluster (a instância da EC2 Amazon à qual você está conectado).

Para obter mais informações, faça uma solicitação HTTP get ao `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"
                }
             ]
          }
       ]
    }
 ]
}
```

Essa estrutura descreve a tarefa que é executada para implantar os dois contêineres do Docker deste projeto de exemplo deste tutorial. São exibidas as seguintes informações:

- KnownStatus— O `RUNNING` status indica que os contêineres ainda estão ativos.
- Família: o nome da definição de tarefa que o Elastic Beanstalk criou a partir de `Dockerrun.aws.json`.
- Versão: a versão da definição de tarefa. Ela é incrementada cada vez que o arquivo de definição de tarefas é atualizado.
- Contêineres: informações sobre os contêineres em execução na instância.

Mais informações estão disponíveis no próprio serviço Amazon ECS, que você pode chamar usando o AWS Command Line Interface. Para obter instruções sobre como usar o AWS CLI com o Amazon ECS e informações sobre o Amazon ECS em geral, consulte o Guia do usuário do Amazon ECS.

Migrando seu aplicativo Elastic Beanstalk do Docker de vários contêineres gerenciado pelo ECS para o ECS no Amazon Linux 2023 AL1

> **Note**
>
> Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 .

Este tópico orienta você na migração de seus aplicativos da filial descontinuada da plataforma Docker de vários contêineres em execução no Amazon Linux de 64 bits para o ECS executado em 023 de 64 bits. AL2 Essa ramificação de plataforma de destino é atual e compatível. Assim como a ramificação anterior do Docker de vários contêineres, a AL1 ramificação mais recente da AL2plataforma ECS 023 usa o Amazon ECS para coordenar a implantação de vários contêineres do Docker em um cluster do Amazon ECS em um ambiente do Elastic Beanstalk. A nova ramificação da plataforma ECS AL2 023 oferece suporte a todos os recursos da ramificação anterior da plataforma Docker AL1 de vários contêineres. Além disso, o mesmo arquivo `Dockerrun.aws.json` v2 é compatível.

Seções

- Migrar com o console do Elastic Beanstalk
- Migre com o AWS CLI

Migrar com o console do Elastic Beanstalk

Para migrar usando o console do Elastic Beanstalk, implante o mesmo código-fonte em um novo ambiente baseado na ramificação da plataforma ECS Running on 023. AL2 Não é necessária nenhuma alteração no código-fonte.

Para migrar para a ramificação da plataforma ECS em execução no Amazon Linux 2023

1. Usando o fonte da aplicação que já está implantada no ambiente antigo, crie um pacote do fonte da aplicação. Você pode usar o mesmo pacote do fonte da aplicação e o mesmo arquivo `Dockerrun.aws.json` v2.
2. Crie um novo ambiente usando a ramificação da plataforma ECS em execução no Amazon Linux 2023. Use o pacote do fonte da etapa anterior para código da aplicação. Para ter etapas

mais detalhadas, consulte Implantar no Elastic Beanstalk no Tutorial do Docker gerenciado pelo ECS incluído anteriormente neste capítulo.

Migre com o AWS CLI

Você também tem a opção de usar o AWS Command Line Interface (AWS CLI) para migrar seu ambiente atual do Docker de vários contêineres Amazon Linux Docker para a nova AL2 ramificação da plataforma ECS 023. Nesse caso, não é necessário criar um novo ambiente nem reimplantar o código-fonte. Você só precisa executar o comando AWS CLI update-environment. Ele executará uma atualização da plataforma para migrar o ambiente existente para a ramificação da plataforma Amazon Linux 2023 do ECS.

Use a sintaxe a seguir para migrar o ambiente para a nova ramificação da plataforma.

```
aws elasticbeanstalk update-environment \
--environment-name my-env \
--solution-stack-name "64bit Amazon Linux 2023 version running ECS" \
--region my-region
```

Veja a seguir um exemplo do comando para migrar o ambiente beta-101 para a versão 3.0.0 da ramificação da plataforma Amazon Linux 2023 do ECS na região 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
```

O parâmetro `solution-stack-name` fornece a ramificação da plataforma e sua versão. Use a versão mais recente da ramificação da plataforma especificando o nome da pilha de soluções adequado. A versão de cada ramificação da plataforma está incluída no nome da pilha de soluções, conforme mostrado no exemplo acima. Para obter uma lista das pilhas de soluções mais atuais para a plataforma Docker, consulte Plataformas compatíveis no guia Plataformas do AWS Elastic Beanstalk .

> **Note**
>
> O list-available-solution-stackscomando fornece uma lista das versões da plataforma disponíveis para sua conta em uma AWS região.

```
aws elasticbeanstalk list-available-solution-stacks --region us-east-1 --query
 SolutionStacks
```

Para saber mais sobre o AWS CLI, consulte o Guia AWS Command Line Interface do usuário. Para obter mais informações sobre AWS CLI comandos para o Elastic Beanstalk, AWS CLI consulte a Referência de comandos para o Elastic Beanstalk.

Usar imagens de um repositório privado no Elastic Beanstalk

Este tópico descreve como autenticar em um repositório de imagens online privado com o Elastic Beanstalk. O Elastic Beanstalk deve se autenticar no registro online antes de poder extrair e implantar suas imagens. Há várias opções de configuração.

Usar imagens de um repositório do Amazon ECR

Você pode armazenar suas imagens personalizadas do Docker AWS com o Amazon Elastic Container Registry (Amazon ECR).

Quando você armazena suas imagens do Docker no Amazon ECR, o Elastic Beanstalk autentica automaticamente no registro do Amazon ECR no perfil de instância do ambiente. Portanto, você precisará fornecer às suas instâncias permissão para acessar as imagens no seu repositório do Amazon ECR. Para fazer isso, adicione permissões ao perfil de instância do seu ambiente anexando a política EC2 ContainerRegistryReadOnly gerenciada da Amazon ao perfil da instância. Isso fornece acesso somente leitura a todos os repositórios do Amazon ECR na sua conta. Você também tem a opção de acessar apenas um único repositório usando o seguinte modelo para criar uma política personalizada:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "AllowEbAuth",
        "Effect": "Allow",
        "Action": [
          "ecr:GetAuthorizationToken"
        ],
```

```
        "Resource": [
          "*"
        ]
      },
      {
        "Sid": "AllowPull",
        "Effect": "Allow",
        "Resource": [
          "arn:aws:ecr:us-east-2:account-id:repository/repository-name"
        ],
        "Action": [
          "ecr:GetAuthorizationToken",
          "ecr:BatchCheckLayerAvailability",
          "ecr:GetDownloadUrlForLayer",
          "ecr:GetRepositoryPolicy",
          "ecr:DescribeRepositories",
          "ecr:ListImages",
          "ecr:BatchGetImage"
        ]
      }
    ]
  }
```

Substitua o nome de recurso da Amazon (ARN) na política acima pelo ARN de seu repositório.

Você precisará especificar as informações da imagem no arquivo `Dockerrun.aws.json`. A configuração será diferente dependendo da plataforma usada.

Para a plataforma Docker gerenciada pelo ECS, use a chave `image` em um objeto de definição de contêiner :

```
"containerDefinitions": [
        {
        "name": "my-image",
        "image": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
```

Para a plataforma Docker, consulte a imagem por URL. O URL vai na definição `Image` do seu arquivo `Dockerrun.aws.json`:

```
  "Image": {
      "Name": "account-id.dkr.ecr.us-east-2.amazonaws.com/repository-name:latest",
      "Update": "true"
```

```
    },
```

Usando o AWS Systems Manager (SSM) Parameter Store

Você pode configurar o Elastic Beanstalk para fazer login no repositório privado antes que ele inicie o processo de implantação. Isso permite que o Elastic Beanstalk acesse as imagens do repositório e implante essas imagens no ambiente do Elastic Beanstalk.

Essa configuração inicia eventos na fase de pré-compilação do processo de implantação do Elastic Beanstalk. Você configurou isso no diretório de configuração .ebextentions. A configuração usa scripts de hook de plataforma que chamam docker login para autenticar no registro online que hospeda o repositório privado. Os scripts de gancho da plataforma leem com segurança os dados de credenciais das variáveis de ambiente do Elastic Beanstalk que são inicializadas por uma configuração que origina os valores do Parameter Store. AWS Systems Manager Suas plataformas Docker gerenciadas pelo Elastic Beanstalk e pelo ECS devem ser uma versão lançada em ou após 26 de março de 2025 para oferecer suporte a essa configuração variável de ambiente. Veja a seguir detalhes das etapas de configuração.

Para configurar o Elastic Beanstalk para se autenticar em seu repositório privado com o Parameter Store AWS Systems Manager

> **Note**
>
> Você precisa configurar suas credenciais no AWS Systems Manager Parameter Store e também configurar as permissões necessárias do IAM para concluir essas etapas. Para obter mais informações, consulte Pré-requisitos para configurar segredos como variáveis de ambiente.

1. Crie sua estrutura de diretórios `.ebextensions` da seguinte forma.

```
### .ebextensions
#   ### env.config
### .platform
#   ### confighooks
#   #   ### prebuild
#   #       ### 01login.sh
#   ### hooks
```

```
#       ### prebuild
#           ### 01login.sh
### docker-compose.yml
```

2. Use o AWS Systems ManagerParameter Store para salvar as credenciais do seu repositório privado. Execute o seguinte put-parameter comando do AWS CLI Systems Manager para criá-los no Parameter Store.

```
aws ssm put-parameter --name USER --type String --value "username"
aws ssm put-parameter --name PASSWD --type String --value "passwd"
```

3. Crie o seguinte arquivo `env.config` e coloque-o no diretório `.ebextensions`, como mostrado na estrutura de diretórios anterior. Essa configuração usa o aws: elasticbeanstalk: aplicação: segredos do meio ambiente namespace para inicializar as variáveis de ambiente e do `PASSWD` Elastic `USER` Beanstalk com os valores armazenados no Systems Manager Parameter Store.

> **Note**
>
> `USER` e `PASSWD` no script devem corresponder às mesmas strings usadas nos comandos ssm put-parameter anteriores.

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    USER: arn:aws:ssm:us-east-1:111122223333:parameter/user
    PASSWD: arn:aws:secretsmanager:us-east-1:111122223333:passwd
```

4. Crie o seguinte arquivo de script `01login.sh` e coloque-o nos seguintes diretórios (também mostrado na estrutura de diretórios anterior):

   - `.platform/confighooks/prebuild`
   - `.platform/hooks/prebuild`

```
### example 01login.sh
#!/bin/bash
echo $PASSWD | docker login -u $USER --password-stdin
```

   Ele `01login.sh` faz referência às variáveis de ambiente do Elastic Beanstalk que você configurou na Etapa 3 para armazenar as credenciais do repositório e canaliza a senha

diretamente docker login para o comando no fluxo de entrada. `stdin` A `--password-stdin` opção usa o fluxo de entrada, impedindo que a senha persista no histórico do shell ou nos registros de auditoria. Para obter mais informações sobre autenticação com a interface de linha de comando do Docker, consulte docker login (login do docker) no site da documentação do Docker.

> **Observações**
>
> - Todos os arquivos de script devem ter permissão de execução. Use chmod +x para definir a permissão de execução em seus arquivos de hook. Em todas as versões da plataforma baseadas no Amazon Linux 2 que foram lançadas a partir de 29 de abril de 2022, o Elastic Beanstalk concede automaticamente permissões de execução para todos os scripts de hook da plataforma. Nesse caso, não é necessário conceder manualmente as permissões de execução. Para obter uma lista dessas versões da plataforma, consulte as notas de versão 29 de abril de 2022 - Plataforma Linux no Guia de notas de versão do AWS Elastic Beanstalk .
> - Os arquivos de hook podem ser arquivos binários ou arquivos de script que começam com uma linha #! contendo seu caminho de interpretador, como #!/bin/bash.
> - Para obter mais informações, consulte the section called “Hooks de plataforma” em Estender as plataformas do Elastic Beanstalk Linux.

Depois que o Elastic Beanstalk puder autenticar com o registro online que hospeda o repositório privado, as imagens poderão ser implantadas e extraídas.

Usar o arquivo **`Dockerrun.aws.json`**

Esta seção descreve outra abordagem para autenticar o Elastic Beanstalk em um repositório privado. Com essa abordagem, gere um arquivo de autenticação com o comando do Docker e, depois, carregue o arquivo de autenticação em um bucket do Amazon S3. Você também deve incluir as informações do bucket no arquivo `Dockerrun.aws.json`.

Como gerar e fornecer um arquivo de autenticação ao Elastic Beanstalk

1. Gere um arquivo de autenticação com o comando docker login. Para repositórios no Docker Hub, execute docker login:

```
$ docker login
```

Para outros registros, inclua o URL do servidor de registro:

```
$ docker login registry-server-url
```

> **Note**
>
> Se o ambiente do Elastic Beanstalk usar a versão da plataforma Docker da AMI do Amazon Linux (que precede o Amazon Linux 2), leia as informações relevantes em the section called “Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)”.

Para obter mais informações sobre o arquivo de autenticação, consulte Store images on Docker Hub e docker login no site do Docker.

2. Faça upload de uma cópia do arquivo de autenticação chamado `.dockercfg` em um bucket do Amazon S3 seguro.

   - O bucket do Amazon S3 deve ser hospedado no Região da AWS mesmo ambiente que o está usando. O Elastic Beanstalk não pode fazer download de arquivos de um bucket do Amazon S3 hospedado em outras regiões.
   - Conceda permissões para a operação `s3:GetObject` à função do IAM no perfil da instância. Para obter mais informações, consulte Gerenciar perfis de instância do Elastic Beanstalk.

3. Inclua as informações do bucket do Amazon S3 no parâmetro `Authentication` no arquivo `Dockerrun.aws.json`.

   O exemplo a seguir mostra o uso de um arquivo de autenticação chamado `mydockercfg` em um bucket chamado `amzn-s3-demo-bucket` para usar uma imagem privada em um registro de terceiros. Para obter o número de versão correto para o `AWSEBDockerrunVersion`, consulte a observação que segue o exemplo.

```
{
  "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"
}
```

> **Versões do `Dockerrun.aws.json`**
>
> O parâmetro `AWSEBDockerrunVersion` indica a versão do arquivo `Dockerrun.aws.json`.
>
> - As plataformas Docker AL2 e AL2 023 usam as seguintes versões do arquivo.
>   - `Dockerrun.aws.json v3`: ambientes que usam o Docker Compose.
>   - `Dockerrun.aws.json v1`: ambientes que não usam o Docker Compose.
> - O ECS executado no Amazon Linux 2 e o ECS executado no AL2 023 usam o arquivo. `Dockerrun.aws.json v2` A plataforma descontinuada ECS-The Multicontainer Docker Amazon Linux AMI (AL1) também usou essa mesma versão.

Depois que o Elastic Beanstalk puder autenticar com o registro online que hospeda o repositório privado, as imagens poderão ser implantadas e extraídas.

Configuração de ambientes do Docker do Elastic Beanstalk

Este capítulo explica informações adicionais de configuração para todas as ramificações da plataforma Docker compatíveis, incluindo a ramificação da plataforma Docker gerenciada pelo ECS. A menos que uma ramificação de plataforma específica ou componente de ramificação de plataforma seja identificado em uma seção, ela se aplica a todos os ambientes que executam plataformas do Docker compatíveis e plataformas do Docker gerenciadas pelo ECS.

> **Note**
>
> Se o ambiente do Elastic Beanstalk usar uma versão da plataforma do Amazon Linux AMI Docker (que precede o Amazon Linux 2), leia as informações adicionais no the section called "Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)".

Seções

- Configurar o software em ambientes do Docker
- Fazer referência a variáveis de ambiente em contêineres
- Usar o recurso de interpolação para variáveis de ambiente com o Docker Compose
- Geração de logs para relatórios de integridade aprimorados com o Docker Compose
- Registro em log personalizado de contêineres do Docker com o Docker Compose
- Imagens de Docker
- Configurar atualizações gerenciadas para ambientes do Docker
- Namespaces de configuração do Docker
- Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)

Configurar o software em ambientes do Docker

É possível usar o console do Elastic Beanstalk para configurar o software em execução nas instâncias do ambiente.

Como configurar o ambiente do Docker no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Faça as alterações necessárias na configuração.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Para obter informações sobre como definir configurações de software em qualquer ambiente, consulte the section called “Variáveis de ambiente e configurações de software”. As seções a seguir abordam informações específicas do Docker.

Opções de contêiner

A seção Container options (Opções de contêiner) tem opções específicas da plataforma. Para ambientes do Docker, isso permite que você escolha se o ambiente incluirá ou não o servidor de proxy Nginx.

Ambientes com Docker Compose

Se gerenciar seu ambiente do Docker com o Docker Compose, o Elastic Beanstalk assumirá que você executa um servidor de proxy como um contêiner. Portanto, ele usa como padrão None (Nenhum) para a configuração Proxy server (Servidor de proxy) e o Elastic Beanstalk não fornece uma configuração NGINX.

> **Note**
>
> Mesmo que você selecione NGINX como servidor de proxy, essa configuração será ignorada em um ambiente com Docker Compose. A configuração Proxy server (Servidor de proxy) ainda é padrão para None (Nenhum).

Como o proxy do servidor Web NGINX está desabilitado para a plataforma Docker no Amazon Linux 2 com o Docker Compose, é necessário seguir as instruções para gerar logs para relatórios de integridade aprimorada. Para obter mais informações, consulte Geração de logs para relatórios de integridade aprimorados com o Docker Compose.

Propriedades do ambiente (variáveis de ambiente)

Você pode usar propriedades de ambiente (também conhecidas como variáveis de ambiente) para passar valores, como endpoints, configurações de depuração e outras informações para seu

aplicativo. A seção Variáveis de ambiente do console permite que você especifique variáveis de ambiente nas EC2 instâncias que estão executando seu aplicativo. As variáveis de ambiente são passadas como pares de valores-chave para o aplicativo.

O código do aplicativo em execução em um contêiner pode se referir à variável de um ambiente pelo nome e ler seu valor. O código-fonte que lê essas variáveis de ambiente variará de acordo com a linguagem de programação. É possível encontrar instruções para ler valores de variáveis de ambiente nas linguagens de programação compatíveis com as plataformas gerenciadas do Elastic Beanstalk no respectivo tópico de plataforma. Para obter uma lista de links para esses tópicos, consulte the section called “Variáveis de ambiente e configurações de software”.

Segredos e parâmetros nas variáveis de ambiente do Elastic Beanstalk

O Elastic Beanstalk oferece a capacidade de AWS Secrets Manager AWS Systems Manager referenciar e armazenar dados de parâmetros em variáveis de ambiente. Essa é uma opção segura para que seu aplicativo acesse nativamente segredos e parâmetros armazenados por esses serviços sem precisar gerenciar chamadas de API para eles. Suas plataformas Docker gerenciadas pelo Elastic Beanstalk e pelo ECS devem ser uma versão lançada em ou após 26 de março de 2025 para oferecer suporte a esse recurso. Para obter mais informações sobre o uso de variáveis de ambiente para referenciar segredos, consulteBuscando segredos e parâmetros para variáveis de ambiente do Elastic Beanstalk.

Ambientes com Docker Compose

Se você gerenciar seu ambiente do Docker com o Docker Compose, deverá fazer alguma configuração adicional para recuperar as variáveis de ambiente nos contêineres. Para que os executáveis em execução no contêiner acessem essas variáveis de ambiente, será necessário fazer referência a eles no `docker-compose.yml`. Para mais informações, consulte Fazer referência a variáveis de ambiente em contêineres.

Fazer referência a variáveis de ambiente em contêineres

Se você estiver usando a ferramenta Docker Compose na plataforma Docker do Amazon Linux 2, o Elastic Beanstalk gerará um arquivo de ambiente do Docker Compose chamado `.env` no diretório raiz do projeto de aplicação. Esse arquivo armazena as variáveis de ambiente configuradas para Elastic Beanstalk.

> **Note**
>
> Se você incluir um arquivo `.env` no pacote de aplicações, o Elastic Beanstalk não gerará um arquivo `.env`.

Para que um contêiner faça referência às variáveis de ambiente definidas no Elastic Beanstalk, é necessário seguir uma ou ambas as abordagens de configuração.

- Adicione o arquivo `.env` gerado pelo Elastic Beanstalk à opção de configuração `env_file` no arquivo `docker-compose.yml`.
- Defina as variáveis de ambiente diretamente no arquivo `docker-compose.yml`.

Os arquivos a seguir fornecem um exemplo. O arquivo demonstrativo `docker-compose.yml` mostra as duas abordagens.

- Se você definir as propriedades de ambiente `DEBUG_LEVEL=1` e `LOG_LEVEL=error`, o Elastic beanstalk gerará o seguinte arquivo `.env`:

```
DEBUG_LEVEL=1
LOG_LEVEL=error
```

- Neste arquivo `docker-compose.yml`, a opção de configuração `env_file` aponta para o arquivo `.env` e também define a variável de ambiente `DEBUG=1` diretamente no arquivo `docker-compose.yml`.

```
services:
  web:
    build: .
    environment:
      - DEBUG=1
    env_file:
      - .env
```

Observações

- Se você definir a mesma variável de ambiente nos dois arquivos, a variável definida no arquivo `docker-compose.yml` terá precedência maior do que a variável definida no arquivo `.env`.
- Tenha cuidado para não deixar espaços entre o sinal de igual (=) e o valor atribuído à sua variável para evitar que espaços sejam adicionados à string.

Para saber mais sobre variáveis de ambiente no Docker Compose, consulte Variáveis de ambiente no Compose

Usar o recurso de interpolação para variáveis de ambiente com o Docker Compose

A partir de 28 de julho de 2023, data de lançamento da plataforma, a ramificação da plataforma Docker Amazon Linux 2 oferece o recurso de interpolação Docker Compose. Com esse recurso, os valores em um arquivo do Compose podem ser definidos por variáveis e interpolados em runtime. Para obter mais informações sobre esse recurso, consulte Interpolação no site de documentação do Docker.

Important

Se desejar usar esse atributo, com suas aplicações esteja ciente de que precisará implementar uma abordagem que use hooks de plataforma.
Isso é necessário devido a uma mitigação que implementamos no mecanismo da plataforma. Essa mitigação garante compatibilidade com as versões anteriores para clientes que não conhecem o novo atributo de interpolação e têm aplicações existentes que usam variáveis de ambiente com o caractere $. O mecanismo de plataforma atualizado escapa da interpolação por padrão substituindo o caractere $ por caracteres $$.

Veja a seguir um exemplo de um script de hook de plataforma que é possível configurar para permitir o uso do recurso de interpolação.

```
#!/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}"
```

Coloque os hooks da plataforma em ambos os diretórios:

- `.platform/confighooks/predeploy/`
- `.platform/hooks/predeploy/`

Para obter mais informações, consulte Hooks de plataforma no tópico Estender plataformas Linux deste guia.

Geração de logs para relatórios de integridade aprimorados com o Docker Compose

O agente de integridade do Elastic Beanstalk fornece métricas de integridade do sistema operacional e da aplicação para ambientes do Elastic Beanstalk. Ele se baseia em formatos de log do servidor da Web que retransmitem informações em um formato específico.

O Elastic Beanstalk pressupõe que você execute um proxy de servidor da Web como um contêiner. Como resultado, o proxy do servidor da Web NGINX é desativado para ambientes do Docker que

executam o Docker Compose. É necessário configurar o servidor para gravar logs no local e no formato usados pelo agente de integridade do Elastic Beanstalk. Isso permite que você faça uso completo dos relatórios de integridade aprimorados, mesmo que o proxy do servidor da Web esteja desabilitado.

Para obter instruções sobre como fazer isso, consulte Configuração do log de servidor web

Registro em log personalizado de contêineres do Docker com o Docker Compose

Para solucionar problemas de forma eficiente e monitorar os serviços em contêineres, é possível solicitar logs de instância do Elastic Beanstalk por meio do console de gerenciamento de ambiente ou da CLI do EB. Os logs de instância são compostos de logs de pacote e logs de cauda, combinados e empacotados para permitir que você visualize logs e eventos recentes de maneira eficiente e direta.

O Elastic Beanstalk cria diretórios de logs na instância de contêiner, um para cada serviço definido no arquivo `docker-compose.yml`, em `/var/log/eb-docker/containers/`*`<service name>`*. Se você estiver usando o recurso Docker Compose na plataforma Docker do Amazon Linux 2, poderá montar esses diretórios no local dentro da estrutura do arquivo de contêiner onde os logs são gravados. Quando você monta diretórios de logs para gravar dados de log, o Elastic Beanstalk pode coletar dados de log desses diretórios.

Se as aplicações estiverem em uma plataforma Docker que não esteja usando o Docker Compose, você poderá seguir o procedimento padrão descrito em Registro em log personalizado de contêineres do Docker com o Docker Compose.

Como configurar os arquivos de logs do seu serviço para serem arquivos de cauda e logs de pacote recuperáveis

1. Edite o arquivo `docker-compose.yml`.
2. Na chave `volumes` do seu serviço, adicione uma montagem de ligação deste modo:

   `"${EB_LOG_BASE_DIR}/`*`<service name>`*`:`*`<log directory inside container>`*

   No arquivo demonstrativo `docker-compose.yml` abaixo:

   - `nginx-proxy` é *`<service name>`*
   - `/var/log/nginx` é *`<log directory inside container>`*

```
services:
  nginx-proxy:
    image: "nginx"
    volumes:
      - "${EB_LOG_BASE_DIR}/nginx-proxy:/var/log/nginx"
```

- O diretório `var/log/nginx` contém os logs do serviço nginx-proxy no contêiner e será mapeado para o diretório `/var/log/eb-docker/containers/nginx-proxy` no host.
- Todos os logs nesse diretório agora são recuperáveis como logs finais e de pacote por meio da funcionalidade de solicitação de logs de instância do Elastic Beanstalk.

Observações

- $ {EB_LOG_BASE_DIR} é uma variável de ambiente definida pelo Elastic Beanstalk com o valor `/var/log/eb-docker/containers`.
- O Elastic Beanstalk cria automaticamente o diretório `/var/log/eb-docker/containers/<service name>` para cada serviço no arquivo `docker-compose.yml`.

Imagens de Docker

As ramificações da plataforma Docker e Docker gerenciado pelo ECS para o Elastic Beanstalk são compatíveis com o uso de imagens do Docker armazenadas em um repositório de imagens online público ou privado.

Especifique as imagens por nome em `Dockerrun.aws.json`. Observe as seguintes convenções:

- As imagens em repositórios oficiais no Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
- As imagens em outros repositórios no Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).

- As imagens em outros repositórios online são ainda mais qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu` ou *`account-id`*`.dkr.ecr.us-east-2.amazonaws.com/ubuntu:trusty`).

Para ambientes que usam apenas a plataforma Docker, também é possível criar sua própria imagem durante a criação do ambiente com um Dockerfile. Para mais detalhes, consulte Criar imagens personalizadas com um Dockerfile. A plataforma Docker gerenciada pelo ECS não oferece suporte a essa funcionalidade.

Configurar atualizações gerenciadas para ambientes do Docker

Com as atualizações gerenciadas de plataforma, é possível configurar o ambiente para ser atualizado automaticamente para a versão mais recente de uma plataforma em um agendamento.

No caso de ambientes do Docker, pode ser conveniente decidir se uma atualização automática de plataforma deverá acontecer entre versões do Docker, quando a nova versão da configuração da plataforma inclui uma nova versão do Docker. O Elastic Beanstalk é compatível com atualizações de plataforma gerenciada nas versões do Docker ao atualizar a partir de um ambiente que execute uma versão da plataforma Docker mais recente do que a 2.9.0. Quando uma nova versão de plataforma inclui uma nova versão do Docker, o Elastic Beanstalk incrementa o número da versão de atualização secundária. Portanto, para permitir as atualizações gerenciadas de plataforma entre versões do Docker, habilite essas atualizações para atualizações de versão secundária e de patch. Para evitar as atualizações gerenciadas de plataforma entre versões do Docker, habilite essas atualizações para aplicar somente atualizações de versão de patch.

Por exemplo, o seguinte arquivo de configuração habilita as atualizações gerenciadas de Plataforma às 9h UTC toda terça-feira para atualizações de versão secundária e de patch, permitindo as atualizações gerenciadas entre versões do Docker:

Example .ebextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: minor
```

Para ambientes que executam versões de plataforma do Docker 2.9.0 ou anterior, o Elastic Beanstalk nunca executará as atualizações gerenciadas de plataforma se a nova versão da configuração da plataforma incluir uma nova versão do Docker.

Namespaces de configuração do Docker

Você pode usar um arquivo de configuração para definir opções de configuração e executar outras tarefas de configuração de instância durante implantações. Opções de configuração podem ser específicas da plataforma ou aplicáveis a todas as plataformas no serviço Elastic Beanstalk como um todo. As opções de configuração são organizadas em namespaces.

> **Note**
>
> Essas informações só se aplicam ao ambiente do Docker que não esteja executando o Docker Compose. Essa opção tem um comportamento diferente com ambientes do Docker que executam o Docker Compose. Para obter mais informações sobre serviços proxy com o Docker Compose, consulte Opções de contêiner.

A plataforma Docker é compatível com as opções nos namespaces a seguir, além das opções compatíveis com todos os ambientes do Elastic Beanstalk:

- `aws:elasticbeanstalk:environment:proxy`: escolha o servidor de proxy para o ambiente. O Docker oferece suporte à execução com o servidor de proxy Nginx ou nenhum servidor.

O arquivo de configuração de exemplo a seguir configura um ambiente do Docker para não executar nenhum servidor de proxy.

Example .ebextensions/docker-settings.config

```
option_settings:
  aws:elasticbeanstalk:environment:proxy:
    ProxyServer: none
```

Configuração do Docker na AMI do Amazon Linux (que precede o Amazon Linux 2)

Se o ambiente Docker do Elastic Beanstalk usar uma versão da plataforma AMI do Amazon Linux (que precede o Amazon Linux 2), leia as informações adicionais nesta seção.

Usar um arquivo de autenticação para um repositório privado

Essas informações são relevantes se você estiver usando imagens de um repositório privado. A partir do Docker versão 1.7, o comando docker login alterou o nome do arquivo de autenticação e o formato do arquivo. As versões da plataforma Docker da AMI do Amazon Linux (que precede o Amazon Linux 2) exigem o arquivo de configuração de formato `~/.dockercfg` mais antigo.

Com o Docker versão 1.7 e posterior, o comando docker login cria o arquivo de autenticação em `~/.docker/config.json` no formato a seguir.

```
{
    "auths":{
      "server":{
        "auth":"key"
      }
    }
  }
```

Com o Docker versão 1.6.2 e anterior, o comando docker login cria o arquivo de autenticação em `~/.dockercfg` no formato a seguir.

```
{
    "server" :
    {
      "auth" : "auth_token",
      "email" : "email"
    }
  }
```

Para converter um arquivo `config.json`, remova a chave `auths` externa, adicione uma chave `email` e simplifique o documento JSON para que corresponda ao formato antigo.

Nas versões da plataforma Docker do Amazon Linux 2, o Elastic Beanstalk usa o nome e o formato mais recentes do arquivo de autenticação. Se você estiver usando uma versão da plataforma Docker do Amazon Linux 2, será possível usar o arquivo de autenticação criado pelo comando docker login sem nenhuma conversão.

Configurar volumes de armazenamento adicionais

Para melhorar o desempenho no Amazon Linux AMI, o Elastic Beanstalk configura dois volumes de armazenamento do Amazon EBS para as instâncias da Amazon do seu ambiente Docker. EC2 Além

do volume raiz provisionado para todos os ambientes do Elastic Beanstalk, um segundo volume de 12 GB chamado `xvdcz` é provisionado para armazenamento de imagens em ambientes do Docker.

Se você precisar de mais espaço de armazenamento ou de mais IOPS para imagens de Docker, pode personalizar o volume de armazenamento de imagens usando a opção de configuração `BlockDeviceMapping` no namespace aws:autoscaling:launchconfiguration.

Por exemplo, o seguinte arquivo de configuração aumenta o tamanho do volume de armazenamento para 100 GB com 500 IOPS provisionadas:

Example .ebextensions/blockdevice-xvdcz.config

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:100::io1:500
```

Se você usar a opção `BlockDeviceMappings` para configurar volumes adicionais para seu aplicativo, inclua um mapeamento para `xvdcz` a fim de garantir que ele seja criado. O exemplo a seguir configura dois volumes, o volume de armazenamento de imagens `xvdcz` com as configurações padrão e um volume de aplicativo adicional de 24 GB chamado `sdh`:

Example .ebextensions/blockdevice-sdh.config

```
option_settings:
  aws:autoscaling:launchconfiguration:
    BlockDeviceMappings: /dev/xvdcz=:12:true:gp2,/dev/sdh=:24
```

> **Note**
>
> Quando você altera as configurações nesse namespace, o Elastic Beanstalk substitui todas as instâncias no ambiente por instâncias que executam a nova configuração. Para mais detalhes, consulte Alterações de configuração.

Plataformas herdadas

Este capítulo lista o conteúdo relacionado às plataformas Docker anteriores que não são mais suportadas pelo AWS Elastic Beanstalk. Os tópicos listados aqui permanecem neste documento

como referência para todos os clientes que usaram esses recursos ou componentes antes de sua retirada.

Tópicos

- Migração para o Elastic Beanstalk Docker em execução no Amazon Linux 2 a partir do Docker de vários contêineres em execução no Amazon Linux
- GlassFish Contêineres Docker pré-configurados no Elastic Beanstalk

Migração para o Elastic Beanstalk Docker em execução no Amazon Linux 2 a partir do Docker de vários contêineres em execução no Amazon Linux

Antes do lançamento da ramificação da plataforma ECS em execução no Amazon Linux 2 de 64 bits, o Elastic Beanstalk oferecia um caminho alternativo de migração para o Amazon Linux 2 para clientes com ambientes baseados no Docker de vários contêineres em execução no Amazon Linux de 64 bits. Este tópico descreve esse caminho de migração e foi mantido neste documento como referência para todos os clientes que concluíram esse caminho de migração.

Agora recomendamos que os clientes com ambientes baseados na ramificação da plataforma Docker de vários contêineres em execução no Amazon Linux de 64 bits migrem para a ramificação da plataforma ECS em execução no Amazon Linux 2 de 64 bits. Ao contrário do caminho alternativo de migração, essa abordagem continua a usar o Amazon ECS para coordenar implantações de contêineres em ambientes do Docker gerenciado pelo ECS. Esse aspecto permite uma abordagem mais direta. Não é necessária nenhuma alteração no código-fonte e o mesmo arquivo `Dockerrun.aws.json` v2 é compatível. Para obter mais informações, consulte Migrando seu aplicativo Elastic Beanstalk do Docker de vários contêineres gerenciado pelo ECS para o ECS no Amazon Linux 2023 AL1 .

(Herdado) Migração do Docker de vários contêineres no Amazon Linux para a ramificação da plataforma Docker no Amazon Linux 2

É possível migrar as aplicações em execução na plataforma Docker de vários contêineres na AMI do Amazon Linux para a plataforma do Amazon Linux 2 Docker. A plataforma Docker de vários contêineres na AMI do Amazon Linux requer que você especifique imagens de aplicação pré-criadas para serem executadas como contêineres. Após a migração, você não terá mais essa limitação, pois a plataforma Docker do Amazon Linux 2 também permite ao Elastic Beanstalk criar imagens de contêiner durante a implantação. As aplicações continuarão sendo executadas em ambientes de vários contêineres com os benefícios adicionais da ferramenta Docker Compose.

O Docker Compose é uma ferramenta para definição e execução de aplicações do Docker de vários contêineres. Para saber mais sobre o Docker Compose e como instalá-lo, consulte Visão geral do Docker Compose e Instalar o Docker Compose no site do Docker.

O arquivo **`docker-compose.yml`**

A ferramenta Docker Compose usa o arquivo `docker-compose.yml` para configuração dos serviços de aplicações. Esse arquivo substitui o arquivo `Dockerrun.aws.json v2` no diretório de projetos de aplicações e no pacote de origem da aplicação. O arquivo `docker-compose.yml` é criado manualmente. Consulte o arquivo `Dockerrun.aws.json v2` para a maioria dos valores de parâmetro.

Veja abaixo um arquivo demonstrativo `docker-compose.yml` e o arquivo `Dockerrun.aws.json v2` correspondente para a mesma aplicação. Para obter mais informações sobre o arquivo `docker-compose.yml`, consulte Referência de arquivo do Compose. Para obter mais informações sobre o arquivo `Dockerrun.aws.json v2`, consulte `Dockerrun.aws.json v2`.

`docker-compose.yml`

```
version: '2.4'
services:
  php-app:
    image: "php:fpm"
    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
"
```

`Dockerrun.aws.json v2`

```
{
  "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": [
    {
```

docker-compose.yml

```
      - "./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",
      "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
        },
```

docker-compose.yml	Dockerrun.aws.json v2

```
        {
          "sourceVolume": "nginx-pr
oxy-conf",
          "containerPath": "/etc/ngi
nx/conf.d",
          "readOnly": true
        },
        {
          "sourceVolume": "awseb-lo
gs-nginx-proxy",
          "containerPath": "/var/log
/nginx"
        }
      ]
    }
  ]
}
```

Considerações adicionais sobre migração

A plataforma Docker Amazon Linux 2 e a plataforma Multi-container Docker Amazon Linux AMI implementam as propriedades do ambiente de forma diferente. Essas duas plataformas também têm diretórios de log diferentes que o Elastic Beanstalk cria para cada um de seus contêineres. Depois de migrar da plataforma do Amazon Linux AMI Multi-Container Docker, você precisará estar ciente dessas diferentes implementações para seu novo ambiente de plataforma do Amazon Linux 2 Docker.

Área	Plataforma Docker no Amazon Linux 2 com Docker Compose	Plataforma Docker de vários contêiner es na AMI do Amazon Linux
Proprieda des de ambiente	Para que seus contêineres acessem as propriedades do ambiente, é necessário adicionar uma referência ao arquivo `.env` no arquivo `docker-compose.yml` . O Elastic Beanstalk gera o arquivo `.env`, listando cada uma	O Elastic Beanstalk pode transmitir as propriedades do ambiente diretamen te para o contêiner. Seu código em execução no contêiner pode acessar essas propriedades como variáveis de

Área	Plataforma Docker no Amazon Linux 2 com Docker Compose	Plataforma Docker de vários contêiner es na AMI do Amazon Linux
	das propriedades como variáveis de ambiente. Para obter mais informações, consulte Fazer referência a variáveis de ambiente em contêineres.	ambiente sem nenhuma configuração adicional.
Diretórios de log	Para cada contêiner, o Elastic Beanstalk cria um diretório de logs chamado `/var/log/eb-docker/containers/` *`<service name>`* (ou `${EB_LOG_BASE_DIR}/<service name>`). Para obter mais informações, consulte Registro em log personalizado de contêineres do Docker com o Docker Compose.	Para cada contêiner, o Elastic Beanstalk cria um diretório de logs chamado `/var/log/containers/` *`<containername>`* . Para obter mais informações, consulte o campo `mountPoints` no Formato de definição de contêiner.

Etapas da migração

Como migrar para a plataforma Docker do Amazon Linux 2

1. Crie o arquivo `docker-compose.yml` para a aplicação, com base no arquivo `Dockerrun.aws.json` v2 existente. Para obter mais informações, consulte a seção acim O arquivo `docker-compose.yml`.
2. No diretório raiz da pasta do projeto da aplicação, substitua o arquivo `Dockerrun.aws.json` v2 pelo `docker-compose.yml` que você acabou de criar.

   A estrutura do diretório deve ser a seguinte.

   ```
   ~/myApplication
   |-- docker-compose.yml
   |-- .ebextensions
   |-- php-app
   |-- proxy
   ```

3. Use o comando eb init para configurar o diretório local para implantação no Elastic Beanstalk.

```
~/myApplication$ eb init -p docker application-name
```

4. Use o comando eb create para criar um ambiente e implante sua imagem do Docker.

```
~/myApplication$ eb create environment-name
```

5. Se for uma aplicação Web, depois que o ambiente for iniciado, use o comando eb open para exibi-la em um navegador da Web.

```
~/myApplication$ eb open environment-name
```

6. Você pode exibir o status do ambiente recém-criado usando o comando eb status.

```
~/myApplication$ eb status environment-name
```

GlassFish Contêineres Docker pré-configurados no Elastic Beanstalk

> **Note**
>
> Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

A ramificação pré-configurada da GlassFish plataforma Docker que é executada no Amazon Linux AMI (AL1) não é mais suportada. Para migrar seu GlassFish aplicativo para uma plataforma Amazon Linux 2023 compatível, implante GlassFish e seu código de aplicativo em uma imagem Docker do Amazon Linux 2023. Para obter mais informações, consulte o tópico a seguir, the section called “Tutorial - GlassFish no Docker: caminho para o Amazon Linux 2023”.

Primeiros passos com contêineres do Docker pré-configurados no Amazon Linux AMI (anterior ao Amazon Linux 2)

Esta seção mostra como desenvolver uma aplicação demonstrativa localmente e implantá-la no Elastic Beanstalk com um contêiner pré-configurado do Docker.

Configurar seu ambiente de desenvolvimento local

Para este passo a passo, usamos um aplicativo de GlassFish exemplo.

Para configurar seu ambiente

1. Crie uma pasta para o aplicativo de exemplo.

```
~$ mkdir eb-preconf-example
~$ cd eb-preconf-example
```

2. Faça o download do código de aplicativo de exemplo para a nova pasta.

```
~$ wget https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/docker-
glassfish-v1.zip
~$ unzip docker-glassfish-v1.zip
~$ rm docker-glassfish-v1.zip
```

Desenvolver e testar localmente

Para desenvolver um GlassFish aplicativo de exemplo

1. Adicione um `Dockerfile` à pasta raiz do seu aplicativo. No arquivo, especifique a imagem base do AWS Elastic Beanstalk Docker a ser usada para executar seu contêiner Docker pré-configurado local. Posteriormente, você implantará seu aplicativo em uma versão pré-configurada da plataforma Docker do Elastic GlassFish Beanstalk. Escolha a imagem de base do Docker que essa versão da plataforma usa. Para descobrir a imagem do Docker atual da versão da plataforma, consulte a seção Docker pré-configurado na página Plataformas compatíveis com o AWS Elastic Beanstalk no guia de Plataformas do AWS Elastic Beanstalk .

   Example ~/E/Dockerfile b-preconf-example

```
# For Glassfish 5.0 Java 8
FROM amazon/aws-eb-glassfish:5.0-al-onbuild-2.11.1
```

   Para obter mais informações sobre como usar um `Dockerfile`, consulte Preparar a imagem do Docker para implantação no Elastic Beanstalk.

2. Crie a imagem do Docker.

```
~/eb-preconf-example$ docker build -t my-app-image .
```

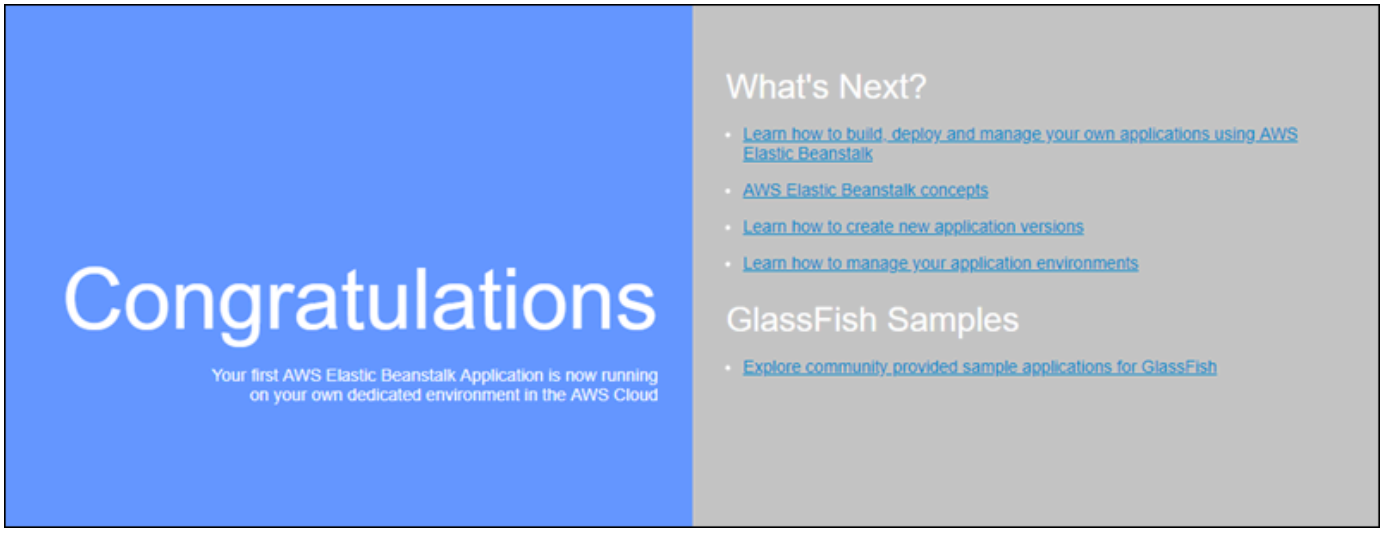
3. Execute o contêiner Docker da imagem.

> **Note**
>
> Você deve incluir o sinalizador `-p` para mapear a porta 8080 no contêiner para a porta 3000 do localhost. Os contêineres do Docker do Elastic Beanstalk sempre expõem a aplicação na porta 8080 no contêiner. O sinalizador `-it` executa a imagem como processo interativo. O sinalizador `--rm` limpa o sistema de arquivos do contêiner quando o contêiner é encerrado. Você tem a opção de incluir o sinalizador `-d` para executar a imagem como um daemon.

```
$ docker run -it --rm -p 3000:8080 my-app-image
```

4. Para visualizar o aplicativo de exemplo, digite o seguinte URL no seu navegador da web.

```
http://localhost:3000
```



Implantar no Elastic Beanstalk

Depois de testar a aplicação, você estará pronto para implantá-la no Elastic Beanstalk.

Como implantar a aplicação no Elastic Beanstalk

1. Na pasta raiz do seu aplicativo, renomeie o `Dockerfile` como `Dockerfile.local`. Essa etapa é necessária para que o Elastic Beanstalk `Dockerfile` use a que contém as instruções corretas para que o Elastic Beanstalk crie uma imagem personalizada do Docker em cada instância da Amazon em seu ambiente do Elastic Beanstalk. EC2

   > **Note**
   >
   > Não será necessário executar essa etapa se o `Dockerfile` incluir instruções que modifiquem a imagem de base do Docker da versão da plataforma. Não será necessário usar um `Dockerfile` se o seu `Dockerfile` incluir apenas uma linha `FROM` para especificar a imagem de base da qual criar o contêiner. Neste caso, o `Dockerfile` é redundante.

2. Crie um pacote de origem do aplicativo.

   ```
   ~/eb-preconf-example$ zip myapp.zip -r *
   ```

3. Abra o console do Elastic Beanstalk com este link pré-configurado: console.aws.amazon. com/elasticbeanstalk/home#/newApplication? Nome do aplicativo = Tutoriais e tipo de ambiente = LoadBalanced
4. Em Platform (Plataforma), em Preconfigured – Docker (Pré-configurado - Docker), escolha Glassfish.
5. Para Application code, escolha Upload your code e Upload.
6. Escolha Local file (Arquivo local), selecione Browse (Procurar) e abra o pacote de origem do aplicativo que você acabou de criar.
7. Escolha Carregar.
8. Selecione Review and launch.
9. Revise as configurações disponíveis e, em seguida, escolha Create app.
10. Quando o ambiente é criado, você pode visualizar o aplicativo implantado. Escolha o URL do ambiente exibido na parte superior do painel do console.

Implantação de um GlassFish aplicativo na plataforma Docker: um caminho de migração para o Amazon Linux 2023

O objetivo deste tutorial é fornecer aos clientes que usam a GlassFish plataforma Docker pré-configurada (baseada no Amazon Linux AMI) um caminho de migração para o Amazon Linux 2023. Você pode migrar seu GlassFish aplicativo para o Amazon Linux 2023 implantando seu código GlassFish de aplicativo em uma imagem Docker do Amazon Linux 2023.

O tutorial explica como usar a plataforma AWS Elastic Beanstalk Docker para implantar um aplicativo baseado no servidor de aplicativos Java EE em um ambiente do GlassFish Elastic Beanstalk.

Demonstramos duas abordagens para criar uma imagem do Docker:

- Simples — forneça o código-fonte do seu GlassFish aplicativo e deixe o Elastic Beanstalk criar e executar uma imagem do Docker como parte do provisionamento do seu ambiente. Isso é fácil de configurar e tem um custo de maior tempo de provisionamento de instâncias.
- Avançado: crie uma imagem do Docker personalizada que tenha o código da aplicação e as dependências e forneça-a para o Elastic Beanstalk para uso no ambiente. Essa abordagem é um pouco mais envolvida e diminui o tempo de provisionamento de instâncias em seu ambiente.

Pré-requisitos

Este tutorial pressupõe que você tenha algum conhecimento das operações básicas do Elastic Beanstalk, da Elastic Beanstalk Command Line Interface (EB CLI) e do Docker. Caso ainda não tenha, siga as instruções em Saiba como começar a usar o Elastic Beanstalk para iniciar seu primeiro ambiente do Elastic Beanstalk. Este tutorial usa a EB CLI, mas também é possível criar ambientes e fazer upload de aplicações usando o console do Elastic Beanstalk.

Para concluir este tutorial, você precisará fazer o seguinte:

- Uma instalação local de trabalho do Docker. Para obter mais informações, consulte Obter o Docker no site de documentação do Docker.
- Acesso ao Docker Hub. Você precisará criar um ID do Docker para acessar o Docker Hub. Para obter mais informações, consulte Compartilhar a aplicação no site de documentação do Docker.

Para saber mais sobre como configurar ambientes do Docker em plataformas do Elastic Beanstalk, consulte Preparar a imagem do Docker para implantação no Elastic Beanstalk neste mesmo capítulo.

Exemplo simples: fornecer o código do aplicativo

Essa é uma maneira fácil de implantar seu GlassFish aplicativo. Forneça o código-fonte da aplicação juntamente com o `Dockerfile` incluído neste tutorial. O Elastic Beanstalk cria uma imagem do Docker que inclui seu aplicativo e a pilha de software. GlassFish Depois, o Elastic Beanstalk executa a imagem nas instâncias do ambiente.

Um problema com essa abordagem é que o Elastic Beanstalk cria a imagem do Docker localmente sempre que ela cria uma instância para o ambiente. A criação da imagem aumenta o tempo de provisionamento de instâncias. Esse impacto não se limita à criação inicial do ambiente. Ele também ocorre durante ações de expansão.

Para iniciar um ambiente com um GlassFish aplicativo de exemplo

1. Faça download do exemplo `docker-glassfish-al2-v1.zip` e expanda o arquivo `.zip` para um diretório em seu ambiente de desenvolvimento.

   ```
   ~$ 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
   ```

   A estrutura do diretório deve ser a seguinte.

   ```
   ~/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
   ```

   Os arquivos a seguir são fundamentais para criar e executar um contêiner do Docker em seu ambiente:

- `Dockerfile`: fornece instruções que o Docker usa para criar uma imagem com a aplicação e as dependências necessárias.
- `glassfish-start.sh`: um script shell que a imagem do Docker executa para iniciar a aplicação.
- `Dockerrun.aws.json`— Fornece uma chave de registro, para incluir o log do servidor de GlassFish aplicativos nas solicitações do arquivo de log. Se você não estiver interessado em GlassFish registros, você pode omitir esse arquivo.

2. Configure o diretório local para implantação no Elastic Beanstalk.

```
~/glassfish-example$ eb init -p docker glassfish-example
```

3. (Opcional) Use o comando eb local run para criar e executar seu contêiner localmente.

```
~/glassfish-example$ eb local run --port 8080
```

> **Note**
>
> Para saber mais sobre o comando eb local, consulte the section called “eb local”. O comando não é compatível no Windows. Como alternativa, você pode criar e executar seu contêiner com os comandos docker build e docker run. Para obter mais informações, consulte a Documentação do Docker.

4. (Opcional) Embora o contêiner esteja em execução, use o comando eb local open para visualizar o aplicativo em um navegador da web. Opcionalmente, abra http://localhost:8080/ em um navegador da Web.

```
~/glassfish-example$ eb local open
```

5. Use o comando eb create para criar um ambiente e implante seu aplicativo.

```
~/glassfish-example$ eb create glassfish-example-env
```

6. Depois que o ambiente for iniciado, use o comando eb open para visualizá-lo em um navegador da web.

```
~/glassfish-example$ eb open
```

Quando terminar de trabalhar com o exemplo, encerre o ambiente e exclua os recursos relacionados.

```
~/glassfish-example$ eb terminate --all
```

Exemplo avançado: fornecer uma imagem do Docker pré-compilada

Essa é uma forma mais avançada de implantar seu GlassFish aplicativo. Com base no primeiro exemplo, você cria uma imagem do Docker contendo o código do aplicativo e a pilha de GlassFish software e a envia para o Docker Hub. Depois de realizar essa etapa única, é possível iniciar ambientes do Elastic Beanstalk com base na imagem personalizada.

Ao iniciar um ambiente e fornecer sua imagem do Docker, as instâncias no seu ambiente fazem download e usam essa imagem diretamente e não precisam criar uma imagem do Docker. Portanto, o tempo de provisionamento de instâncias é reduzido.

> **Observações**
>
> - As etapas a seguir criam uma imagem do Docker disponível publicamente.
> - Você usará comandos do Docker da instalação local do Docker, juntamente com as credenciais do Docker Hub. Para obter mais informações, consulte a seção Pré-requisitos deste tópico.

Para iniciar um ambiente com uma imagem Docker de GlassFish aplicativo pré-criada

1. Faça download e expanda o exemplo `docker-glassfish-al2-v1.zip` como no exemplo simples anterior. Se concluiu esse exemplo, é possível usar o diretório que você já tem.
2. Crie uma imagem do Docker e envie-a para o Docker Hub. Insira seu ID do Docker para *`docker-id`* fazer login no Docker Hub.

   ```
   ~/glassfish-example$ docker build -t docker-id/beanstalk-glassfish-example:latest .
   ~/glassfish-example$ docker push docker-id/beanstalk-glassfish-example:latest
   ```

> **Note**
>
> Antes de enviar sua imagem, talvez seja necessário executar docker login. Suas credenciais do Docker Hub serão solicitadas se você executar o comando sem parâmetros.

3. Crie um diretório adicional.

```
~$ mkdir glassfish-prebuilt
~$ cd glassfish-prebuilt
```

4. Copie o exemplo a seguir em um arquivo chamado `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. Configure o diretório local para implantação no Elastic Beanstalk.

```
~/glassfish-prebuilt$ eb init -p docker glassfish-prebuilt$
```

6. (Opcional) Use o comando eb local run para executar o contêiner localmente.

```
~/glassfish-prebuilt$ eb local run --port 8080
```

7. (Opcional) Embora o contêiner esteja em execução, use o comando eb local open para visualizar o aplicativo em um navegador da web. Opcionalmente, abra http://localhost:8080/ em um navegador da Web.

```
~/glassfish-prebuilt$ eb local open
```

8. Use o comando eb create para criar um ambiente e implante sua imagem do Docker.

```
~/glassfish-prebuilt$ eb create glassfish-prebuilt-env
```

9. Depois que o ambiente for iniciado, use o comando eb open para visualizá-lo em um navegador da web.

```
~/glassfish-prebuilt$ eb open
```

Quando terminar de trabalhar com o exemplo, encerre o ambiente e exclua os recursos relacionados.

```
~/glassfish-prebuilt$ eb terminate --all
```

Ambientes de monitoramento no Elastic Beanstalk

Com o monitoramento da integridade do Elastic Beanstalk, você pode verificar a disponibilidade do aplicativo e criar alertas que são ativados quando as métricas excedem seus limites. Você pode usar o monitoramento de saúde do Elastic Beanstalk no console e na linha de comando para monitorar o status do seu ambiente.

Tópicos

- Monitorando a integridade do ambiente no console AWS de gerenciamento
- Usar a EB CLI para monitorar a integridade do ambiente
- Relatórios de integridade básica
- Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk
- Gerenciar alarmes
- Visualizar o histórico de alterações de um ambiente do Elastic Beanstalk
- Visualizar o stream de eventos de um ambiente do Elastic Beanstalk
- Listar e conectar com instâncias de servidor
- Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk

Monitorando a integridade do ambiente no console AWS de gerenciamento

É possível acessar informações operacionais sobre sua aplicação no console do Elastic Beanstalk. O console exibe o status do ambiente e a integridade do aplicativo rapidamente. Na página Environments (Ambientes) do console e na página de cada aplicativo, os ambientes na lista são codificados por cores para indicar o status.

Como monitorar um ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Monitoring (Monitoramento).

A página Monitoramento mostra as estatísticas gerais do ambiente, como a latência média e a utilização de CPU. Além das estatísticas gerais, é possível visualizar gráficos de monitoramento que mostram o uso de recursos ao longo do tempo. Você pode clicar em qualquer um dos gráficos para visualizar informações mais detalhadas.

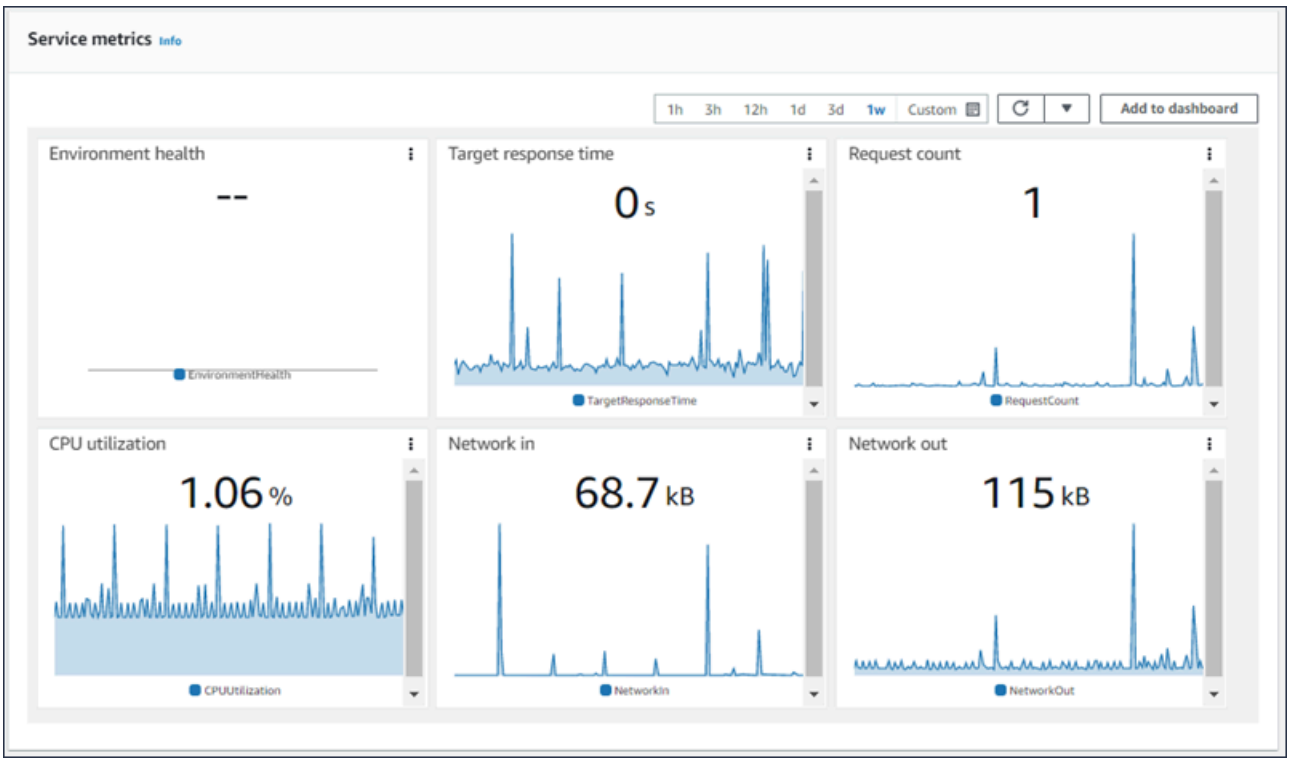
> **Note**
>
> Por padrão, somente CloudWatch métricas básicas são ativadas, que retornam dados em períodos de cinco minutos. Você pode ativar CloudWatch métricas mais granulares de um minuto editando as configurações do seu ambiente.

Grafos de monitoramento

A página Monitoring (Monitoramento) exibe uma visão geral das métricas relacionadas à integridade do seu ambiente. Isso inclui o conjunto padrão de métricas fornecido pelo Elastic Load Balancing e pela Amazon EC2, além de gráficos que mostram como a saúde do ambiente mudou ao longo do tempo.

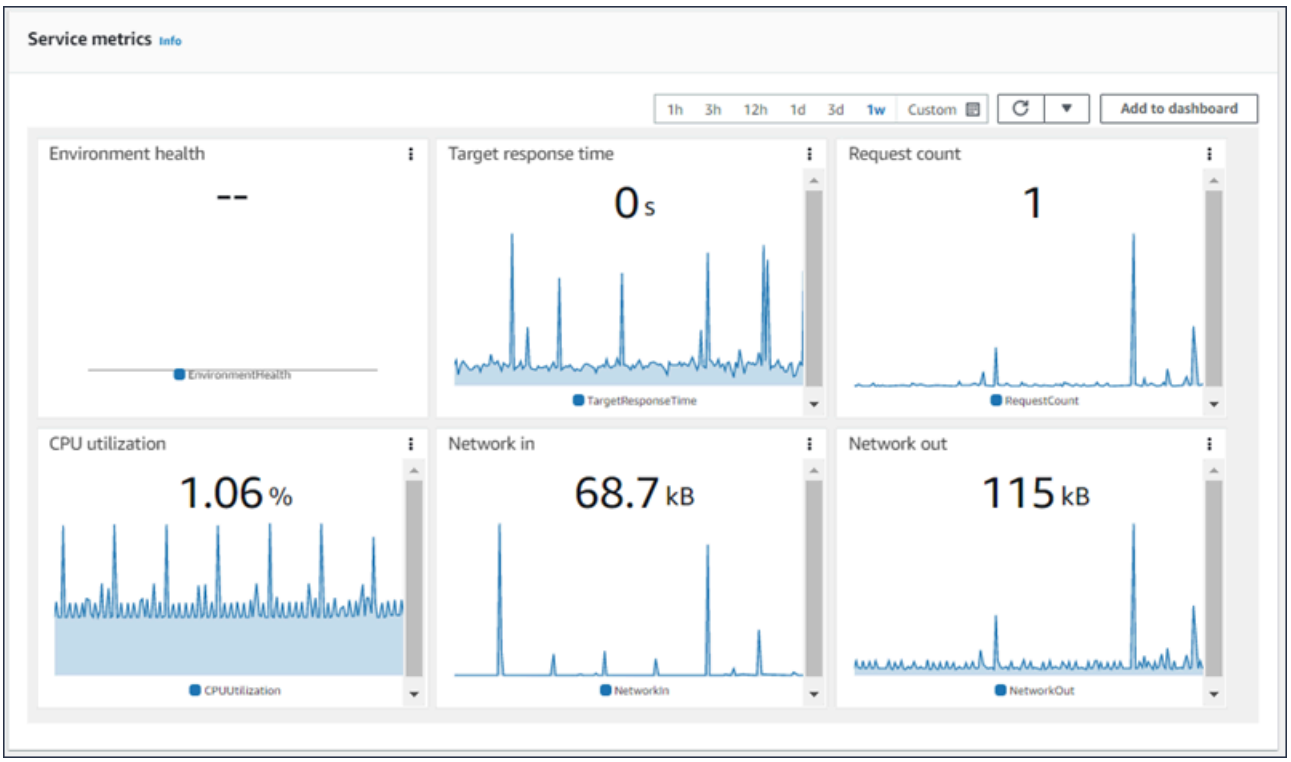
A barra acima dos gráficos fornece uma variedade de intervalos de tempo para você selecionar. Por exemplo, selecione 1w para exibir informações que abrangem a última semana. Ou selecione 3h para exibir informações que abrangem as últimas três horas.

Para uma maior variedade de seleções de intervalo de tempo, escolha Custom (Personalizar). A partir daqui, você tem duas opções de intervalo: Absoluto ou Relativo. A opção Absolute permite que você especifique um intervalo de datas específico, como 1º de janeiro de 2023 a 30 de junho de 2023. A opção Relativo permite selecionar um número inteiro com uma unidade de tempo específica: minutos, horas, dias, semanas ou meses. Os exemplos incluem 10 horas, 10 dias e 10 meses.



Personalizar o console de monitoramento

Para criar e visualizar métricas personalizadas, você deve usar a Amazon CloudWatch. Com CloudWatch você pode criar painéis personalizados para monitorar seus recursos em uma única visualização. Selecione Adicionar ao painel para navegar até o CloudWatch console da Amazon a partir da página de monitoramento. CloudWatch A Amazon oferece a opção de criar um novo painel ou selecionar um existente. Para obter mais informações, consulte Usando CloudWatch painéis da Amazon no Guia do CloudWatch usuário da Amazon.



As EC2 métricas do Elastic Load Balancing e da Amazon estão habilitadas para todos os ambientes.

Com a integridade aprimorada, a EnvironmentHealth métrica é ativada e um gráfico é adicionado automaticamente ao console de monitoramento. A integridade aprimorada também acrescenta a página Integridade ao console de gerenciamento. Para obter uma lista das métricas de integridade aprimorada, consulte Publicação de métricas CloudWatch personalizadas da Amazon para um ambiente.

Usar a EB CLI para monitorar a integridade do ambiente

A interface de linha de comando do Elastic Beanstalk (EB CLI) é uma ferramenta de linha de comando para gerenciar ambientes. AWS Elastic Beanstalk Também é possível usar a CLI do EB para monitorar a integridade do ambiente em tempo real e com mais granularidade do que está disponível atualmente no console do Elastic Beanstalk

Depois de instalar e configurar a CLI do EB, você pode iniciar um novo ambiente e implantar seu código nele com eb create o comando. Se você já tem um ambiente criado no console do Elastic Beanstalk, é possível anexar a CLI do EB a ele, executando eb init em uma pasta do projeto e seguindo as solicitações (a pasta do projeto pode estar vazia).

> **Important**
>
> Certifique-se de que está usando a versão mais recente da EB CLI executando `pip install` com a opção `--upgrade`:
>
> ```
> $ sudo pip install --upgrade awsebcli
> ```
>
> Para concluir as instruções de instalação da CLI do EB, consulte Instale o EB CLI com script de configuração (recomendado).

Para usar a EB CLI para monitorar a integridade do ambiente, configure primeiro uma pasta de projeto local executando eb init e seguindo as solicitações. Para obter instruções completas, consulte Configurar a EB CLI.

Se você já tem um ambiente em execução no Elastic Beanstalk e deseja usar a CLI do EB para monitorar a integridade, siga estas etapas para anexá-lo ao ambiente existente.

Para anexar a EB CLI a um ambiente existente

1. Abra um terminal de linha de comando e navegue até a pasta do usuário.
2. Crie e abra uma nova pasta para o seu ambiente.
3. Execute o comando eb init e escolha o aplicativo e o ambiente cuja integridade que você quer monitorar. Se você tiver apenas um ambiente executando o aplicativo escolhido, a CLI do EB o selecionará automaticamente e não será necessário escolher o ambiente, como mostra o exemplo a seguir.

```
~/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
```

Para monitorar a integridade usando a EB CLI

1. Abra uma linha de comando e navegue até a pasta do projeto.
2. Execute o comando eb health para exibir o status de integridade das instâncias em seu ambiente. Neste exemplo, há cinco instâncias em execução em um 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
```

Neste exemplo, há uma única instância em execução em um 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
```

Leitura da saída

A saída exibe o nome do ambiente, a integridade geral do ambiente e a data atual na parte superior da tela.

```
elasticBeanstalkExa-env                                  Ok
 2015-07-08 23:13:20
```

As próximas três linhas exibem o tipo de ambiente (” WebServer "neste caso), a configuração (Ruby 2.1 com Puma) e um detalhamento de quantas instâncias estão em cada um dos sete estados.

```
WebServer
 Ruby 2.1 (Puma)
  total      ok    warning  degraded  severe    info   pending  unknown
    5        5        0        0        0        0        0        0
```

O resto da saída é dividido em quatro seções. A primeira exibe o status e a causa do status do ambiente geral e, em seguida, de cada instância. O exemplo a seguir mostra duas instâncias no ambiente com o status `Info` e a causa indicando que uma implantação foi iniciada.

```
  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
```

Para informações sobre o status de integridade e cores, consulte Status e cores de integridade.

A seção de solicitações exibe informações de logs do servidor Web sobre cada instância. Neste exemplo, cada instância está fazendo solicitações normalmente e não há erros.

```
  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
```

A seção de cpu mostra as métricas do sistema operacional de cada instância. A saída difere de acordo com o sistema operacional. Veja aqui a saída para ambientes 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
```

Veja aqui a saída para ambientes 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
```

Para obter informações sobre as métricas de servidor e de sistema operacional mostradas, consulte Métricas de instância.

A seção final, de implantações, mostra o status da implantação de cada instância. Se uma implantação contínua falhar, use o ID da implantação, o status e o rótulo da versão mostrados para identificar as instâncias no ambiente que estão executando a versão incorreta.

```
  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
```

Visualização de integridade interativa

O comando eb health exibe um snapshot da integridade do ambiente. Para atualizar as informações exibidas a cada 10 segundos, use a opção `--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, # # # #)
```

Este exemplo mostra um ambiente que foi recentemente ampliado de uma para cinco instâncias. A operação de escalabilidade foi bem-sucedida e agora todas as instâncias passam pelas verificações de integridade e estão prontas para fazer solicitações. No modo interativo, o status de integridade é atualizado a cada 10 segundos. No canto superior direito, um temporizador faz a contagem regressiva para a próxima atualização.

No canto inferior esquerdo, o relatório exibe uma lista de opções. Para sair do modo interativo, pressione Q. Para rolar, pressione as teclas de seta. Para ver uma lista de comandos adicionais, pressione H.

Opções de visualização de integridade interativa

Na visualização interativa da integridade do ambiente, use o teclado para ajustar a visualização e informar ao Elastic Beanstalk para substituir ou reinicializar instâncias individuais. Para ver uma lista de comandos disponíveis na visualização do relatório de integridade no modo interativo, pressione 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)
```

Relatórios de integridade básica

Este tópico explica a funcionalidade oferecida pela integridade básica do Elastic Beanstalk.

AWS Elastic Beanstalk usa informações de várias fontes para determinar se seu ambiente está disponível e processando solicitações da Internet. A integridade de um ambiente é representada por uma das quatro cores e é exibida na página de visão geral do ambiente do console do Elastic Beanstalk. Também está disponível na DescribeEnvironmentsAPI e por meio de chamadas eb status com a CLI do EB.

O sistema básico de relatórios de integridade fornece informações sobre a integridade das instâncias em um ambiente do Elastic Beanstalk com base em verificações de integridade realizadas pelo Elastic Load Balancing para ambientes com carga balanceada ou pelo Amazon Elastic Compute Cloud para ambientes de instância única.

Além de verificar a integridade de suas EC2 instâncias, o Elastic Beanstalk também monitora os outros recursos em seu ambiente e relata recursos ausentes ou configurados incorretamente que podem fazer com que seu ambiente fique indisponível para os usuários.

As métricas coletadas pelos recursos em seu ambiente são publicadas CloudWatch na Amazon em intervalos de cinco minutos. Isso inclui métricas do sistema operacional e métricas EC2 de solicitação do Elastic Load Balancing. Você pode visualizar gráficos com base nessas CloudWatch métricas na página Monitoramento do console do ambiente. Para integridade básica, essas métricas não são usadas para determinar a integridade de um ambiente.

Tópicos

- Cores de integridade
- Verificações de integridade do Elastic Load Balancing
- Verificações de integridade do ambiente de nível do operador e instância única
- Verificações adicionais
- CloudWatch Métricas da Amazon

Cores de integridade

O Elastic Beanstalk relata a integridade de um ambiente de servidor Web de acordo com o modo como a aplicação em execução responde à verificação de integridade. O Elastic Beanstalk usa uma de quatro cores para descrever o status, como mostrado na tabela a seguir:

Cor	Descrição
Cinza	O ambiente está sendo atualizado.
Verde	O ambiente passou na verificação de integridade mais recente. Pelo menos uma instância em seu ambiente está disponível e aceitando solicitações.

Cor	Descrição
Amarelo	Seu ambiente falhou em uma ou mais verificações de integridade. Algumas solicitações a seu ambiente estão falhando.
Vermelha	Seu ambiente falhou em três ou mais verificações de integridade, ou um recurso do ambiente ficou indisponível. As solicitações falham consisten temente.

Essas descrições se aplicam somente a ambientes que usam relatórios de integridade básica. Consulte Status e cores de integridade para obter detalhes relacionados à integridade aprimorada.

Verificações de integridade do Elastic Load Balancing

Em um ambiente com carga balanceada, o Elastic Load Balancing envia uma solicitação para cada instância em um ambiente a cada 10 segundos para confirmar que as instâncias estão íntegras. Por padrão, o load balancer é configurado para abrir uma conexão TCP na porta 80. Se a instância reconhece a conexão, ela é considerada íntegra.

Você pode optar por substituir essa configuração especificando um recurso existente no seu aplicativo. Se você especificar um caminho, como `/health`, o URL da verificação de integridade é definido como `HTTP:80/health`. O URL da verificação de integridade deve ser definido como um caminho que sempre é atendido pelo seu aplicativo. Se estiver definido como uma página estática que é atendida ou armazenada em cache pelo servidor web na frente de seu aplicativo, as verificações de integridade não revelarão problemas com o servidor de aplicativo ou o contêiner da Web. Para obter instruções sobre como modificar o URL de verificação de integridade, consulte Verificação de integridade.

Se o URL de uma verificação de integridade está configurado, o Elastic Load Balancing espera uma solicitação GET que ele envia para retornar a resposta `200 OK`. O aplicativo na passará na verificação de integridade se não responder dentro de 5 segundos ou se responder com qualquer outro código de status HTTP. Após 5 falhas consecutivas na verificação de integridade, o Elastic Load Balancing tira a instância de serviço.

Para obter mais informações sobre verificações de integridade do Elastic Load Balancing, consulte Verificação de integridade no Guia do usuário do Elastic Load Balancing

> **Note**
>
> A configuração de um URL de verificação de integridade não altera o comportamento de verificação de integridade de um grupo de Auto Scaling de um ambiente. Uma instância não íntegra é removida do balanceador de carga, mas não é substituída automaticamente pelo Amazon Auto EC2 Scaling, a menos que você configure o Amazon Auto EC2 Scaling para usar a verificação de saúde do Elastic Load Balancing como base para substituir instâncias. Para configurar o Amazon EC2 Auto Scaling para substituir instâncias que falham em uma verificação de integridade do Elastic Load Balancing, consulte. Configuração de verificação de integridade do Auto Scaling para seu ambiente Elastic Beanstalk

Verificações de integridade do ambiente de nível do operador e instância única

Em uma única instância ou ambiente de nível de trabalho, o Elastic Beanstalk determina a integridade da instância monitorando o status da instância Amazon. EC2 As configurações de integridade do Elastic Load Balancing, incluindo a verificação de integridade HTTP URLs, não podem ser usadas nesses tipos de ambiente.

Para obter mais informações sobre as verificações de status de EC2 instâncias da Amazon, consulte Monitoramento de instâncias com verificações de status no Guia EC2 do usuário da Amazon.

Verificações adicionais

Além das verificações de integridade do Elastic Load Balancing, o Elastic Beanstalk monitora recursos no ambiente e altera o status de integridade para vermelho caso eles não sejam implantados, não estiverem configurados corretamente ou ficarem indisponíveis. Essas verificações confirmam que:

- O grupo de Auto Scaling do ambiente está disponível e tem no mínimo uma instância.
- O security group do ambiente está disponível e configurado para permitir o tráfego de entrada na porta 80.
- O CNAME do ambiente existe e está apontando para o load balancer correto.
- Em um ambiente de operador, a fila do Amazon Simple Queue Service (Amazon SQS) está sendo pesquisada pelo menos uma vez a cada três minutos.

CloudWatch Métricas da Amazon

Com relatórios básicos de saúde, o serviço Elastic Beanstalk não publica nenhuma métrica na Amazon. CloudWatch As CloudWatch métricas usadas para produzir gráficos na página Monitoramento do console do ambiente são publicadas pelos recursos do seu ambiente.

Por exemplo, EC2 publica as seguintes métricas para as instâncias no grupo Auto Scaling do seu ambiente:

`CPUUtilization`

Porcentagem de unidades de computação em uso no momento.

`DiskReadBytes`, `DiskReadOps`, `DiskWriteBytes`, `DiskWriteOps`

O número de bytes lidos e gravados e o número de operações de leitura e gravação.

`NetworkIn`, `NetworkOut`

O número de bytes enviados e recebidos.

O Elastic Load Balancing publica as métricas a seguir para o balanceador de carga do ambiente:

`BackendConnectionErrors`

Número de falhas de conexão entre o load balancer e as instâncias do ambiente.

`HTTPCode_Backend_2XX`, `HTTPCode_Backend_4XX`

Número de códigos de resposta de êxito (2XX) e erro do cliente (4XX) gerados pelas instâncias em seu ambiente.

`Latency`

O número de segundos entre o momento em que o load balancer retransmite uma solicitação para uma instância e o momento em que a resposta é recebida.

`RequestCount`

O número de solicitações concluídas.

Essas listas não são abrangentes. Para obter uma lista completa das métricas que podem ser relatadas para esses recursos, consulte os seguintes tópicos no Amazon CloudWatch Developer Guide:

Métricas

Namespace	Tópico
AWS::ElasticLoadBalancing::LoadBalancer	Métricas e recursos do Elastic Load Balancing
AWS::AutoScaling::AutoScalingGrupo	Métricas e recursos do Amazon Elastic Compute Cloud
AWS::SQS::Queue	Métricas e recursos do Amazon SQS
AWS: :RDS:: DBInstance	Métricas e dimensões do Amazon RDS

Métrica de integridade do ambiente de operador

Somente para ambientes de trabalho, o daemon SQS publica uma métrica personalizada para a integridade do ambiente CloudWatch, em que um valor de 1 é verde. Você pode revisar os dados da métrica de CloudWatch saúde em sua conta usando o `ElasticBeanstalk/SQSD` namespace. A dimensão da métrica é `EnvironmentName` e o nome da métrica é `Health`. Todas as instâncias publicam suas métricas no mesmo namespace.

Para permitir que o daemon publique métricas, o perfil de instância do ambiente deve ter permissão para chamar `cloudwatch:PutMetricData`. Essa permissão está incluída no perfil de instância padrão. Para obter mais informações, consulte Gerenciar perfis de instância do Elastic Beanstalk.

Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk

Esta seção explica a funcionalidade do recurso Integridade aprimorada do Elastic Beanstalk.

Os relatórios de integridade aprimorados são um recurso que você pode ativar em seu ambiente AWS Elastic Beanstalk para permitir a coleta de informações adicionais sobre recursos em seu ambiente. O Elastic Beanstalk analisa as informações reunidas para fornecer uma visão melhor da integridade geral do ambiente e ajudar na identificação de problemas que podem fazer com que a aplicação se torne indisponível.

Além das alterações na forma como as cores de integridade funcionam, a integridade aprimorada adiciona um descritor de status que fornece um indicador da gravidade dos problemas observados

quando um ambiente está amarelo ou vermelho. Quando mais informações sobre o status atual estiverem disponíveis, você poderá escolher o botão Causes para ver informações detalhadas de integridade na página de integridade.

Para fornecer informações de saúde detalhadas sobre as EC2 instâncias da Amazon em execução em seu ambiente, o Elastic Beanstalk inclui um agente de saúde na Amazon Machine Image (AMI) para cada versão da plataforma que oferece suporte à integridade aprimorada. O agente de integridade monitora os logs de servidor da Web e as métricas do sistema, e os transmite para o serviço do Elastic Beanstalk. O Elastic Beanstalk analisa essas métricas e dados do Elastic Load Balancing e do Amazon EC2 Auto Scaling para fornecer uma visão geral da saúde de um ambiente.

Além de coletar e apresentar informações sobre os recursos de ambiente, o Elastic Beanstalk monitora os recursos no ambiente em busca de várias condições de erro e fornece notificações para ajudar você a evitar falhas e resolver problemas de configuração. Os fatores que influenciam a integridade do seu ambiente incluem os resultados de cada solicitação atendida pelo seu aplicativo, as métricas do sistema operacional de suas instâncias e o status da implantação mais recente.

É possível visualizar o status de integridade em tempo real usando a página visão geral do ambiente do console do Elastic Beanstalk ou o comando eb health na Interface de Linhas de Comando do Elastic Beanstalk (CLI do EB). Para registrar e monitorar a integridade do ambiente e da instância ao longo do tempo, você pode configurar seu ambiente para publicar as informações coletadas pelo Elastic Beanstalk para melhorar os relatórios de saúde para a CloudWatch Amazon como métricas personalizadas. CloudWatch as cobranças por métricas personalizadas se aplicam a todas as métricas`EnvironmentHealth`, exceto, o que é gratuito.

> **Notas da plataforma Windows**
>
> Ao habilitar os relatórios de integridade aprimorada em um ambiente do Windows Server, não altere a configuração de log do IIS. Para o monitoramento de integridade aprimorada funcionar corretamente, o registro do IIS deve ser configurado com o formato W3C e os destinos de evento de log ETW event only (Evento ETW somente) ou Both log file and ETW event (Arquivo de log e evento ETW).
>
> Além disso, não desabilite nem interrompa o serviço do Windows do agente de integridade do Elastic Beanstalk em nenhuma das instâncias do ambiente. Para coletar e relatar informações de integridade aprimorada em uma instância, esse serviço deve estar habilitado e em execução.

Na primeira vez que você cria um ambiente, o Elastic Beanstalk solicita a criação dos perfis necessários e habilita relatórios de integridade aprimorados por padrão. Continue lendo para obter detalhes sobre como os relatórios de integridade aprimorada funcionam ou consulte Habilitar relatórios de integridade aprimorada do Elastic Beanstalk para começar a usá-los imediatamente.

Tópicos

- Agente de integridade do Elastic Beanstalk
- Fatores que determinam a integridade de instâncias e ambientes
- Personalização de regra de verificação de integridade
- Funções de integridade aperfeiçoada
- Autorização de integridade aprimorada
- Eventos de integridade aprimorada
- Comportamento dos relatórios de integridade aprimorada durante as atualizações, as implantações e a escalabilidade
- Habilitar relatórios de integridade aprimorada do Elastic Beanstalk
- Monitoramento de integridade aprimorada com o console de gerenciamento do ambiente
- Status e cores de integridade
- Métricas de instância
- Configurar as regras de integridade aperfeiçoada de um ambiente
- Publicação de métricas CloudWatch personalizadas da Amazon para um ambiente
- Usar relatórios de integridade aprimorada com a API do Elastic Beanstalk
- Formato de log de integridade aprimorado
- Notificações e solução de problemas

Agente de integridade do Elastic Beanstalk

O agente de saúde do Elastic Beanstalk é um processo daemon (ou serviço, em ambientes Windows) executado em cada instância da EC2 Amazon em seu ambiente, monitorando as métricas de integridade do sistema operacional e do aplicativo e relatando problemas ao Elastic Beanstalk. O agente de integridade está incluído em todas as versões da plataforma, a partir da versão 2.0 de cada plataforma.

O agente de saúde relata métricas semelhantes às publicadas CloudWatch pelo Amazon EC2 Auto Scaling e pelo Elastic Load Balancing como parte dos relatórios básicos de saúde, incluindo carga

de CPU, códigos HTTP e latência. O agente de integridade, no entanto, relata diretamente ao Elastic Beanstalk, com maior granularidade e frequência do que os relatórios de integridade básica.

Para a integridade básica, essas métricas são publicados a cada cinco minutos e podem ser monitoradas com gráficos no console de gerenciamento do ambiente. Com a integridade aprimorada, o agente de integridade do Elastic Beanstalk relata métricas para o Elastic Beanstalk a cada 10 segundos. O Elastic Beanstalk usa as métricas fornecidas pelo agente de integridade para determinar o status de integridade de cada instância no ambiente e, combinadas com outros fatores, para determinar a integridade geral do ambiente.

A integridade geral do ambiente pode ser visualizada em tempo real na página de visão geral do ambiente do console do Elastic Beanstalk e é publicada CloudWatch pela Elastic Beanstalk a cada 60 segundos. É possível visualizar as métricas detalhadas relatadas pelo agente de integridade em tempo real com o comando eb health na CLI do EB.

Por uma taxa adicional, você pode optar por publicar métricas individuais de instância e ambiente a CloudWatch cada 60 segundos. As métricas publicadas no CloudWatch podem ser usadas para criar gráficos de monitoramento no console de gerenciamento de ambiente.

Os relatórios de integridade aprimorada implicam em cobrança apenas se você optar por publicar métricas de integridade aprimorada no CloudWatch. Quando você usa a integridade aprimorada, ainda tem as métricas de integridade básica publicadas gratuitamente, mesmo se optar por não publicar as métricas de integridade aprimorada.

Consulte Métricas de instância para obter detalhes sobre as métricas relatadas pelo agente de integridade. Para obter detalhes sobre a publicação de métricas de saúde aprimoradas em CloudWatch, consultePublicação de métricas CloudWatch personalizadas da Amazon para um ambiente.

Fatores que determinam a integridade de instâncias e ambientes

Além das verificações do sistema de relatórios de integridade básica, incluindo o Verificações de integridade do Elastic Load Balancing e o monitoramento de recursos, os relatórios de integridade aprimorada do Elastic Beanstalk coletam dados adicionais sobre o estado das instâncias no ambiente. Isso inclui métricas de sistema operacional, logs de servidor e o estado de operações contínuas de ambiente, como implantações e atualizações. O serviço de relatório de integridade do Elastic Beanstalk combina informações de todas as origens disponíveis e as analisa para determinar a integridade geral do ambiente.

Operações e comandos

Quando você executa uma operação no ambiente, como a implantação de uma nova versão de uma aplicação, o Elastic Beanstalk faz várias alterações que afetam o status de integridade do ambiente.

Por exemplo, quando você implanta uma nova versão de um aplicativo em um ambiente que está executando várias instâncias, você pode ver mensagens semelhantes à seguinte ao monitorar a integridade do ambiente com a CLI do EB.

```
  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
```

Neste exemplo, o status geral do ambiente é 0k e a causa desse status é que o comando está sendo executado em 3 de 5 instâncias. Três das instâncias no ambiente têm o status Pendente, indicando que há uma operação em andamento.

Quando uma operação é concluída, o Elastic Beanstalk relata as informações adicionais sobre ela. Por exemplo, o Elastic Beanstalk exibe as informações a seguir sobre uma instância que já foi atualizada com a nova versão da aplicação:

```
i-f6a2d525      Ok         Application deployment completed 23 seconds ago and took 26
 seconds
```

As informações de integridade da instância também incluem detalhes sobre a implantação mais recente a cada instância em seu ambiente. Cada instância relata um ID de implantação e o status. O ID de implantação é um número inteiro que aumenta em um a cada vez que você implanta uma nova versão do aplicativo ou altera as definições das opções de configuração na instância, como as variáveis de ambiente. Você pode usar as informações de implantação para identificar as instâncias que estão executando a versão incorreta de seu aplicativo após uma falha de implantação contínua.

Na coluna de causa, o Elastic Beanstalk inclui mensagens informativas sobre operações bem-sucedidas e outros estados de integridade em várias verificações de integridade, mas elas não são

mantidas indefinidamente. As causas para status de não integridade no ambiente são mantidas até o ambiente retornar a um status de integridade.

Tempo de espera do comando

O Elastic Beanstalk aplica um tempo de espera do comando a partir do momento que uma operação começa para permitir que uma instância faça a transição para um estado de integridade. Esse tempo limite do comando é definido na configuração de implantação e atualização do ambiente (no namespace aws:elasticbeanstalk:command), sendo 10 minutos o padrão.

Durante atualizações contínuas, o Elastic Beanstalk aplica um tempo de espera separado para cada lote na operação. Esse tempo de espera é definido como parte da configuração de atualização contínua do ambiente (no namespace aws:autoscaling:updatepolicy:rollingupdate). Se todas as instâncias no lote estiverem íntegras dentro do tempo limite de atualização contínua, a operação continuará no próximo lote. Se não, a operação falha.

> **Note**
>
> Caso a aplicação não passe nas verificações de integridade com um status OK, mas esteja estável em outro nível, você poderá definir a opção `HealthCheckSuccessThreshold` no `aws:elasticbeanstalk:command namespace` para alterar o nível no qual o Elastic Beanstalk considera uma instância como íntegra.

Para que um ambiente de servidor web seja considerado íntegro, cada instância no ambiente ou no lote deve passar por 12 verificações de integridade consecutivas ao longo de dois minutos. Para um ambiente do nível de operador, cada instância deve passar por 18 verificações de integridade. Antes de o comando atingir o tempo limite, o Elastic Beanstalk não reduz o status de integridade de um ambiente quando a verificação de integridade falha. Se as instâncias no ambiente se tornarem íntegras dentro do tempo limite do comando, a operação será bem-sucedida.

Solicitações HTTP

Quando nenhuma operação está em andamento em um ambiente, a origem principal de informações sobre a integridade do ambiente e da instância são os logs do servidor web de cada instância. Para determinar a integridade de uma instância e a integridade geral do ambiente, o Elastic Beanstalk considera o número de solicitações, o resultado de cada uma e a velocidade com que cada solicitação foi resolvida.

Em plataformas Linux, o Elastic Beanstalk lê e analisa logs do servidor Web para obter informações sobre solicitações HTTP. Na plataforma Windows Server, o Elastic Beanstalk recebe essas informações diretamente do servidor Web do IIS.

O ambiente pode não ter um servidor web ativo. Por exemplo, a plataforma do Docker de vários contêineres não inclui um servidor web. Outras plataformas incluem um servidor web e o aplicativo pode desabilitá-lo. Nesses casos, o ambiente exige configuração adicional para fornecer ao agente de integridade do Elastic Beanstalk os logs no formato necessário para que as informações de integridade sejam retransmitidas ao serviço do Elastic Beanstalk. Para mais detalhes, consulte Formato de log de integridade aprimorado.

Métricas do sistema operacional

O Elastic Beanstalk monitora as métricas do sistema operacional relatadas pelo agente de integridade para identificar as instâncias que são consistentemente baixas em recursos do sistema.

Consulte Métricas de instância para obter detalhes sobre as métricas relatadas pelo agente de integridade.

Personalização de regra de verificação de integridade

Os relatórios de integridade aprimorada do Elastic Beanstalk dependem de um conjunto de regras para determinar a integridade do ambiente. Algumas dessas regras podem não ser adequadas para o seu aplicativo. Um caso comum é um aplicativo que retorna erros HTTP 4xx frequentes por padrão. O Elastic Beanstalk, usando uma de suas regras padrão, conclui que algo está errado e altera o status de integridade do ambiente de OK para Warning (Aviso), Degraded (Reduzido) ou Severe (Grave), de acordo com a taxa de erros. Para lidar com esse caso corretamente, o Elastic Beanstalk permite configurar essa regra e ignorar erros de HTTP 4xx da aplicação Para obter mais detalhes, consulte Configurar as regras de integridade aperfeiçoada de um ambiente.

Funções de integridade aperfeiçoada

Os relatórios de integridade aperfeiçoada requerem duas funções: uma função de serviço para o Elastic Beanstalk e um perfil de instância para o ambiente. A função de serviço permite que o Elastic Beanstalk interaja AWS com outros serviços em seu nome para coletar informações sobre os recursos em seu ambiente. O perfil de instância permite que as instâncias no ambiente gravem logs no Amazon S3 e comuniquem informações de integridade aprimoradas ao serviço do Elastic Beanstalk.

Quando você cria um ambiente do Elastic Beanstalk usando o console do Elastic Beanstalk ou a CLI do EB, o Elastic Beanstalk cria uma função de serviço padrão e anexa as políticas gerenciadas necessárias a um perfil de instância padrão para o ambiente.

Se você usa a API, um SDK ou o AWS CLI para criar ambientes, deve criar essas funções com antecedência e especificá-las durante a criação do ambiente para usar a integridade aprimorada. Para obter instruções sobre a criação de funções adequadas para seus ambientes, consulte Perfis do Elastic Beanstalk Service, perfis de instância e políticas de usuário.

Recomendamos que você use políticas gerenciadas para o perfil de instância e a função de serviço. Políticas gerenciadas são políticas AWS Identity and Access Management (IAM) que o Elastic Beanstalk mantém. O uso de políticas gerenciadas garante que o ambiente tenha todas as permissões necessárias para funcionar corretamente.

Para o perfil de instância, você pode usar a política gerenciada `AWSElasticBeanstalkWebTier` ou `AWSElasticBeanstalkWorkerTier`, para um ambiente de nível de servidor Web ou de nível de operador, respectivamente. Para obter detalhes sobre essas duas políticas gerenciadas de perfil de instância, consulte the section called “Perfis de instância”.

Autorização de integridade aprimorada

As políticas gerenciadas do perfil da instância do Elastic Beanstalk contêm permissão para a ação `elasticbeanstalk:PutInstanceStatistics` . Essa ação não faz parte da API do Elastic Beanstalk. Ela faz parte de uma API diferente que as instâncias de ambiente usam internamente para transmitir informações de integridade aprimoradas para o serviço do Elastic Beanstalk. Você não chama essa API diretamente.

Quando você cria um novo ambiente, a autorização para a ação `elasticbeanstalk:PutInstanceStatistics` é habilitada por padrão. Para aumentar a segurança do ambiente e ajudar a impedir a falsificação de dados de integridade em seu nome, recomendamos manter a autorização para esta ação habilitada. Se você usar políticas gerenciadas para seu perfil da instância, esse recurso estará disponível para seu novo ambiente sem necessidade de configuração adicional. Se você usar um perfil da instância personalizado em vez de uma política gerenciada, o ambiente poderá mostrar o status de integridade Sem dados Isso acontece porque as instâncias não estão autorizadas para a ação que comunica dados de integridade aprimorados ao serviço.

Para autorizar a ação, inclua a instrução a seguir no perfil de instância.

```
    {
```

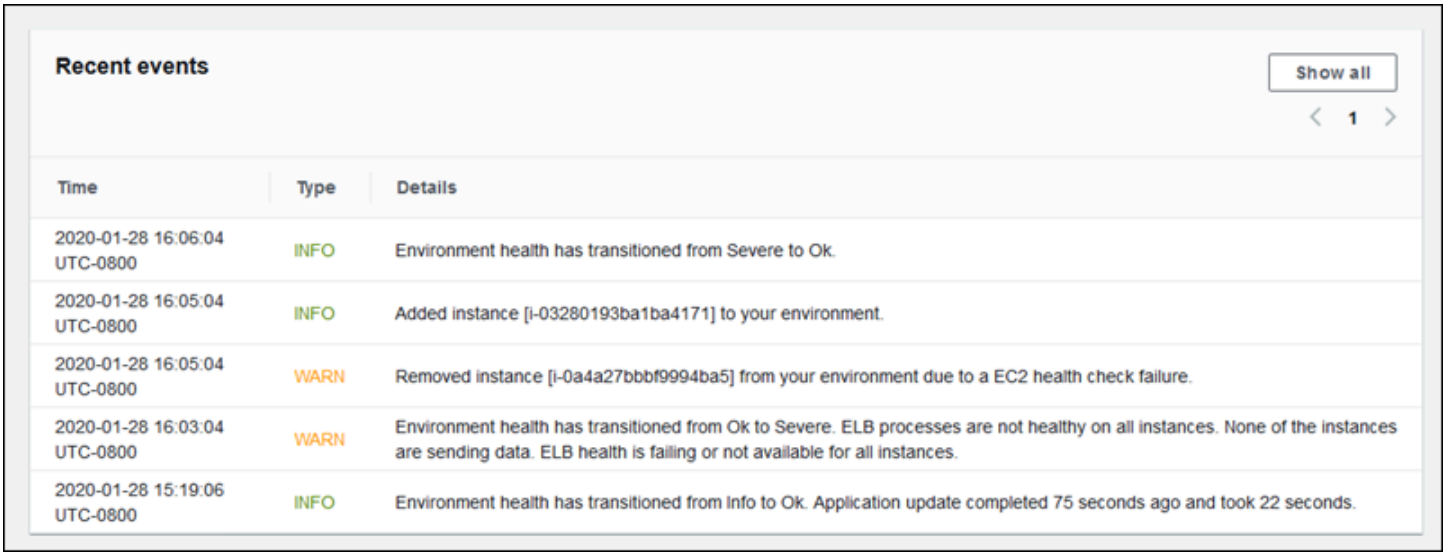
```
      "Sid": "ElasticBeanstalkHealthAccess",
      "Action": [
        "elasticbeanstalk:PutInstanceStatistics"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:elasticbeanstalk:*:*:application/*",
        "arn:aws:elasticbeanstalk:*:*:environment/*"
      ]
    }
```

Se você não quiser usar a autorização de integridade aprimorada no momento, desabilite-a definindo a opção `EnhancedHealthAuthEnabled` no namespace the section called "aws:elasticbeanstalk:healthreporting:system" como `false`. Se a opção estiver desabilitada, as permissões descritas anteriormente não são necessárias. Você pode removê-las do perfil de instância para acesso mínimo a suas aplicações e ambientes.

> **Note**
>
> Anteriormente, a configuração padrão para `EnhancedHealthAuthEnabled` foi `false`, que resultou na autorização para a ação `elasticbeanstalk:PutInstanceStatistics` também ser desabilitada por padrão. Para habilitar essa ação para um ambiente existente, defina a opção `EnhancedHealthAuthEnabled` no the section called "aws:elasticbeanstalk:healthreporting:system" namespace to `true`. É possível configurar essa opção usando uma configuração de opção em um arquivo de configuração.

Eventos de integridade aprimorada

O sistema de integridade aprimorada gera eventos quando um ambiente alterna entre estados. O exemplo a seguir mostra eventos gerados pela transição de um ambiente entre os estados Info (Informações), OK e Severe (Grave).



Quando ocorre uma transição para um estado pior, o evento de integridade aprimorada inclui uma mensagem indicando a causa da transição.

Nem todas as alterações no status em um nível de instância fazem com que o Elastic Beanstalk gere um evento. Para evitar alarmes falsos, o Elastic Beanstalk gerará um evento relacionado à integridade somente se um problema persistir em várias verificações.

As informações de integridade em tempo real no nível do ambiente, incluindo status, cor e causa, estão disponíveis na página de visão geral do ambiente do console do Elastic Beanstalk e na CLI do EB. Ao anexar a CLI do EB ao ambiente e executar o comando eb health, você também poderá visualizar o status em tempo real de cada uma das instâncias no ambiente.

Comportamento dos relatórios de integridade aprimorada durante as atualizações, as implantações e a escalabilidade

Habilitar os relatórios de integridade aprimorada pode afetar como seu ambiente se comporta durante atualizações e implantações de configuração. O Elastic Beanstalk não concluirá um lote de atualizações até que todas as instâncias passem nas verificações de integridade de forma consistente. Além disso, como os relatórios de integridade aprimorada aplicam um critério mais rigoroso de integridade e monitoram mais fatores, as instâncias que passam na verificação de integridade do ELB dos relatórios de integridade básica não passarão necessariamente com os relatórios de integridade aprimorada. Consulte os tópicos sobre atualizações de configuração contínuas e implantações contínuas para obter detalhes sobre como as verificações de integridade afetam o processo de atualização.

Os relatórios de integridade aprimorada também podem destacar a necessidade de definir um URL de verificação de integridade adequado para o Elastic Load Balancing. Quando seu ambiente expandir para atender à demanda, novas instâncias começarão a atender solicitações assim que passarem por verificações de integridade do ELB suficientes. Se não for configurado um URL de verificação de integridade, isso poderá levar apenas 20 segundos após uma nova instância ser capaz de aceitar uma conexão TCP.

Se o seu aplicativo não tiver concluído a inicialização até o momento em que o load balancer o declarar íntegro o suficiente para receber tráfego, você verá um afluxo de solicitações falhadas e seu ambiente começará a não passar nas verificações de integridade. Um URL de verificação de integridade que atinge um caminho fornecido pelo aplicativo pode evitar esse problema. As verificações de integridade do ELB não passarão até que uma solicitação GET para a URL de verificação de integridade retorne um código de status 200.

Habilitar relatórios de integridade aprimorada do Elastic Beanstalk

Este tópico explica como os relatórios de integridade aprimorados são habilitados. Ele fornece procedimentos para ativar o recurso de integridade aprimorada para o seu ambiente com o console do Elastic Beanstalk, a EB CLI e com uma configuração .ebextensions.

Os novos ambientes criados com as versões mais recentes da plataforma incluem o agente de AWS Elastic Beanstalk saúde, que oferece suporte a relatórios de saúde aprimorados. Se você criar o ambiente no console do Elastic Beanstalk ou com a CLI do EB, a integridade aprimorada será habilitada por padrão. Também é possível definir a preferência de relatórios de integridade no código-fonte do aplicativo usando arquivos de configuração.

Os relatórios de integridade aprimorada exigem o perfil da instância e a função de serviço com o conjunto padrão de permissões. Quando você cria um ambiente no console do Elastic Beanstalk, o Elastic Beanstalk cria as funções necessárias automaticamente. Para obter instruções sobre como criar seu primeiro ambiente, consulte Saiba como começar a usar o Elastic Beanstalk.

Tópicos

- Habilitar relatórios de integridade aprimorada usando o console do Elastic Beanstalk
- Habilitar relatórios de integridade aprimorada usando a CLI do EB
- Habilitar relatórios de integridade aprimorada usando um arquivo de configuração

Habilitar relatórios de integridade aprimorada usando o console do Elastic Beanstalk

Como habilitar relatórios de integridade aprimorada em um ambiente em execução usando o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatórios de integridade), para System (Sistema), escolha Enhanced (Avançado).

   > **Note**
   >
   > As opções de relatórios de integridade avançados não serão exibidas se você estiver usando uma Plataforma ou versão incompatível.

6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

O console do Elastic Beanstalk usa como padrão os relatórios de integridade aprimorada ao criar um novo ambiente com uma versão de plataforma de versão 2 (v2). Você pode desabilitar os relatórios de integridade aprimorada alterando a respectiva opção durante a criação do ambiente.

Como desabilitar os relatórios de integridade aprimorada durante a criação de um ambiente usando o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. Crie um aplicativo ou selecione um que já exista.
3. Crie um ambiente. Na página Create a new environment (Criar um novo ambiente), antes de escolher Create environment (Criar ambiente), escolha Configure more options (Configurar mais opções).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatórios de integridade), para System (Sistema), escolha Basic (Básico).
6. Escolha Salvar.

Habilitar relatórios de integridade aprimorada usando a CLI do EB

Quando você cria um novo ambiente com o comando eb create, a EB CLI habilita relatórios de integridade aprimorada, por padrão, e aplica a função de serviço e o perfil da instância padrão.

É possível especificar outra função de serviço por nome usando a opção `--service-role`.

Se você tem um ambiente em execução com relatórios de integridade básica da versão de plataforma v2 e quer mudar para a integridade aprimorada, siga estas etapas.

Para habilitar a integridade aprimorada no ambiente em execução usando a EB CLI

1. Use o comando eb config para abrir o arquivo de configuração no editor de texto padrão.

   ```
   ~/project$ eb config
   ```

2. Localize o namespace `aws:elasticbeanstalk:environment` na seção de configurações. Certifique-se de que o valor de `ServiceRole` não seja nulo e que corresponda ao nome da sua função de serviço.

   ```
     aws:elasticbeanstalk:environment:
       EnvironmentType: LoadBalanced
       ServiceRole: aws-elasticbeanstalk-service-role
   ```

3. No namespace `aws:elasticbeanstalk:healthreporting:system:`, altere o valor de `SystemType` para **enhanced**.

   ```
     aws:elasticbeanstalk:healthreporting:system:
       SystemType: enhanced
   ```

4. Salve o arquivo de configuração e feche o editor de texto.

5. A EB CLI inicia uma atualização do ambiente para aplicar as alterações de configuração. Aguarde a conclusão da operação ou pressione Ctrl+C para sair com segurança.

   ```
   ~/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.
   ```

Habilitar relatórios de integridade aprimorada usando um arquivo de configuração

Habilite os relatórios de integridade aprimorada, incluindo um arquivo de configuração no pacote de origem. O exemplo a seguir mostra um arquivo de configuração que habilita relatórios de integridade aprimorada e atribui o perfil da instância e serviço padrão ao 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 você criou seu próprio perfil de instância ou função de serviço, substitua o texto destacado pelos nomes dessas funções.

Monitoramento de integridade aprimorada com o console de gerenciamento do ambiente

Ao habilitar os relatórios de integridade aprimorados no AWS Elastic Beanstalk, você pode monitorar a integridade do ambiente no console de gerenciamento do ambiente.

Tópicos

- Visão geral do ambiente
- Página de integridade do ambiente
- Página de monitoramento

Visão geral do ambiente

A visão geral do ambiente exibe o status da integridade do ambiente e lista eventos que fornecem informações sobre alterações recentes no status da integridade.

Como visualizar a visão geral do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS

2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.

Para obter informações detalhadas sobre a integridade do ambiente atual, abra a página Health (Integridade) selecionando Causes (Causas). Como alternativa, no painel de navegação, escolha Health (Integridade).

Página de integridade do ambiente

A página Health exibe status de saúde, métricas e causas para o meio ambiente e para cada EC2 instância da Amazon no ambiente.

> **Note**
>
> O Elastic Beanstalk exibirá a página Health (Integridade) somente se você tiver habilitado o monitoramento de integridade aprimorada para o ambiente.

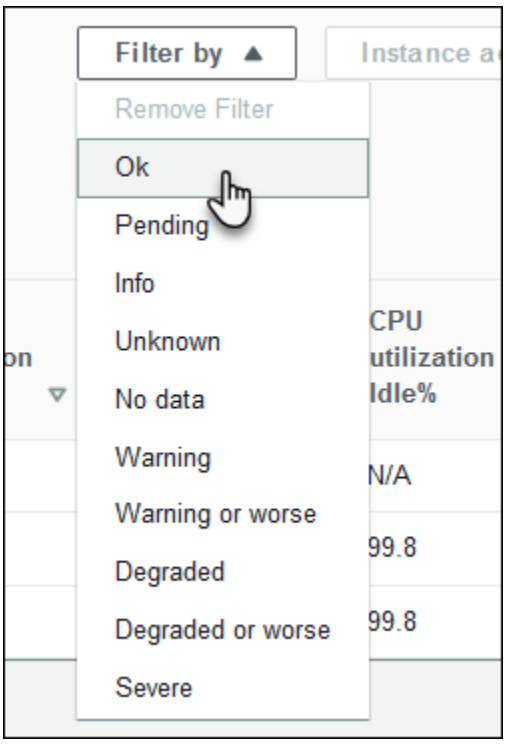
A imagem a seguir mostra a página Health (Integridade) para um ambiente Linux.



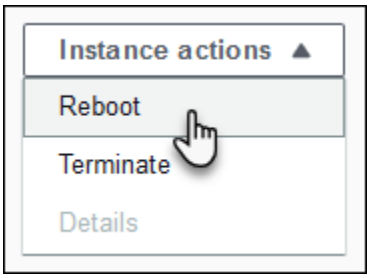
A imagem a seguir mostra a página Health (Integridade) para um ambiente Windows. Observe que as métricas de CPU são diferentes das contidas em um ambiente Linux.



Na parte superior da página, é possível ver o número total de instâncias de ambiente, bem como o número de instâncias por status. Para exibir somente as instâncias que têm um status específico, escolha Filter By (Filtrar por) e selecione um status.



Para reinicializar ou encerrar uma instância não íntegra, escolha Instance Actions e, em seguida, Reboot ou Terminate.



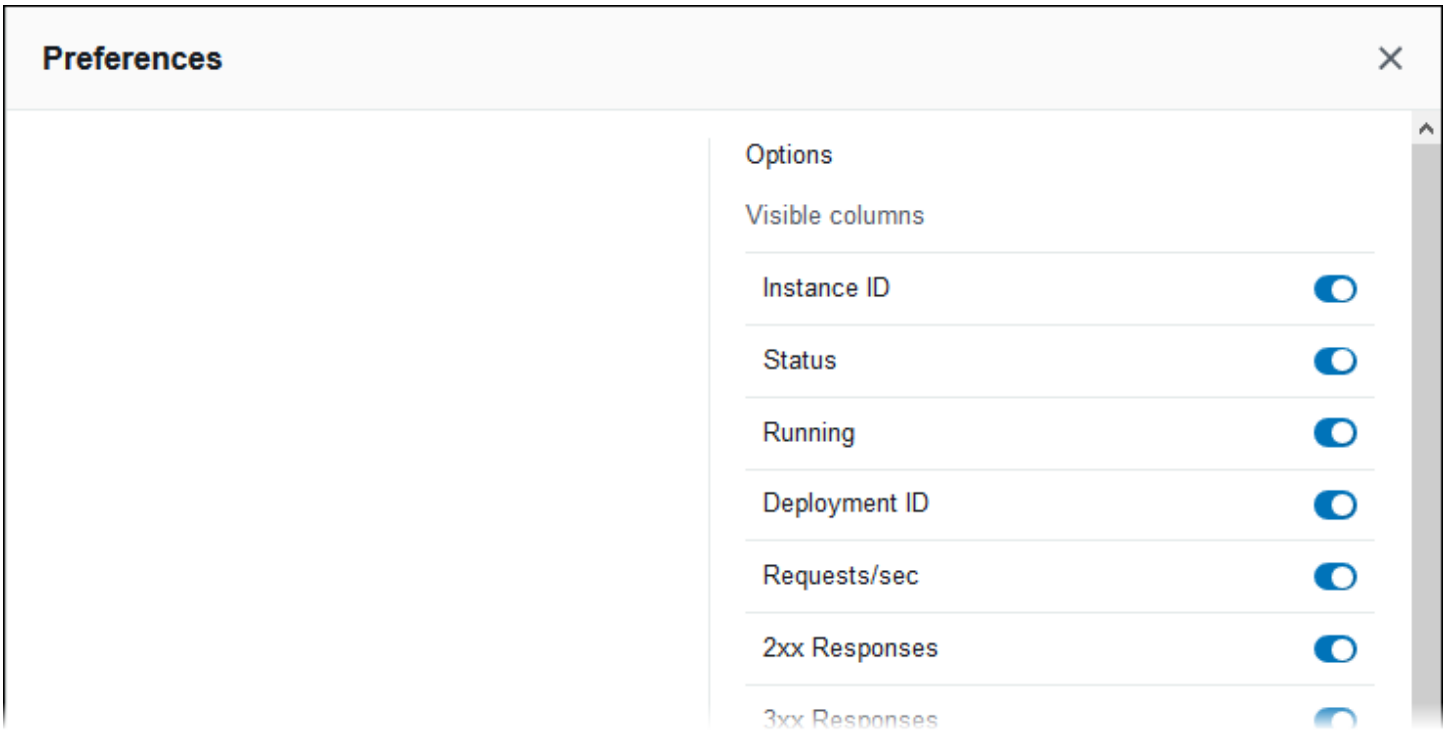
O Elastic Beanstalk atualiza a página Health (Integridade) a cada 10 segundos. Ela relata informações sobre o ambiente e a integridade da instância.

Para cada EC2 instância da Amazon no ambiente, a página exibe o ID e o status da instância, a quantidade de tempo desde que a instância foi lançada, o ID da implantação mais recente executada na instância, as respostas e a latência das solicitações atendidas pela instância e as informações de carga e utilização da CPU. A linha Overall (Geral) exibe informações de médias de resposta e latência para todo o ambiente.

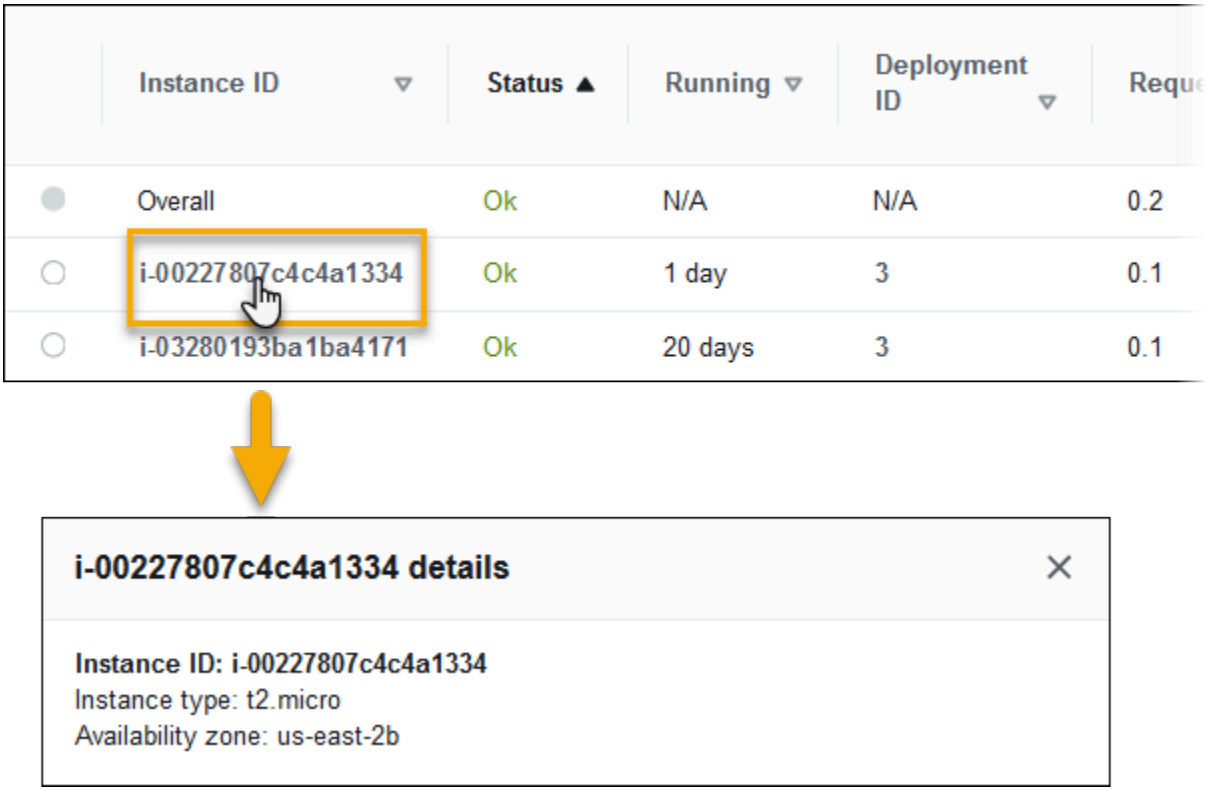
A página exibe muitos detalhes em uma tabela muito ampla. Para ocultar algumas das colunas, escolha



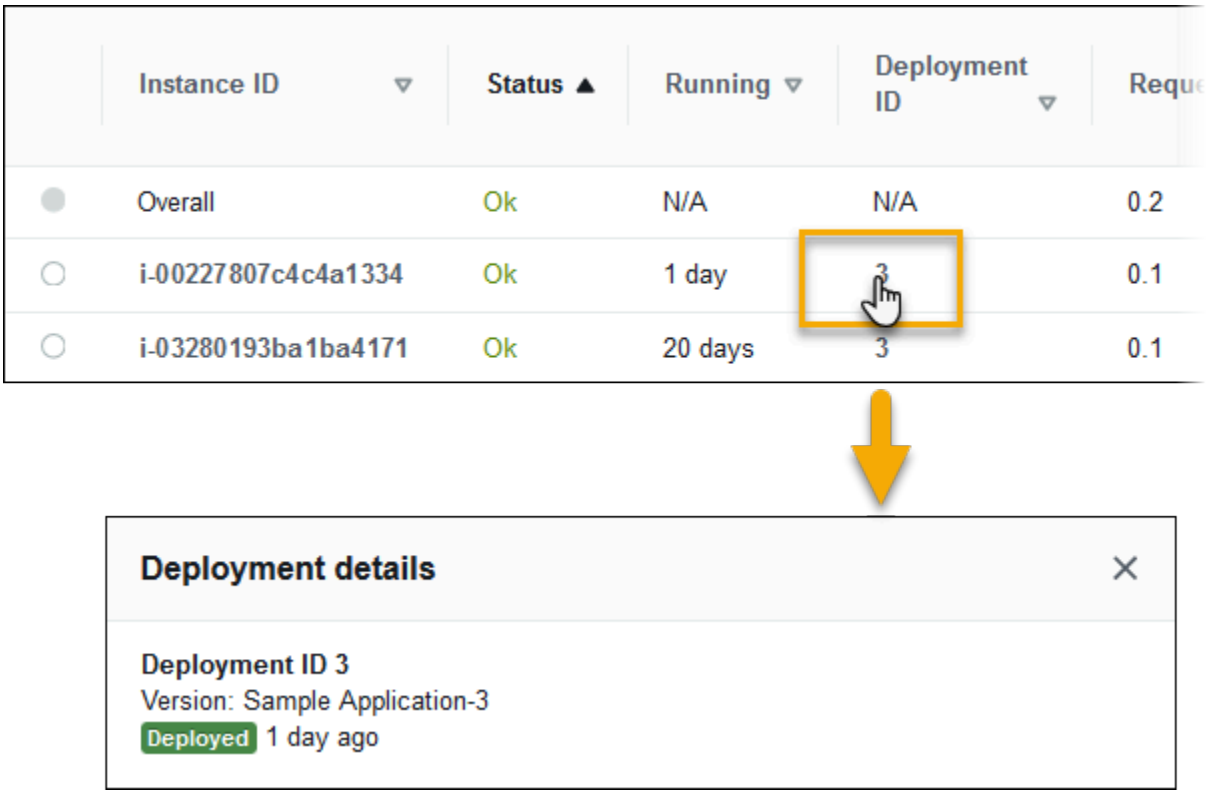
(Preferences (Preferências)). Selecione ou desmarque os nomes das colunas e escolha Confirm (Confirmar).



Escolha o Instance ID (ID da instância) de qualquer instância para exibir mais informações sobre a instância, incluindo a zona de disponibilidade e o tipo de instância.



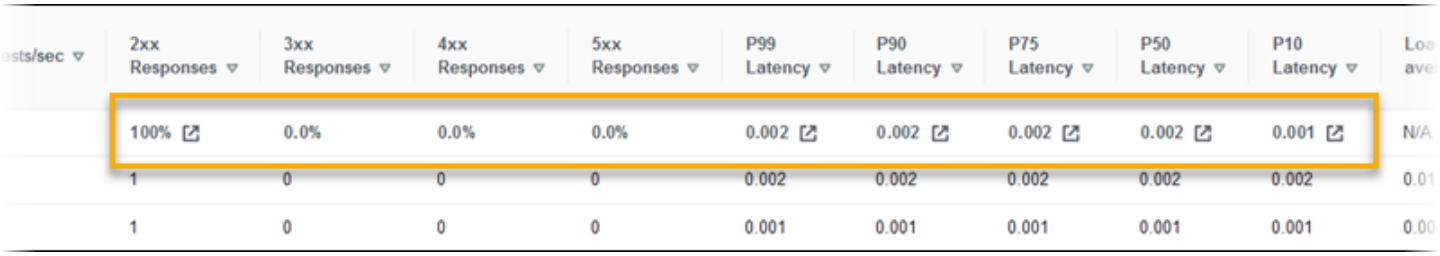
Escolha o Deployment ID (ID de implantação) de qualquer instância para exibir informações sobre a última implantação na instância.



As informações de implantação incluem:

- ID da implantação: o identificador exclusivo da implantação. IDs A implantação começa em 1 e aumenta em um cada vez que você implanta uma nova versão do aplicativo ou altera as configurações que afetam o software ou o sistema operacional em execução nas instâncias do seu ambiente.
- Versão: o rótulo da versão do código-fonte da aplicação usada na implantação.
- Status: o status da implantação, que pode ser `In Progress`, `Deployed` ou `Failed`.
- Horário: para implantações em andamento, o horário em que a implantação começou. Para implantações concluídas, o horário em que a implantação terminou.

Se você habilitar a integração do X-Ray em seu ambiente e instrumentar seu aplicativo com o AWS X-Ray SDK, a página Health adicionará links para o AWS X-Ray console na linha de visão geral.



sts/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	P50 Latency	P10 Latency	Lo... 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

Escolha um link para ver os traços relacionados à estatística destacada no AWS X-Ray console.

Página de monitoramento

A página de monitoramento exibe estatísticas resumidas e gráficos das CloudWatch métricas personalizadas da Amazon geradas pelo sistema aprimorado de relatórios de saúde. Consulte Monitorando a integridade do ambiente no console AWS de gerenciamento para obter instruções sobre como adicionar gráficos e estatísticas a essa página.

Status e cores de integridade

Relatórios de integridade aprimorada representam a integridade geral do ambiente e da instância usando quatro cores, semelhantes aos relatórios de integridade básica. Os relatórios de integridade aprimorada também fornecem sete status de integridade, que são descritores de palavra única que fornecem uma melhor indicação do estado do ambiente.

Status da instância e status do ambiente

Toda vez que o Elastic Beanstalk executa uma verificação de integridade no ambiente, os relatórios de integridade aprimorada verificam a integridade de cada instância do ambiente, analisando todos os dados disponíveis. Se alguma verificação de nível inferior falhar, o Elastic Beanstalk fará downgrade da integridade da instância.

O Elastic Beanstalk exibe as informações de integridade do ambiente geral (cor, status e causa) no console de gerenciamento do ambiente. Essas informações também estão disponíveis na EB CLI. As mensagens de causa e status de integridade de instâncias individuais são atualizadas a cada dez segundos e estão disponíveis a partir da CLI do EB quando você visualiza o status de integridade com eb health.

O Elastic Beanstalk usa as alterações de integridade da instância para avaliar a integridade do ambiente, mas não altera imediatamente o status de integridade do ambiente. Quando uma instância não passa nas verificações de integridade pelo menos três vezes no período de um minuto, o Elastic Beanstalk pode fazer downgrade da integridade do ambiente. Dependendo do número de instâncias

no ambiente e do problema identificado, uma instância não íntegra pode fazer com que o Elastic Beanstalk exiba uma mensagem informativa ou altere o status de integridade do ambiente de verde (OK) para amarelo (Warning (Aviso)) ou vermelho (Degraded (Reduzido) ou Severe (Grave)).

OK (verde)

Esse status é exibido quando:

- Uma instância está passando nas verificações de integridade e o agente de integridade não está encontrando nenhum problema.
- A maioria das instâncias do ambiente está passando nas verificações de integridade e o agente de integridade não está encontrando problemas importantes.
- Uma instância está passando nas verificações de integridade e está concluindo as solicitações normalmente.

Exemplo: o ambiente foi implantado recentemente e está recebendo solicitações normalmente. Cinco por cento das solicitações estão retornando erros da série 400. Implantação concluída normalmente em cada instância.

Mensagem (instância): implantação do aplicativo concluída há 23 segundos e levou 26 segundos.

Aviso (amarelo)

Esse status é exibido quando:

- O agente de integridade está encontrando um número moderado de falhas de solicitação ou outros problemas em uma instância ou ambiente.
- Uma operação está em andamento em uma instância e está demorando muito.

Exemplo: uma instância no ambiente tem um status de Severe (Grave).

Mensagem (ambiente): serviços com problemas em 1 de 5 instâncias.

Reduzido (vermelho)

Esse status é exibido quando o agente de integridade está encontrando um número alto de falhas de solicitação ou outros problemas em uma instância ou ambiente.

Exemplo: o ambiente está no processo de aumentar para até 5 instâncias.

Mensagem (ambiente): 4 instâncias ativas estão abaixo do tamanho mínimo 5 do grupo de Auto Scaling.

Grave (vermelho)

Esse status é exibido quando o agente de integridade está encontrando um número muito alto de falhas de solicitação ou outros problemas em uma instância ou ambiente.

Exemplo: o Elastic Beanstalk não consegue entrar em contato com o balanceador de carga para obter a integridade da instância.

Mensagem (ambiente): a integridade do ELB está falhando ou não está disponível para todas as instâncias. Nenhuma das instâncias está enviando dados. Não é possível assumir a função “arn:aws:iam: :123456789012:role/”. aws-elasticbeanstalk-service-role Verifique se a função existe e se está configurada corretamente.

Mensagem (instâncias): a integridade do ELB da instância não está disponível há 37 minutos. Não há dados. Visto pela última vez 37 minutos atrás.

Informações (verde)

Esse status é exibido quando:

- Uma operação está em andamento em uma instância.
- Uma operação está em andamento em várias instâncias de um ambiente.

Exemplo: uma nova versão do aplicativo está sendo implantada nas instâncias em execução.

Mensagem (ambiente): o comando está sendo executado em 3 de 5 instâncias.

Mensagem (instância): realizando a implantação do aplicativo (em execução por 3 segundos).

Pendente (cinza)

Esse status é exibido quando uma operação está em andamento em uma instância dentro do tempo limite do comando.

Exemplo: você criou recentemente o ambiente e as instâncias estão sendo inicializadas.

Mensagem: realizando a inicialização (em execução por 12 segundos).

Desconhecido (cinza)

Esse status é exibido quando o Elastic Beanstalk e o agente de integridade estão relatando uma quantidade insuficiente de dados em uma instância.

Exemplo: não há dados sendo recebidos.

Suspenso (cinza)

Esse status é exibido quando o Elastic Beanstalk interrompe o monitoramento da integridade do ambiente. O ambiente pode não funcionar corretamente. Se algumas condições de integridade graves durarem muito tempo, o Elastic Beanstalk fará a transição do ambiente para o status Suspended (Suspenso).

Exemplo: o Elastic Beanstalk não pode acessar a função de serviçodo ambiente.

Exemplo: o grupo de Auto Scaling criado pelo Elastic Beanstalk para o ambiente foi excluído.

Mensagem: a integridade do ambiente mudou de OK para Severe (Grave). Não há instâncias listadas. A capacidade desejada do grupo do Auto Scaling está definida como 1.

Métricas de instância

As métricas de instância fornecem informações sobre a integridade das instâncias em seu ambiente. O agente de integridade do Elastic Beanstalk é executado em cada instância. Ele reúne e transmite novamente métricas sobre instâncias para o Elastic Beanstalk, que as analisa para determinar a integridade das instâncias em seus ambientes.

O agente de integridade do Elastic Beanstalk na instância reúne métricas sobre instâncias de servidores Web e o sistema operacional. Para obter informações sobre o servidor Web em plataformas Linux, o Elastic Beanstalk lê e analisa os logs do servidor Web. Na plataforma Windows Server, o Elastic Beanstalk recebe essas informações diretamente do servidor Web do IIS. Os servidores web fornecem informações sobre solicitações HTTP recebidas: quantas solicitações chegaram, quantas resultaram em erros e quanto tempo levaram para serem resolvidas. O sistema operacional fornece informações do snapshot sobre o estado dos recursos das instâncias: a carga de CPU e a distribuição de tempo gasto em cada tipo de processo.

O agente de integridade reúne métricas de sistema operacional e servidor Web e as transmite novamente para o Elastic Beanstalk a cada 10 segundos. O Elastic Beanstalk analisa os dados e usa os resultados para atualizar o status de integridade de cada instância e o ambiente.

Tópicos

- Métricas do servidor web
- Métricas do sistema operacional
- Métricas do servidor web capturadas no IIS no Windows Server

Métricas do servidor web

Em plataformas baseadas no Linux, o agente de integridade do Elastic Beanstalk lê métricas do servidor Web nos logs gerados pelo contêiner da Web ou pelo servidor que processa solicitações em cada instância no ambiente. As plataformas do Elastic Beanstalk são configuradas para gerar dois logs: um no formato legível por humanos e um no formato legível por máquina. O agente de integridade transmite logs legíveis por máquina para o Elastic Beanstalk a cada 10 segundos.

Para obter mais informações sobre o formato de log usado pelo Elastic Beanstalk, consulte Formato de log de integridade aprimorado.

Na plataforma do Windows Server, o Elastic Beanstalk adiciona um módulo ao pipeline de solicitações do servidor Web do IIS e captura as métricas sobre os tempos de solicitação HTTP e os códigos de resposta. O módulo envia essas métricas para o agente de integridade na instância usando um canal IPC (comunicação entre processos) de alta performance. Para obter mais detalhes sobre a implementação, visite Métricas do servidor web capturadas no IIS no Windows Server.

Métricas do servidor web relatadas

`RequestCount`

Número de solicitações atendidas pelo servidor Web por segundo nos últimos 10 segundos. Mostrado como uma média `r/sec` (solicitações por segundo) na CLI do EB e Página de integridade do ambiente.

`Status2xx`, `Status3xx`, `Status4xx`, `Status5xx`

Número de solicitações que resultaram em cada tipo de código de status nos últimos 10 segundos. Por exemplo, as solicitações bem-sucedidas retornam um 200 OK, redirecionamentos são 301, e se o URL inserido não corresponder a nenhum recurso no aplicativo, retornará 404.

A CLI do EB e Página de integridade do ambiente mostram essas métricas como um número bruto de solicitações de instâncias e como uma porcentagem do total de solicitações de ambientes.

`p99.9`, `p99`, `p95`, `p90`, `p85`, `p75`, `p50`, `p10`

A latência média para x % mais lento de solicitações nos últimos 10 segundos, em que x é a diferença entre o número e 100. Por exemplo, `p99 1.403` indica que 1% das solicitações mais lentas nos últimos 10 segundos tiveram uma latência média de 1,403 segundos.

Métricas do sistema operacional

O agente de integridade do Elastic Beanstalk relata as métricas a seguir do sistema operacional. O Elastic Beanstalk usa essas métricas para identificar instâncias que estão sob carga pesada constante. As métricas diferem por sistema operacional.

Métricas do sistema operacional relatadas — Linux

`Running`

O tempo que passou desde que a instância foi iniciada.

`Load 1`, `Load 5`

Média de carga nos últimos períodos de um minuto e cinco minutos. Mostrado como um valor decimal indicando o número médio de processos em execução durante esse período. Se o número mostrado for maior que o número de v CPUs (threads) disponíveis, o restante será o número médio de processos que estavam aguardando.

Por exemplo, se seu tipo de instância tiver quatro v CPUs e a carga for 4,5, haverá uma média de 0,5 processos em espera durante esse período, o equivalente a um processo aguardando 50% do tempo.

`User %`, `Nice %`, `System %`, `Idle %`, `I/O Wait %`

Porcentagem de tempo que a CPU passou em cada estado nos últimos 10 segundos.

Métricas do sistema operacional relatadas — Windows

`Running`

O tempo que passou desde que a instância foi iniciada.

`% User Time`, `% Privileged Time`, `% Idle Time`

Porcentagem de tempo que a CPU passou em cada estado nos últimos 10 segundos.

Métricas do servidor web capturadas no IIS no Windows Server

Na plataforma do Windows Server, o Elastic Beanstalk adiciona um módulo ao pipeline de solicitações do servidor Web do IIS e captura as métricas sobre os tempos de solicitação HTTP e os códigos de resposta. O módulo envia essas métricas para o agente de integridade na instância usando um canal IPC (comunicação entre processos) de alta performance. O agente de integridade agrega essas métricas, combina-as com as métricas do sistema operacional e as envia para o serviço do Elastic Beanstalk.

Detalhes da implantação

Para capturar as métricas do IIS, o Elastic Beanstalk implementa um `IHttpModule` gerenciado e assina os eventos `BeginRequest` e `EndRequest`. Isso permite que o módulo relate a latência das solicitações HTTP e os códigos de resposta de todas as solicitações da web atendidas pelo IIS. Para adicionar o módulo ao pipeline de solicitações do IIS, o Elastic Beanstalk registra o módulo na seção `<modules>` do arquivo de configuração do IIS, `%windir%\System32\inetsrv\config\applicationHost.config`.

O módulo do Elastic Beanstalk no IIS envia as métricas capturadas da solicitação da Web ao agente de integridade na instância, que é um serviço Windows chamado `HealthD`. Para enviar esses dados, o módulo usa `NetNamedPipeBinding`, que fornece uma vinculação segura e confiável que é otimizada para a comunicação na máquina.

Configurar as regras de integridade aperfeiçoada de um ambiente

AWS Elastic Beanstalk os relatórios de integridade aprimorados dependem de um conjunto de regras para determinar a integridade do seu ambiente. Algumas dessas regras podem não ser adequadas para o seu aplicativo. Veja a seguir alguns exemplos comuns:

- Você usa ferramentas de teste no lado do cliente. Nesse caso, são esperados erros frequentes de cliente HTTP (4xx).
- Use o AWS WAF junto com o Application Load Balancer do ambiente para bloquear tráfego de entrada indesejado. Nesse caso, o Application Load Balancer retorna HTTP 403 para cada mensagem recebida rejeitada.

Por padrão, o Elastic Beanstalk inclui todos os erros HTTP 4xx da aplicação ao determinar a integridade do ambiente. Ele altera o status de integridade do ambiente de OK para Warning (Aviso), Degraded (Degradado) ou Severe (Grave), dependendo da taxa de erros. Para lidar corretamente

com casos como os exemplos que mencionamos, o Elastic Beanstalk permite configurar algumas regras de integridade aprimorada. É possível optar por ignorar os erros HTTP 4xx do aplicativo nas instâncias do ambiente ou ignorar os erros HTTP 4xx retornados pelo load balancer do ambiente. Este tópico descreve como fazer essas alterações de configuração.

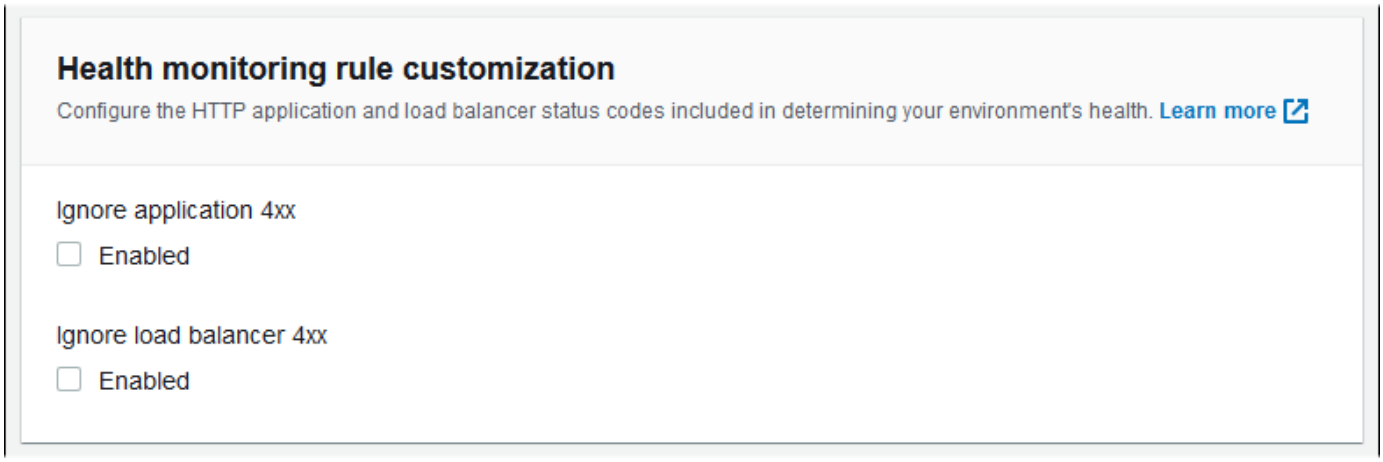
> **Note**
>
> No momento, essas são as únicas personalizações de regras de integridade avançada disponíveis. Não é possível configurar a integridade avançada para ignorar outros erros HTTP além do 4xx.

Configurar regras de integridade aprimorada usando o console do Elastic Beanstalk

É possível usar o console do Elastic Beanstalk para configurar regras de integridade aprimorada no ambiente.

Como configurar a verificação de código de status HTTP 4xx usando o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health monitoring rule customization (Personalização da regra de monitoramento de integridade), habilite ou desabilite as opções Ignore (Ignorar) desejadas.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configurar regras de integridade aprimorada usando a EB CLI

Use a CLI do EB para configurar regras de integridade aprimorada, salvando a configuração do ambiente localmente, adicionando uma entrada que defina essas regras e fazendo upload da configuração no Elastic Beanstalk. Você pode aplicar a configuração salva a um ambiente durante ou após a criação.

Como configurar a verificação de código de status HTTP 4xx usando a CLI do EB e configurações salvas

1. Inicialize a pasta do projeto com eb init.
2. Crie um ambiente executando o comando eb create.
3. Salve um modelo de configuração localmente, executando o comando eb config save. O exemplo a seguir usa a opção `--cfg` para especificar o nome da configuração.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-
state.cfg.yml
```

4. Abra o arquivo de configuração salvo em um editor de texto.
5. Em `OptionSettings` > `aws:elasticbeanstalk:healthreporting:system:`, adicione uma chave `ConfigDocument` para listar cada regra de integridade aprimorada a ser configurada. O `ConfigDocument` a seguir desabilita a verificação dos códigos de status HTTP 4xx do aplicativo, mantendo a verificação do código HTTP 4xx do load balancer habilitada.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      Rules:
        Environment:
          Application:
            ApplicationRequests4xx:
              Enabled: false
          ELB:
            ELBRequests4xx:
              Enabled: true
      Version: 1
    SystemType: enhanced
...
```

> **Note**
>
> É possível combinar `Rules` e `CloudWatchMetrics` na mesma configuração da opção `ConfigDocument`. `CloudWatchMetrics` são descritas em Publicação de métricas CloudWatch personalizadas da Amazon para um ambiente.
> Se você tiver habilitado `CloudWatchMetrics` anteriormente, o arquivo de configuração recuperado usando o comando eb config save já terá uma chave `ConfigDocument` com uma seção `CloudWatchMetrics`. Não o exclua, em vez disso, adicione uma seção `Rules` no mesmo valor da opção `ConfigDocument`.

6. Salve o arquivo de configuração e feche o editor de texto. Neste exemplo, o arquivo de configuração atualizado é salvo com um nome (`02-cloudwatch-enabled.cfg.yml`) diferente do arquivo de configuração obtido por download. Isso cria uma configuração salva separada quando o arquivo é carregado. Você pode usar o mesmo nome que o arquivo baixado para substituir a configuração existente, sem criar um novo.
7. Use o comando eb config put para fazer upload do arquivo de configuração atualizado no Elastic Beanstalk.

   ```
   $ eb config put 02-cloudwatch-enabled
   ```

   Ao usar os comandos eb config, `get` e `put` com configurações salvas, não inclua a extensão do nome do arquivo.
8. Aplique a configuração salva ao ambiente em execução.

   ```
   $ eb config --cfg 02-cloudwatch-enabled
   ```

   A opção `--cfg` especifica um arquivo de configuração nomeado que é aplicado ao ambiente. É possível salvar o arquivo de configuração localmente ou no Elastic Beanstalk. Se um arquivo de configuração com o nome especificado já existir em ambos os locais, a EB CLI usará o arquivo local.

Configurar regras de integridade aprimorada usando um documento de configuração

O documento de configuração (config) para regras de integridade aprimorada é um documento JSON que lista as regras a serem configuradas.

O exemplo a seguir mostra um documento de configuração que desabilita a verificação de códigos de status HTTP 4xx do aplicativo e habilita a verificação de códigos de status HTTP 4xx do load balancer.

```
{
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    }
  },
  "Version": 1
}
```

Para o AWS CLI, você passa o documento como um valor para a `Value` chave em um argumento de configurações de opção, que por si só é um objeto JSON. Nesse caso, você deve escapar as aspas no documento incorporado. O comando a seguir verifica se as definições de configuração são válidas.

```
$ 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 }"
    }
]'
```

Para um arquivo de configuração `.ebextensions` em YAML, você pode fornecer o documento JSON no estado em que está.

```
  option_settings:
    - namespace: aws:elasticbeanstalk:healthreporting:system
      option_name: ConfigDocument
      value: {
  "Rules": {
    "Environment": {
      "Application": {
        "ApplicationRequests4xx": {
          "Enabled": false
        }
      },
      "ELB": {
        "ELBRequests4xx": {
          "Enabled": true
        }
      }
    }
  },
  "Version": 1
}
```

Publicação de métricas CloudWatch personalizadas da Amazon para um ambiente

Você pode publicar os dados coletados por meio de relatórios de saúde AWS Elastic Beanstalk aprimorados na Amazon CloudWatch como métricas personalizadas. A publicação de métricas CloudWatch permite monitorar as mudanças no desempenho do aplicativo ao longo do tempo e identificar possíveis problemas rastreando como o uso de recursos e a latência da solicitação aumentam com a carga.

Ao publicar métricas em CloudWatch, você também as disponibiliza para uso com gráficos de monitoramento e alarmes. Uma métrica gratuita, EnvironmentHealth, é ativada automaticamente quando você usa relatórios de saúde aprimorados. Métricas personalizadas que não EnvironmentHealthsejam CloudWatch cobranças padrão.

Para publicar métricas CloudWatch personalizadas para um ambiente, você deve primeiro habilitar relatórios de integridade aprimorados sobre o ambiente. Para obter instruções, consulte Habilitar relatórios de integridade aprimorada do Elastic Beanstalk.

Tópicos

- Métricas de relatórios de integridade aprimorada
- Configurando CloudWatch métricas usando o console do Elastic Beanstalk
- Configurando métricas CloudWatch personalizadas usando o EB CLI
- Fornecer documentos Config de métrica personalizada

Métricas de relatórios de integridade aprimorada

Quando você ativa relatórios de integridade aprimorados em seu ambiente, o sistema aprimorado de relatórios de integridade publica automaticamente uma métrica CloudWatch personalizada, EnvironmentHealth. Para publicar métricas adicionais CloudWatch, configure seu ambiente com essas métricas usando o console do Elastic Beanstalk, EB CLI ou .ebextensions.

Você pode publicar as seguintes métricas de saúde aprimoradas do seu ambiente para CloudWatch.

Métricas disponíveis — todas as plataformas

`EnvironmentHealth`

Ambiente apenas. Essa é a única CloudWatch métrica que o sistema aprimorado de relatórios de saúde publica, a menos que você configure métricas adicionais. A integridade do ambiente é representada por um dos sete status. No CloudWatch console, esses status são mapeados para os seguintes valores:

- 0: OK
- 1: Informações
- 5: Desconhecido
- 10: Sem dados
- 15: Aviso
- 20: Reduzido
- 25: Grave

`InstancesSevere`, `InstancesDegraded`, `InstancesWarning`, `InstancesInfo`, `InstancesOk`, `InstancesPending`, `InstancesUnknown`, `InstancesNoData`

Ambiente apenas. Essas métricas indicam o número de instâncias no ambiente com cada status de integridade. `InstancesNoData` indica o número de instâncias para as quais não há dados recebidos.

`ApplicationRequestsTotal`, `ApplicationRequests5xx`, `ApplicationRequests4xx`, `ApplicationRequests3xx`, `ApplicationRequests2xx`

Instância e ambiente. Indica o número total de solicitações concluídas pela instância ou ambiente, e o número de solicitações que foram concluídas com cada categoria de código de status.

`ApplicationLatencyP10`, `ApplicationLatencyP50`, `ApplicationLatencyP75`, `ApplicationLatencyP85`, `ApplicationLatencyP90`, `ApplicationLatencyP95`, `ApplicationLatencyP99`, `ApplicationLatencyP99.9`

Instância e ambiente. Indica o tempo médio, em segundos, que leva para concluir com mais rapidez x % de solicitações.

`InstanceHealth`

Instância apenas. Indica o status da integridade atual da instância. A integridade da instância é representada por um dos sete status. No CloudWatch console, esses status são mapeados para os seguintes valores:

- 0: OK
- 1: Informações
- 5: Desconhecido
- 10: Sem dados
- 15: Aviso
- 20: Reduzido
- 25: Grave

Métricas disponíveis — Linux

`CPUIrq`, `CPUIdle`, `CPUUser`, `CPUSystem`, `CPUSoftirq`, `CPUIowait`, `CPUNice`

Instância apenas. Indica a porcentagem de tempo que a CPU passou em cada estado no último minuto.

`LoadAverage1min`

Instância apenas. A carga média de CPU da instância durante o último minuto.

`RootFilesystemUtil`

Instância apenas. Indica a porcentagem do espaço em disco em uso.

Métricas disponíveis — Windows

`CPUIdle`, `CPUUser`, `CPUPrivileged`

Instância apenas. Indica a porcentagem de tempo que a CPU passou em cada estado no último minuto.

Configurando CloudWatch métricas usando o console do Elastic Beanstalk

Você pode usar o console do Elastic Beanstalk para configurar seu ambiente para publicar CloudWatch métricas aprimoradas de relatórios de saúde e disponibilizá-las para uso com gráficos de monitoramento e alarmes.

Para configurar métricas CloudWatch personalizadas no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatórios de integridade), selecione as métricas de instância e de ambiente a serem publicadas no CloudWatch. Para selecionar várias métricas, mantenha pressionada a tecla Ctrl enquanto escolhe.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

A ativação de métricas CloudWatch personalizadas as adiciona à lista de métricas disponíveis na página Monitoramento.

Configurando métricas CloudWatch personalizadas usando o EB CLI

Use a CLI do EB para configurar métricas personalizadas, salvando a configuração do ambiente localmente, adicionando uma entrada que define as métricas para publicação e fazendo upload da configuração no Elastic Beanstalk. Você pode aplicar a configuração salva a um ambiente durante ou após a criação.

Para configurar métricas CloudWatch personalizadas com a CLI do EB e configurações salvas

1. Inicialize a pasta do projeto com eb init.

2. Crie um ambiente executando o comando eb create.
3. Salve um modelo de configuração localmente, executando o comando eb config save. O exemplo a seguir usa a opção `--cfg` para especificar o nome da configuração.

```
$ eb config save --cfg 01-base-state
Configuration saved at: ~/project/.elasticbeanstalk/saved_configs/01-base-
state.cfg.yml
```

4. Abra o arquivo de configuração salvo em um editor de texto.
5. Em `OptionSettings` >`aws:elasticbeanstalk:healthreporting:system:`, adicione uma `ConfigDocument` chave para ativar cada uma das CloudWatch métricas desejadas. Por exemplo, `ConfigDocument` publica as métricas `ApplicationRequests5xx` e `ApplicationRequests4xx` no nível de ambiente e as métricas `ApplicationRequestsTotal` no nível de instância.

```
OptionSettings:
  ...
  aws:elasticbeanstalk:healthreporting:system:
    ConfigDocument:
      CloudWatchMetrics:
        Environment:
          ApplicationRequests5xx: 60
          ApplicationRequests4xx: 60
        Instance:
          ApplicationRequestsTotal: 60
      Version: 1
    SystemType: enhanced
...
```

No exemplo, 60 indica o número de segundos entre as medidas. No momento, este é o único valor aceito.

> **Note**
>
> É possível combinar `CloudWatchMetrics` e `Rules` na mesma configuração da opção `ConfigDocument`. `Rules` são descritas em Configurar as regras de integridade aperfeiçoada de um ambiente.
> Se você tiver usado `Rules` anteriormente para configurar as regras de integridade aprimorada, o arquivo de configuração recuperado por meio do comando eb config

save já terá uma chave `ConfigDocument` com uma seção `Rules`. Não o exclua, em vez disso, adicione uma seção `CloudWatchMetrics` no mesmo valor da opção `ConfigDocument`.

6. Salve o arquivo de configuração e feche o editor de texto. Neste exemplo, o arquivo de configuração atualizado é salvo com um nome (`02-cloudwatch-enabled.cfg.yml`) que seja diferente do arquivo de configuração que foi baixado. Isso cria uma configuração salva separada quando o arquivo é carregado. Você pode usar o mesmo nome que o arquivo baixado para substituir a configuração existente, sem criar um novo.
7. Use o comando eb config put para fazer upload do arquivo de configuração atualizado no Elastic Beanstalk.

   ```
   $ eb config put 02-cloudwatch-enabled
   ```

   Ao usar os comandos eb config `get` e `put` com configurações salvas, não inclua a extensão de arquivo.
8. Aplique a configuração salva ao ambiente em execução.

   ```
   $ eb config --cfg 02-cloudwatch-enabled
   ```

   A opção `--cfg` especifica um arquivo de configuração nomeado que é aplicado ao ambiente. É possível salvar o arquivo de configuração localmente ou no Elastic Beanstalk. Se um arquivo de configuração com o nome especificado já existir em ambos os locais, a EB CLI usará o arquivo local.

Fornecer documentos Config de métrica personalizada

O documento de configuração (config) das métricas CloudWatch personalizadas da Amazon é um documento JSON que lista as métricas a serem publicadas nos níveis de ambiente e instância. O exemplo a seguir mostra um documento config que habilita todas as métricas disponíveis no 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
}
```

Para o AWS CLI, você passa o documento como um valor para a `Value` chave em um argumento de configurações de opção, que por si só é um objeto JSON. Nesse caso, você deve escapar as aspas no documento incorporado.

```
$ 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}"
    }
]'
```

Para um arquivo de configuração `.ebextensions` em YAML, você pode fornecer o documento JSON no estado em que está.

```
  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
}
```

Usar relatórios de integridade aprimorada com a API do Elastic Beanstalk

Como os relatórios de integridade AWS Elastic Beanstalk aprimorados têm requisitos de função e pilha de soluções, você deve atualizar os scripts e o código usados antes do lançamento dos relatórios de integridade aprimorados antes de poder usá-los. Para manter a compatibilidade com versões anteriores, os relatórios de integridade aprimorada não são habilitados por padrão quando você cria um ambiente usando a API do Elastic Beanstalk.

Você configura relatórios de saúde aprimorados definindo a função de serviço, o perfil da instância e as opções de CloudWatch configuração da Amazon para seu ambiente. Você pode fazer isso de três formas: definindo as opções de configuração na pasta `.ebextensions`, com as configurações salvas, ou configurando-as diretamente no parâmetro `create-environment` da chamada `option-settings`.

Para usar a API ou a interface de linha de AWS comando (CLI) para criar um ambiente que ofereça suporte à integridade aprimorada, você deve: SDKs

- Criar uma função de serviço e um perfil da instância com as permissões apropriadas
- Crie um ambiente com uma nova versão da plataforma
- Definir as opções de configuração de tipo de sistema, perfil da instância e função de serviço

Use as seguintes opções de configuração nos namespaces `aws:elasticbeanstalk:healthreporting:system`, `aws:autoscaling:launchconfiguration` e `aws:elasticbeanstalk:environment` para configurar seu ambiente para os relatórios de integridade aprimorada.

Opções de configuração de integridade aprimorada

SystemType

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Para habilitar os relatórios de integridade aprimorada, defina como **enhanced**.

IamInstanceProfile

Namespace: `aws:autoscaling:launchconfiguration`

Defina como o nome de um perfil da instância configurado para uso com o Elastic Beanstalk.

ServiceRole

Namespace: `aws:elasticbeanstalk:environment`

Defina como o nome de uma função de serviço configurada para uso com o Elastic Beanstalk.

ConfigDocument (Opcional)

Namespace: `aws:elasticbeanstalk:healthreporting:system`

Um documento JSON que define as métricas de instância e ambiente nas quais publicar CloudWatch. Por exemplo:

```
{
  "CloudWatchMetrics":
    {
    "Environment":
      {
      "ApplicationLatencyP99.9":60,
      "InstancesSevere":60
      }
    "Instance":
      {
      "ApplicationLatencyP85":60,
      "CPUUser": 60
      }
    }
  "Version":1
}
```

> **Note**
>
> Os documentos de configuração podem exigir formatação especial, como aspas de escape, dependendo de como você os disponibiliza no Elastic Beanstalk. Consulte Fornecer documentos Config de métrica personalizada para ver exemplos.

Formato de log de integridade aprimorado

AWS Elastic Beanstalk as plataformas usam um formato de log de servidor web personalizado para transmitir com eficiência as informações sobre solicitações HTTP para o sistema aprimorado de relatórios de saúde. O sistema analisa os logs, identifica os problemas e define a integridade da instância e do ambiente adequadamente. Se você desabilitar o servidor de proxy da Web no ambiente e atender às solicitações diretamente do contêiner da Web, ainda poderá usar totalmente os relatórios de integridade aprimorada configurando o servidor para enviar logs para o local e no formato usado pelo agente de integridade do Elastic Beanstalk.

> **Note**
>
> As informações desta página são relevantes apenas para plataformas Linux. Na plataforma Windows Server, o Elastic Beanstalk recebe informações sobre solicitações HTTP diretamente do servidor Web do IIS. Para obter detalhes, consulte Métricas do servidor web capturadas no IIS no Windows Server.

Configuração do log de servidor web

As plataformas do Elastic Beanstalk são configuradas para gerar dois logs com informações sobre solicitações HTTP. O primeiro está no formato verboso e fornece informações detalhadas sobre a solicitação, incluindo informações sobre o agente do usuário do solicitante e um carimbo de data e hora legível.

/var/log/nginx/access.log

O exemplo a seguir é de um proxy nginx em execução em um ambiente de servidor da web Ruby, mas o formato é semelhante ao do 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"
```

O segundo log está no formato conciso. Ele inclui informações relevantes apenas para os relatórios de integridade aprimorados. Esse log é gerado em uma subpasta chamada `healthd` e muda de hora em hora. Os logs antigos são excluídos imediatamente após o rodízio.

/var/log/nginx/healthd/application.log.2015-07-23-00

O exemplo a seguir mostra um log no formato legível por máquina.

```
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
```

O formato do log de integridade aprimorado inclui as seguintes informações:

- A hora da solicitação, no horário do Unix.
- O caminho da solicitação
- O código de status HTTP para o resultado
- A hora da solicitação
- A hora de upstream
- O cabeçalho HTTP `X-Forwarded-For`

Para os proxies nginx, os horários são impressos em segundos de ponto flutuante, com três casas decimais. Para o Apache, são usados microssegundos inteiros.

> **Note**
>
> Se você vir um aviso semelhante ao seguinte em um arquivo de log, em que `DATE-TIME` é uma data e hora, e estiver usando um proxy personalizado, como em um ambiente do Docker de vários contêineres, você deverá usar um .ebextension para configurar o ambiente a fim que de `healthd` possa ler os arquivos de log:
>
> ```
> W, [DATE-TIME #1922] WARN -- : log file "/var/log/nginx/healthd/
> application.log.DATE-TIME" does not exist
> ```
>
> É possível começar com .ebextension no Docker de vários contêineres demonstrativo.

/etc/nginx/conf.d/webapp_healthd.conf

O exemplo a seguir mostra a configuração do log para nginx com o formato de log `healthd` destacado.

```
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

O exemplo a seguir mostra a configuração do log para 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
```

Gerar logs para a geração de relatórios de integridade aprimorados

Para fornecer logs para o agente de integridade, faça o seguinte:

- Gere os logs no formato correto, como mostrado na seção anterior
- Gere logs em `/var/log/nginx/healthd/`
- Nomeie os logs usando o seguinte formato: `application.log.$year-$month-$day-$hour`
- Faça rodízio dos logs uma vez por hora

- Não trunque os logs

Notificações e solução de problemas

Esta página lista exemplos de mensagens da causa de problemas comuns e links para mais informações. As mensagens da causa são exibidas na página de visão geral do ambiente no console do Elastic Beanstalk e são registradas nos eventos quando problemas de integridade persistem em várias verificações.

Implantações

O Elastic Beanstalk monitora a consistência do ambiente após as implantações. Se uma implantação contínua falhar, a versão do aplicativo em execução nas instâncias de seu ambiente pode variar. Isso poderá ocorrer se uma implantação for bem-sucedida em um ou mais lotes, mas falhar antes da conclusão de todos os lotes.

Versão incorreta do aplicativo encontrada em 2 de 5 instâncias. Versão esperada "v1" (implantação 1).

Versão incorreta do aplicativo nas instâncias do ambiente. Versão esperada "v1" (implantação 1).

A versão esperada do aplicativo não está em execução em algumas ou todas as instâncias de um ambiente.

Versão incorreta do aplicativo "v2" (implantação 2). Versão esperada "v1" (implantação 1).

O aplicativo implantado em uma instância não tem a versão esperada. Se uma implantação falhar, a versão esperada será redefinida para a versão da implantação bem-sucedida mais recente. No exemplo acima, a primeira implantação (versão "v1") foi bem-sucedida, mas a segunda falhou (versão "v2"). Todas as instâncias que executam "v2" não são consideradas íntegras.

Para resolver esse problema, inicie outra implantação. Você pode reimplantar uma versão anterior que você sabe que funciona, ou configurar seu ambiente para ignorar verificações de integridade durante a implantação e reimplantar a nova versão para forçar a conclusão da implantação.

Você também pode identificar e encerrar as instâncias que estão executando a versão incorreta do aplicativo. O Elastic Beanstalk iniciará instâncias com a versão correta para substituir as instâncias encerradas. Use o comando de integridade da EB CLI para identificar as instâncias que estão executando a versão incorreta do aplicativo.

Servidor de aplicativos

15% das solicitações estão com erro de HTTP 4xx

20% das solicitações para o ELB estão com erro de HTTP 4xx.

Uma alta porcentagem de solicitações HTTP para uma instância ou um ambiente está falhando com erros 4xx.

Um código de status da série 400 indica que o usuário fez uma solicitação incorreta, como solicitar uma página que não existe (404 Arquivo não encontrado) ou que o usuário não tem acesso (403 Proibido). Um pequeno número de 404s não é incomum, mas um grande número pode significar que existem links internos ou externos para páginas indisponíveis. Esses problemas podem ser resolvidos ao corrigir os links internos inválidos e ao adicionar redirecionamentos de links externos inválidos.

5% das solicitações estão falhando com HTTP 5xx

3% das solicitações para o ELB estão falhando com HTTP 5xx.

Uma alta porcentagem de solicitações HTTP para uma instância ou um ambiente está falhando com códigos de status da série 500.

Uma código de status da série 500 indica que o servidor de aplicativos detectou um erro interno. Esses problemas indicam que há um erro no código do aplicativo que deve ser identificado e corrigido rapidamente.

95% da CPU está em uso

Em uma instância, o agente de integridade está relatando uma porcentagem extremamente alta de uso da CPU e define a integridade da instância como Aviso ou Reduzido.

Escale o ambiente para retirar carga das instâncias.

Instância do operador

20 mensagens em espera na fila (25 segundos atrás)

As solicitações estão sendo adicionadas à fila do ambiente do operador com mais rapidez do que podem ser processadas. Escale o ambiente para aumentar a capacidade.

5 mensagens em Dead Letter Queue (15 segundos atrás)

As solicitações de operador estão falhando repetidamente e sendo adicionadas à the section called “Filas de mensagens não entregues”. Verifique o motivo da falha das solicitações na fila de mensagens mortas.

Outros recursos

4 instâncias ativas estão abaixo do tamanho mínimo 5 do grupo de Auto Scaling

O número de instâncias em execução no ambiente é menor do que o mínimo configurado para o grupo de Auto Scaling.

As notificações do grupo de Auto Scaling (groupname) foram excluídas ou modificadas

As notificações configuradas para o seu grupo de Auto Scaling foram modificadas fora do Elastic Beanstalk.

Gerenciar alarmes

Este tópico orienta pelas etapas para criar alarmes para métricas que você está monitorando. Ele também fornece instruções para visualizar alarmes existentes e verificar o estado deles.

É possível criar alarmes para as métricas que você está monitorando usando o console do Elastic Beanstalk. Os alarmes ajudam você a monitorar as mudanças em seu AWS Elastic Beanstalk ambiente para que você possa identificar e mitigar facilmente os problemas antes que eles ocorram. Por exemplo, defina um alarme que notifique quando o uso da CPU no ambiente exceder um determinado limite, garantindo que você seja notificado antes que ocorra um problema em potencial. Para obter mais informações, consulte Usando o Elastic Beanstalk com a Amazon CloudWatch.

> **Note**
>
> O Elastic CloudWatch Beanstalk usa para monitoramento e alarmes CloudWatch , o que significa que os custos são aplicados AWS à sua conta para todos os alarmes que você usa.

Para obter mais informações sobre monitoramento de métricas específicas, consulte Relatórios de integridade básica.

Para verificar o estado dos alarmes

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, selecione Alarmes.

   A página exibe uma lista de alarmes existentes. Se algum alarme estiver no estado de alarme, ele será sinalizado com o ícone de aviso ().
4. Para filtrar alarmes, escolha o menu suspenso e selecione um filtro.
5. Para editar ou excluir um alarme, selecione o ícone de edição () ou o ícone de exclusão (), respectivamente.

Para criar um alarme

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Monitoring (Monitoramento).
4. Localize a métrica para a qual você deseja criar um alarme e, em seguida, selecione o ícone de alarme ().

   A página Add alarm (Adicionar alarme) é exibida.
5. Insira detalhes sobre o alarme:

   - Name: o nome desse alarme.
   - Description (opcional): uma breve descrição do alarme.
   - Period: o intervalo de tempo entre as leituras.
   - Threshold: descreve o comportamento e o valor que a métrica deve exceder para acionar um alarme.
   - Change state after: o tempo após o limite ter sido excedido para acionar uma alteração no estado do alarme.

- Notify (Notificar): o tópico do Amazon SNS que é notificado quando um alarme muda de estado.
- Notify when state changes to:
  - OK: a métrica está dentro do limite definido.
  - Alarm: a métrica excedeu o limite definido.
  - Insufficient data: o alarme acabou de ser acionado, a métrica não está disponível ou não há dados suficientes para a métrica determinar o estado do alarme.

6. Escolha Adicionar. O status do ambiente muda para cinza enquanto o ambiente é atualizado. É possível visualizar o alarme criado escolhendo Alarms (Alarmes) no painel de navegação.

Visualizar o histórico de alterações de um ambiente do Elastic Beanstalk

Este tópico explica como você pode usar o Console do Elastic Beanstalk para visualizar um histórico das alterações de configuração que foram feitas nos seus ambientes Elastic Beanstalk.

O Elastic Beanstalk busca seu histórico de alterações em eventos registrados no AWS CloudTrail e os exibe em uma lista em que você pode navegar e filtrar facilmente.

O painel Change History (Histórico de alterações) exibe as seguintes informações sobre alterações feitas nos seus ambientes:

- A data e a hora em que uma alteração foi feita
- O usuário do IAM que foi responsável por uma alteração feita
- A ferramenta de origem (interface de linhas de comando do Elastic Beanstalk (CLI do EB) ou console) que foi usada para fazer a alteração
- O parâmetro de configuração e os novos valores que foram definidos

Quaisquer dados confidenciais que fazem parte da alteração, como os nomes dos usuários do banco de dados afetados pela alteração, não são exibidos no painel.

Para ver o histórico de alterações

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Change history (Histórico de alterações).

A página Change history (Histórico de alterações) mostra uma lista das alterações de configuração que foram feitas nos seus ambientes do Elastic Beanstalk.

Observe os seguintes pontos sobre como navegar pelas informações desta página:

- Você pode pesquisar a lista escolhendo **<**(previous) (anterior) ou, **>** (next) (próximo), ao selecionar um número de página específico.
- Na coluna Configuration changes (Alterações de configuração), selecione o ícone de seta para alternar entre expandir e recolher a lista de alterações no cabeçalho Changes made (Alterações feitas).
- Use a barra de pesquisa para filtrar os resultados da lista de histórico de alterações. Você pode inserir qualquer string para refinar a lista de alterações que são exibidas.

Observe o seguinte sobre como filtrar os resultados exibidos:

- O filtro de pesquisa não diferencia maiúsculas de minúsculas.
- Você pode filtrar as alterações exibidas com base nas informações da coluna Configuration changes (Alterações de configuração), mesmo quando ela não estiver visível por estar recolhida dentro das alterações feitas.
- Você só pode filtrar os resultados exibidos. No entanto, o filtro permanece no lugar mesmo se você optar por ir para outra página para exibir mais resultados. Seus resultados filtrados também são anexados ao conjunto de resultados da página seguinte.

Os exemplos a seguir demonstram como os dados mostrados na tela anterior podem ser filtrados:

- Digite **`GettingStartedApp-env`** na caixa de pesquisa para restringir os resultados e incluir apenas as alterações feitas no ambiente chamado GettingStartedApp-env.
- Digite **`example3`** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas por usuários do IAM cujo nome de usuário contém a string example3.
- Digite **`2020-10`** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas durante o mês de outubro de 2020. Altere o valor da pesquisa **`2020-10-16`** para filtrar ainda mais os resultados exibidos para incluir apenas as alterações feitas no dia 16 de outubro de 2020.
- Digite **`proxy:staticfiles`** na caixa de pesquisa para refinar os resultados para incluir apenas as alterações feitas no namespace denominado aws:elasticbeanstalk:environment:proxy:staticfiles.

As linhas exibidas são o resultado do filtro. Isso é verdade mesmo para resultados que são recolhidos em Changes made (Alterações feitas).

Visualizar o stream de eventos de um ambiente do Elastic Beanstalk

Este tópico explica como acessar eventos e notificações associados à sua aplicação.

Visualizar eventos com o console do Elastic Beanstalk

Para visualizar eventos com o console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Eventos.

   A página de eventos exibe uma lista de todos os eventos que foram registrados para o ambiente. É possível percorrer a lista escolhendo < (anterior), > (próximo) ou números de página. É possível filtrar por tipo de evento usando a lista suspensa Severity (Gravidade).

Visualizar eventos com ferramentas de linha de comando

A EB CLI e a AWS CLI fornecem comandos para recuperar eventos. Se você estiver gerenciando seu ambiente usando a EB CLI, use eb events para imprimir uma lista de eventos. Este comando também tem uma opção `--follow` que continua a mostrar novos eventos até que você pressione Ctrl+C para interromper a saída.

Para extrair eventos usando o AWS CLI, use o `describe-events` comando e especifique o ambiente por nome ou 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"
        },
...
```

Para obter mais informações sobre ferramentas de linha de comando, consulte ???.

Listar e conectar com instâncias de servidor

Este tópico explica como visualizar uma lista das EC2 instâncias da Amazon que executam seu ambiente de aplicativos do Elastic Beanstalk e como se conectar a elas.

Você pode ver uma lista de EC2 instâncias da Amazon executando seu ambiente de AWS Elastic Beanstalk aplicativos por meio do console do Elastic Beanstalk. Você pode se conectar a instâncias usando qualquer cliente SSH. Você pode se conectar a instâncias que executam o Windows usando a Área de Trabalho Remota.

> **Important**
>
> Antes de acessar suas instâncias Amazon provisionadas pelo Elastic Beanstalk, você deve criar um par de chaves EC2 da Amazon e configurar sua Amazon provisionada pelo Elastic Beanstalk para usar o EC2 par de chaves da Amazon. EC2instances EC2 Você pode configurar seus pares de EC2 chaves da Amazon usando o AWS Management Console. Para obter instruções sobre como criar um par de chaves para a Amazon EC2, consulte o Amazon EC2 Getting Started Guide. Para obter mais informações sobre como configurar suas EC2 instâncias da Amazon para usar um par de EC2 chaves da Amazon, consulteEC2 key pair.
>
> Por padrão, o Elastic Beanstalk não permite EC2 conexões remotas com instâncias em um contêiner do Windows, exceto aquelas em contêineres legados do Windows. (O Elastic EC2 Beanstalk configura instâncias em contêineres antigos do Windows para usar a porta 3389 para conexões RDP.) Você pode habilitar conexões remotas com suas EC2 instâncias que executam o Windows adicionando uma regra a um grupo de segurança que autoriza o tráfego de entrada para as instâncias. Recomendamos enfaticamente que você remova a regra quando encerrar sua conexão remota. Você pode adicioná-la novamente na próxima vez que precisar fazer login remotamente. Para obter mais informações, consulte Adicionar uma regra para tráfego de entrada RDP para uma instância do Windows e Conectar-se à

instância do Windows no Guia do usuário do Amazon Elastic Compute Cloud para Microsoft Windows.

Para visualizar e se conectar às EC2 instâncias da Amazon para um ambiente

1. Abra o EC2 console da Amazon em https://console.aws.amazon.com/ec2/.
2. No painel de navegação do console, escolha Load Balancers.
3. Os balanceadores de carga criados pelo Elastic Beanstalk têm awseb no nome. Encontre e clique no load balancer do seu ambiente.
4. Escolha a guia Instances (Instâncias) no painel inferior do console.

   Uma lista de instâncias que o balanceador de carga do seu ambiente do Elastic Beanstalk usa é exibida. Anote o ID da instância à qual você deseja se conectar.
5. No painel de navegação do EC2 console da Amazon, escolha Instâncias e encontre o ID da sua instância na lista.
6. Clique com o botão direito do mouse no ID da EC2 instância da Amazon em execução no balanceador de carga do seu ambiente e selecione Connect no menu de contexto.
7. Anote o endereço DNS público da instância na guia Description.
8. Conecte-se a uma instância executando Linux usando o cliente SSH de sua escolha e digite ssh -i .ec2/mykeypair.pem EC2-user@<public-DNS- >. of-the-instance

Para obter mais informações sobre como se conectar a uma instância do Amazon EC2 Linux, consulte Getting Started with Amazon EC2 Linux Instances no Amazon EC2 User Guide.

Se o seu ambiente do Elastic Beanstalk usa o.NET na plataforma Windows Server, consulte Getting Started with EC2 Amazon Windows Instances no Amazon User Guide. EC2

Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk

Este tópico explica os tipos de logs de instância fornecidos pelo Elastic Beanstalk. Também fornece instruções detalhadas para recuperá-los e gerenciá-los.

As EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk geram registros que você pode visualizar para solucionar problemas com seu aplicativo ou arquivos de configuração. Os registros

são criados pelo servidor web, servidor de aplicativos e scripts da plataforma Elastic Beanstalk e são armazenados localmente em AWS CloudFormation instâncias individuais. Você pode recuperá-los facilmente usando o console de gerenciamento do ambiente ou a CLI do EB. Você também pode configurar seu ambiente para transmitir registros para o Amazon CloudWatch Logs em tempo real.

Logs finais são as últimas 100 linhas dos arquivos de log usados com mais frequência, ou seja, logs operacionais do Elastic Beanstalk e logs do servidor Web ou do servidor de aplicações. Quando logs finais são solicitados no console de gerenciamento do ambiente ou com eb logs, uma instância no ambiente concatena as entradas de log mais recentes em um único arquivo de texto e o carrega no Amazon S3.

Logs em pacote são logs completos para uma maior variedade de arquivos de log, incluindo logs do yum e do cron e vários logs do AWS CloudFormation. Quando logs em pacote são solicitados, uma instância no ambiente empacota todos os arquivos de log em um arquivo ZIP e o carrega no Amazon S3.

Para fazer upload de logs alternados para o Amazon S3, as instâncias no ambiente precisam ter um perfil da instância com permissão para gravar no bucket do Amazon S3 do Elastic Beanstalk. Essas permissões são incluídas no perfil de instância padrão cuja criação o Elastic Beanstalk solicita ao iniciar um ambiente no console do Elastic Beanstalk pela primeira vez.

Como recuperar logs de instâncias

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, selecione Logs.
4. Escolha Request Logs (Logs de solicitação) e escolha o tipo de logs a serem recuperados. Para obter logs finais, escolha Últimas 100 linhas. Para obter logs em pacote, selecione Logs completos.
5. Quando o Elastic Beanstalk terminar de recuperar seus logs, escolha Download.

O Elastic Beanstalk armazena logs finais e em pacote em um bucket do Amazon S3 e gera um URL pré-assinado do Amazon S3 que pode ser usado para acessar os logs. O Elastic Beanstalk exclui os arquivos do Amazon S3 após 15 minutos.

> **Warning**
>
> Qualquer pessoa que tenha o URL do Amazon S3 pré-assinado poderá acessar os arquivos antes que eles sejam excluídos. Disponibilize o URL somente para partes confiáveis.

> **Note**
>
> Sua política de usuário deve ter a permissão `s3:DeleteObject`. O Elastic Beanstalk usa suas permissões de usuário para excluir os logs do Amazon S3.

Para mantê-los, configure o ambiente para publicá-los no Amazon S3 automaticamente depois que forem alternados. Para habilitar a alternância de logs para o Amazon S3, siga o procedimento em Configurar visualização de logs de instância. As instâncias em seu ambiente tentarão fazer upload dos logs que foram girados uma vez por hora.

Se o aplicativo gera logs em um local que não faz parte da configuração padrão da plataforma do ambiente, você pode estender a configuração padrão usando arquivos de configuração (`.ebextensions`). É possível adicionar arquivos de log de seu aplicativo aos logs finais, aos logs em pacote e à rotação de logs.

Para streaming de registros em tempo real e armazenamento de longo prazo, configure seu ambiente para transmitir registros para o Amazon CloudWatch Logs.

Seções

- Localização do log nas EC2 instâncias da Amazon
- Local de log no Amazon S3
- Configurações de rotação de log no Linux
- Extensão da configuração padrão da tarefa de log
- Transmissão de arquivos de log para o Amazon CloudWatch Logs

Localização do log nas EC2 instâncias da Amazon

Os registros são armazenados em locais padrão nas EC2 instâncias da Amazon em seu ambiente. O Elastic Beanstalk gera os logs a seguir.

Amazon Linux 2

- `/var/log/eb-engine.log`

Amazon Linux AMI (AL1)

> **Note**
>
> Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou 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`

Esses logs contêm mensagens sobre atividades de implantação, incluindo mensagens relacionadas a arquivos de configuração (`.ebextensions`).

Cada aplicativo e servidor web armazena logs em uma pasta própria:

- 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/`

Local de log no Amazon S3

Ao solicitar logs finais ou em pacote de seu ambiente, ou quando as instâncias fazem upload de logs alternados, eles são armazenados em seu bucket do Elastic Bucket no Amazon S3. O Elastic Beanstalk cria um `elasticbeanstalk-`*`region`*`-`*`account-id`* bucket com o nome de cada região na qual AWS você cria ambientes. Dentro desse bucket, os logs são armazenados no caminho `resources/environments/logs/`*`logtype`*`/`*`environment-id`*`/`*`instance-id`*.

Por exemplo, os registros da instância`i-0a1fd158`, no AWS ambiente do Elastic `e-mpcwnwheky` Beanstalk `us-west-2` na região na conta, são `123456789012` armazenados nos seguintes locais:

- Logs finais:

  `s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/tail/e-mpcwnwheky/i-0a1fd158`
- Logs em pacote:

  `s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/bundle/e-mpcwnwheky/i-0a1fd158`
- Logs alternados:

  `s3://elasticbeanstalk-us-west-2-123456789012/resources/environments/logs/publish/e-mpcwnwheky/i-0a1fd158`

> **Note**
>
> Você pode localizar o ID do ambiente no console de gerenciamento do ambiente.

O Elastic Beanstalk exclui os logs finais e em pacote do Amazon S3 automaticamente 15 minutos após eles serem criados. Os logs alternados são mantidos até você excluí-los ou movê-los para o S3 Glacier.

Configurações de rotação de log no Linux

Em plataformas Linux, o Elastic Beanstalk usa `logrotate` para alternar logs periodicamente. Se configurado, depois que um log é alternado localmente, a tarefa de alternância de logs o seleciona e

faz upload dele no Amazon S3. Por padrão, os logs que são alternados localmente não são exibidos em logs finais ou em pacote.

É possível encontrar arquivos de configuração do Elastic Beanstalk para `logrotate` em `/etc/logrotate.elasticbeanstalk.hourly/`. Essas configurações de rotação são específicas da plataforma e podem mudar em futuras versões dela. Para obter mais informações sobre as definições disponíveis e configurações de exemplo, execute `man logrotate`.

Os arquivos de configuração são chamado de trabalhos cron em `/etc/cron.hourly/`. Para obter mais informações sobre `cron`, execute `man cron`.

Extensão da configuração padrão da tarefa de log

O Elastic Beanstalk usa arquivos em `/opt/elasticbeanstalk/tasks` subpastas do (Linux`C:\Program Files\Amazon\ElasticBeanstalk\config`) ou (Windows Server) na instância da EC2 Amazon para configurar tarefas para registros finais, registros de pacotes e rotação de registros.

No Amazon Linux:

- Logs finais:

  ```
  /opt/elasticbeanstalk/tasks/taillogs.d/
  ```
- Logs em pacote:

  ```
  /opt/elasticbeanstalk/tasks/bundlelogs.d/
  ```
- Logs alternados:

  ```
  /opt/elasticbeanstalk/tasks/publishlogs.d/
  ```

No Windows Server:

- Logs finais:

  ```
  c:\Program Files\Amazon\ElasticBeanstalk\config\taillogs.d\
  ```
- Logs em pacote:

  ```
  c:\Program Files\Amazon\ElasticBeanstalk\config\bundlelogs.d\
  ```
- Logs alternados:

  ```
  c:\Program Files\Amazon\ElasticBeanstalk\config\publogs.d\
  ```

Por exemplo, o arquivo `eb-activity.conf` no Linux adiciona dois arquivos de log à tarefa de logs finais.

`/opt/elasticbeanstalk/tasks/taillogs.d/eb-activity.conf`

```
/var/log/eb-commandprocessor.log
/var/log/eb-activity.log
```

Você pode usar arquivos de configuração do ambiente (`.ebextensions`) para adicionar seus próprios arquivos `.conf` a essas pastas. Um arquivo `.conf` lista os arquivos de log específicos para a aplicação, que o Elastic Beanstalk adiciona às tarefas de arquivo de log.

Use a seção `files` para adicionar os arquivos de configuração às tarefas que você deseja modificar. Por exemplo, o texto de configuração a seguir adiciona um arquivo de configuração de log a cada instância em seu ambiente: Este arquivo de configuração de log, `cloud-init.conf`, adiciona `/var/log/cloud-init.log` a logs finais.

```
files:
  "/opt/elasticbeanstalk/tasks/taillogs.d/cloud-init.conf" :
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/log/cloud-init.log
```

Adicione esse texto a um arquivo com a extensão de nome de arquivo `.config` em seu pacote de origem em uma pasta chamada `.ebextensions`.

```
~/workspace/my-app
|-- .ebextensions
|   `-- tail-logs.config
|-- index.php
`-- styles.css
```

Em plataformas Linux, você também pode usar caracteres curinga em configurações de tarefas de log. Este arquivo de configuração adiciona todos os arquivos com a extensão de nome de arquivo `.log` da pasta `log` na raiz do aplicativo aos logs em pacote.

```
files:
  "/opt/elasticbeanstalk/tasks/bundlelogs.d/applogs.conf" :
```

```
    mode: "000755"
    owner: root
    group: root
    content: |
      /var/app/current/log/*.log
```

As configurações de tarefas de log não oferecem suporte a caracteres curinga em plataformas Windows.

> **Note**
>
> Para ajudar a se familiarizar com os procedimentos de personalização de log, é possível implantar uma aplicação de exemplo usando a CLI do EB. Para isso, a CLI do EB cria um diretório de aplicação local que contém um subdiretório `.ebextentions` com uma configuração de exemplo. Também é possível usar os arquivos de log da aplicação de exemplo para explorar o recurso de recuperação de log descrito neste tópico.

Para obter mais informações sobre o uso de arquivos de configuração, consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`).

Assim como é possível estender logs finais e logs de pacote, você pode estender a rotação de log usando um arquivo de configuração. Sempre que o Elastic Beanstalk alterna seus próprios logs e os envia para o Amazon S3, ele também alterna e faz upload de seus logs adicionais. A extensão de rotação de log se comporta de forma diferente dependendo da plataforma do sistema operacional. As seções a seguir descrevem os dois casos.

Extensão de rotação de log no Linux

Conforme explicado em Configurações de rotação de log no Linux, o Elastic Beanstalk usa `logrotate` para alternar logs em plataformas Linux. Quando você configura os arquivos de log do seu aplicativo para a rotação de log, o aplicativo não precisa criar cópias de arquivos de log. O Elastic Beanstalk configura `logrotate` para criar uma cópia dos arquivos de log da aplicação para cada alternância. Portanto, o aplicativo deve manter os arquivos de log desbloqueados quando não estiver gravando nesses arquivos.

Extensão da rotação de log no Windows Server

No Windows Server, quando você configura os arquivos de log do seu aplicativo para a rotação de log, o aplicativo deve girar os arquivos de log periodicamente. O Elastic Beanstalk procura arquivos

com nomes iniciados pelo padrão que você configurou e seleciona-os para fazer upload para o Amazon S3. Além disso, pontos no nome de arquivo são ignorados e o Elastic Beanstalk considera o nome até o ponto como a base do nome do arquivo de log.

O Elastic Beanstalk faz upload de todas as versões de um arquivo de log base, exceto da mais recente, porque considera essa a versão do arquivo de log da aplicação ativa, que pode ser bloqueada. O aplicativo pode, portanto, manter o arquivo de log ativo bloqueado entre alternâncias.

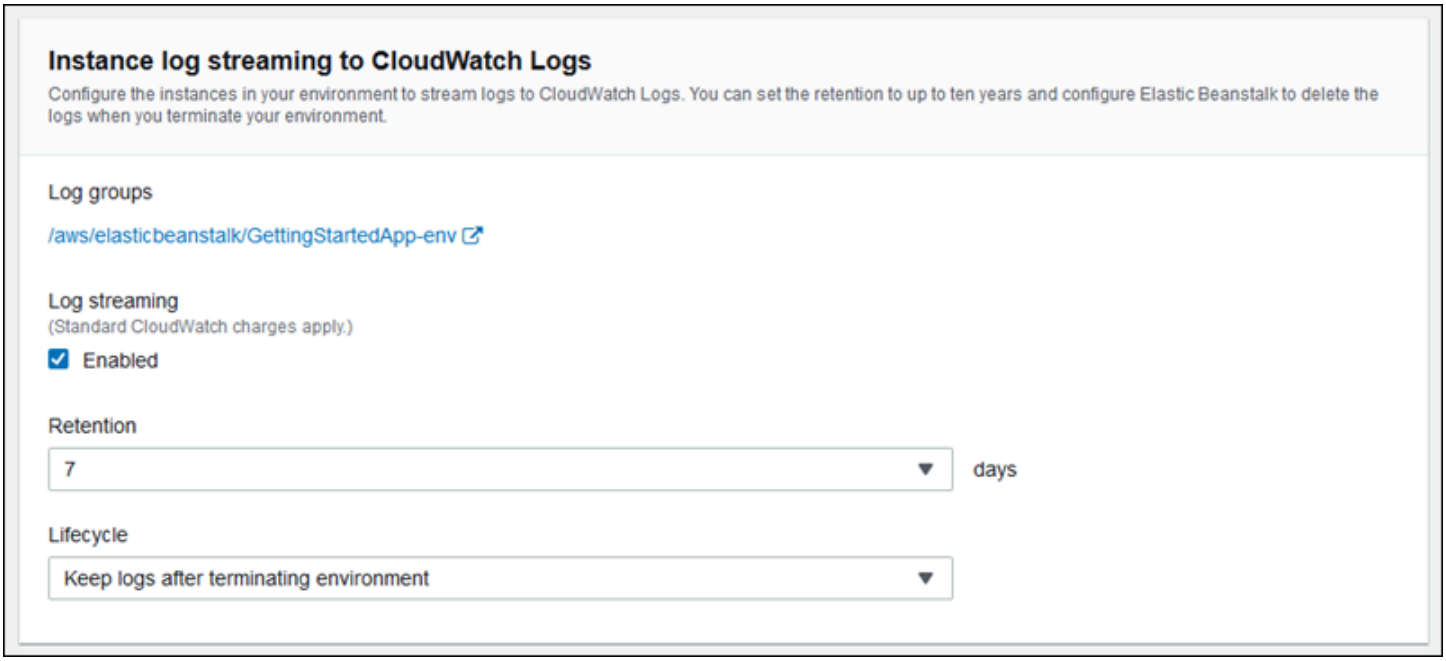
Por exemplo: o aplicativo grava em um arquivo de log chamado `my_log.log` e você especifica esse nome no arquivo `.conf`. O aplicativo periodicamente gira o arquivo. Durante o ciclo de alternância do Elastic Beanstalk, ele localiza os seguintes arquivos na pasta do arquivo de log: `my_log.log`, `my_log.0800.log`, `my_log.0830.log`. O Elastic Beanstalk considera todos eles como versões do nome base `my_log`. O arquivo `my_log.log` tem a hora de modificação mais recente, então o Elastic Beanstalk faz upload apenas dos outros dois arquivos, `my_log.0800.log` e `my_log.0830.log`.

Transmissão de arquivos de log para o Amazon CloudWatch Logs

Você pode configurar seu ambiente para transmitir logs para o Amazon CloudWatch Logs no console do Elastic Beanstalk ou usando opções de configuração. Com o CloudWatch Logs, cada instância em seu ambiente transmite registros para grupos de registros que você pode configurar para serem retidos por semanas ou anos, mesmo após o encerramento do ambiente.

O conjunto de logs transmitido varia por ambiente, mas sempre inclui o `eb-engine.log` e os logs de acesso do servidor de proxy nginx ou Apache executado à frente do aplicativo.

É possível configurar o streaming de logs no console do Elastic Beanstalk durante a criação do ambiente ou em um ambiente existente. Você pode definir as seguintes opções no console: ativar/desativar o streaming de CloudWatch registros para Logs, definir o número de dias de retenção e selecionar entre as opções de ciclo de vida. No exemplo a seguir, os logs são salvos por até sete dias, mesmo quando o ambiente é encerrado.



O arquivo de configuração a seguir permite o streaming de logs com retenção de 180 dias, mesmo se o ambiente estiver encerrado.

Example .ebextensions/log-streaming.config

```
option_settings:
  aws:elasticbeanstalk:cloudwatch:logs:
    StreamLogs: true
    DeleteOnTerminate: false
    RetentionInDays: 180
```

Usando o Elastic Beanstalk com outros serviços AWS

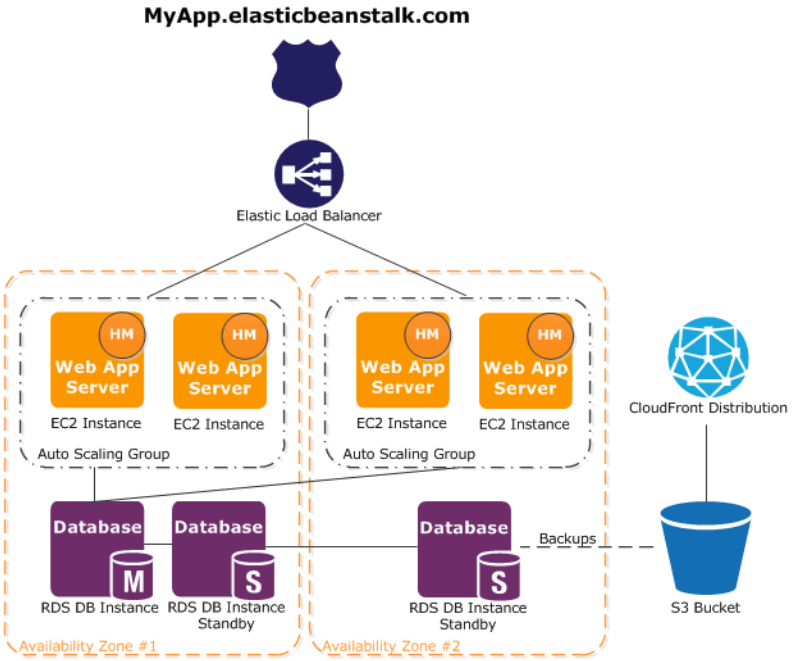
Os tópicos desta seção descrevem as várias maneiras pelas quais você pode usar AWS serviços adicionais com seu aplicativo Elastic Beanstalk. Para implementar os ambientes do seu aplicativo, o Elastic Beanstalk gerencia recursos AWS de outros serviços ou usa suas funcionalidades. Além disso, o Elastic Beanstalk se AWS integra a serviços que ele não usa diretamente como parte de seus ambientes.

Tópicos

- Visão geral da arquitetura
- Usando o Elastic Beanstalk com a Amazon CloudFront
- Registro de chamadas de API do Elastic Beanstalk com AWS CloudTrail
- Usando o Elastic Beanstalk com a Amazon CloudWatch
- Usando o Elastic CloudWatch Beanstalk com o Amazon Logs
- Usando o Elastic Beanstalk com a Amazon EventBridge
- Encontrando e monitorando recursos do Elastic Beanstalk com AWS Config
- Usar o Elastic Beanstalk com o Amazon DynamoDB
- Usando o Elastic Beanstalk com a Amazon ElastiCache
- Usar o Elastic Beanstalk com o Amazon Elastic File System
- Usando o Elastic Beanstalk com AWS Identity and Access Management
- Usar o Elastic Beanstalk com o Amazon RDS
- Usar o Elastic Beanstalk com o Amazon S3
- Usando o Elastic AWS Secrets ManagerAWS Systems Manager Beanstalk com um Parameter Store
- Usar o Elastic Beanstalk com Amazon VPC

Visão geral da arquitetura

O diagrama a seguir ilustra um exemplo de arquitetura do Elastic Beanstalk em várias zonas de disponibilidade trabalhando com AWS outros produtos, como Amazon, Amazon CloudFront Simple Storage Service (Amazon S3) e Amazon Relational Database Service (Amazon RDS).



Para planejar a tolerância a falhas, é recomendável ter instâncias N+1 da EC2 Amazon e distribuir suas instâncias em várias zonas de disponibilidade. No caso improvável de uma zona de disponibilidade cair, você ainda terá suas outras EC2 instâncias da Amazon em execução em outra zona de disponibilidade. Você pode ajustar o Amazon EC2 Auto Scaling para permitir um número mínimo de instâncias, bem como várias zonas de disponibilidade. Para obter instruções sobre como fazer isso, consulte Auto Scaling de suas instâncias do ambiente do Elastic Beanstalk. Para obter mais informações sobre a construção de aplicações tolerantes a falhas, consulte Construção de aplicações tolerantes a falhas na AWS.

As seções a seguir discutem com mais detalhes a integração com Amazon CloudFront, Amazon CloudWatch, Amazon DynamoDB Amazon, ElastiCache Amazon RDS, Amazon Route 53, Amazon Simple Storage Service, Amazon VPC e IAM.

Usando o Elastic Beanstalk com a Amazon CloudFront

CloudFront A Amazon é um serviço web que acelera a distribuição de seu conteúdo web estático e dinâmico, por exemplo, .html, .css, .php, arquivos de imagem e mídia, para usuários finais. CloudFront entrega seu conteúdo por meio de uma rede mundial de pontos de presença. Quando um usuário final solicita conteúdo com o qual você está servindo CloudFront, o usuário é encaminhado para o ponto de presença que fornece a menor latência, para que o conteúdo seja entregue com o melhor desempenho possível. Se o conteúdo já está naquele ponto de presença, o CloudFront o entrega imediatamente. Se o conteúdo não estiver atualmente nesse ponto de presença, CloudFront

recupere-o de um bucket do Amazon S3 ou de um servidor HTTP (por exemplo, um servidor web) que você identificou como a fonte da versão definitiva do seu conteúdo.

Depois de criar e implantar seu aplicativo Elastic Beanstalk, você pode se CloudFront inscrever e CloudFront começar a usá-lo para distribuir seu conteúdo. Saiba CloudFront mais no Amazon CloudFront Developer Guide.

Registro de chamadas de API do Elastic Beanstalk com AWS CloudTrail

O Elastic Beanstalk é AWS CloudTrail integrado com, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um AWS serviço no Elastic Beanstalk. CloudTrailcaptura todas as chamadas de API para o Elastic Beanstalk como eventos, incluindo chamadas do console do Elastic Beanstalk, da CLI do EB e do seu código para o Elastic Beanstalk. APIs Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Elastic Beanstalk. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Elastic Beanstalk, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

Informações sobre o Elastic Beanstalk em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre no Elastic Beanstalk, essa atividade é registrada CloudTrail em um evento junto AWS com outros eventos de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos do Elastic Beanstalk, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, ao criar uma trilha no console, ela é aplicada a todas as regiões da . A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

Todas as ações do Elastic Beanstalk são CloudTrail registradas e documentadas na Referência da API.AWS Elastic Beanstalk Por exemplo, chamadas para as `ListTagsForResource` ações `DescribeApplicationsUpdateEnvironment`, e geram entradas nos arquivos de CloudTrail log.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário-raiz ou usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Noções básicas sobre entradas de arquivo de log do Elastic Beanstalk

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `UpdateEnvironment` ação chamada por um usuário do IAM chamado`intern`, para o `sample-env` ambiente no `sample-app` aplicativo.

```
{
  "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"
  }]
}
```


Usando o Elastic Beanstalk com a Amazon CloudWatch

A Amazon CloudWatch permite monitorar, gerenciar e publicar várias métricas, bem como configurar ações de alarme com base nos dados das métricas. O CloudWatch monitoramento da Amazon permite que você colete, analise e visualize métricas do sistema e do aplicativo para que você possa tomar decisões operacionais e comerciais com mais rapidez e confiança.

Você pode usar CloudWatch a Amazon para coletar métricas sobre seus recursos da Amazon Web Services (AWS), como o desempenho de suas instâncias da Amazon. EC2 Você também pode publicar suas próprias métricas diretamente na Amazon CloudWatch. CloudWatch Os alarmes

da Amazon ajudam você a implementar decisões com mais facilidade, permitindo que você envie notificações ou faça alterações automáticas nos recursos que você está monitorando, com base nas regras que você define. Por exemplo, você pode criar alarmes que iniciem ações do Amazon Auto EC2 Scaling e do Amazon Simple Notification Service (Amazon SNS) em seu nome.

O Elastic Beanstalk CloudWatch usa automaticamente a Amazon para ajudar você a monitorar o status do seu aplicativo e do ambiente. Você pode navegar até o CloudWatch console da Amazon para ver seu painel e ter uma visão geral de todos os seus recursos, bem como dos seus alarmes. Você também pode escolher visualizar mais métricas ou adicionar métricas personalizadas.

Para obter mais informações sobre a Amazon CloudWatch, acesse o Amazon CloudWatch Developer Guide. Para obter um exemplo de como usar a Amazon CloudWatch com o Elastic Beanstalk, consulte. the section called “Exemplo: uso de CloudWatch métricas personalizadas da Amazon”

Usando o Elastic CloudWatch Beanstalk com o Amazon Logs

Este tópico explica os recursos de monitoramento que o serviço Amazon CloudWatch Logs pode fornecer ao Elastic Beanstalk. Ele também orienta na configuração e lista os locais dos logs de cada plataforma do Elastic Beanstalk.

A implementação de CloudWatch registros pode permitir que você realize as seguintes atividades de monitoramento:

- Monitore e arquive seu aplicativo, sistema e arquivos de log personalizados do Elastic Beanstalk a partir das instâncias EC2 Amazon de seus ambientes.
- Configurar alarmes que facilitam a reação a eventos específicos de fluxo de log que seus filtros de métricas extraem.

O agente CloudWatch Logs instalado em cada EC2 instância da Amazon em seu ambiente publica pontos de dados métricos no CloudWatch serviço para cada grupo de log que você configura. Cada grupo de registros aplica seus próprios padrões de filtro para determinar quais eventos de fluxo de log devem ser enviados CloudWatch como pontos de dados. Os streams de log que pertencem ao mesmo grupo de log compartilham as mesmas configurações de retenção, monitoramento e controle de acesso. Você pode configurar o Elastic Beanstalk para transmitir automaticamente os registros CloudWatch para o serviço, conforme descrito em. Transmitir registros de instâncias para CloudWatch Logs Para obter mais informações sobre CloudWatch registros, incluindo terminologia e conceitos, consulte o Guia do usuário do Amazon CloudWatch Logs.

Além dos registros de instância, se você ativar a integridade aprimorada do seu ambiente, poderá configurar o ambiente para transmitir informações de integridade para o CloudWatch Logs. Consulte Streaming de informações sobre a saúde do ambiente do Elastic Beanstalk para o Amazon Logs CloudWatch .

Tópicos

- Pré-requisitos para instanciar o streaming de registros para o Logs CloudWatch
- Como o Elastic Beanstalk configura os registros CloudWatch
- Transmitir registros de instâncias para CloudWatch Logs
- Solução de problemas da integração de CloudWatch registros
- Streaming de informações sobre a saúde do ambiente do Elastic Beanstalk para o Amazon Logs CloudWatch

Pré-requisitos para instanciar o streaming de registros para o Logs CloudWatch

Para permitir o streaming de registros das EC2 instâncias da Amazon do seu ambiente para o CloudWatch Logs, você deve atender às seguintes condições.

- Plataforma: esse recurso só está disponível em versões de plataforma lançadas a partir desta versão, portanto, se você estiver usando uma versão de plataforma anterior, atualize o ambiente para um atual.
- Se você não tem a política gerenciada AWSElasticBeanstalkWebTierou o AWSElasticBeanstalkWorkerTierElastic Beanstalk em seu perfil de instância do Elastic Beanstalk, você deve adicionar o seguinte ao seu perfil para ativar esse recurso.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:PutLogEvents",
      "logs:CreateLogStream"
    ],
    "Resource": [
    "*"
```

```
    ]
  }
  ]
}
```

Como o Elastic Beanstalk configura os registros CloudWatch

O Elastic Beanstalk CloudWatch instala um agente de log com as configurações padrão em cada instância criada. Saiba mais na Referência do CloudWatch Logs Agent.

Quando você ativa o streaming de registros de instâncias para o CloudWatch Logs, o Elastic Beanstalk envia arquivos de log das instâncias do seu ambiente para o Logs. CloudWatch Plataformas diferentes transmitem logs diferentes. A tabela a seguir lista os logs por plataforma.

Plataforma/ramificação da plataforma	Logs
Docker/ ramificação da plataform a: Docker em execução no Amazon Linux 2 de 64 bits	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/docker • /var/log/docker-events.log • /var/log/eb-docker/containers/eb-current-app/stdouterr.log • /var/log/nginx/access.log • /var/log/nginx/error.log
Docker/ ramificação da plataforma: ECS em execução no Amazon Linux 2 de 64 bits	• /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 no Linux	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/web.stdout.log

Plataforma/ramificação da plataforma	Logs
Java/ramificação da plataforma: Corretto em execução no Amazon Linux 2 de 64 bits	• /var/log/nginx/access.log • /var/log/nginx/error.log
Node.js Python	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/web.stdout.log • /var/log/httpd/access_log • /var/log/httpd/error_log • /var/log/nginx/access.log • /var/log/nginx/error.log
Tomcat PHP	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/httpd/access_log • /var/log/httpd/error_log • /var/log/nginx/access.log • /var/log/nginx/error.log
.NET no Windows Server	• C:\inetpub\logs\\ W3 LogFilesSVC1\ u_ex*.log • C:\Program Files\ Amazon\ElasticBeanstalk\ logs\ AWSDeployment .log • C:\Program Files\ Amazon\ElasticBeanstalk\ logs\ Hooks.log
Ruby	• /var/log/eb-engine.log • /var/log/eb-hooks.log • /var/log/puma/puma.log • /var/log/web.stdout.log • /var/log/nginx/access.log • /var/log/nginx/error.log

Arquivos de log em plataformas da AMI do Amazon Linux

> **Note**
>
> Em 18 de julho de 2022, o Elastic Beanstalk definiu o status de todas as ramificações da plataforma baseadas no Amazon Linux AMI () como descontinuadas. AL1 Para obter mais informações sobre como migrar para uma ramificação da plataforma Amazon Linux 2023 atual e totalmente compatível, consulte Migrar a aplicação Linux do Elastic Beanstalk para o Amazon Linux 2023 ou Amazon Linux 2.

A tabela a seguir lista os arquivos de log transmitidos de instâncias em ramificações da plataforma com base na AMI do Amazon Linux (anterior ao Amazon Linux 2), por plataforma.

Plataforma/ramificação da plataforma	Logs
Docker/ ramificação da plataform a: Docker em execução no Amazon Linux de 64 bits	• /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log • /var/log/eb-docker/containers/eb-current-app/stdouterr.log
Docker/ ramificação da plataforma: Docker de vários contêiner es em execução no Amazon Linux de 64 bits	• /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 pré-confi gurado)	• /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/docker-events.log • /var/log/docker • /var/log/nginx/access.log

Plataforma/ramificação da plataforma	Logs
Go	• /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/nginx/access.log
Java/ ramificação da plataforma: Java 8 em execução no Amazon Linux de 64 bits Ramificação da plataforma: Java 7 em execução no Amazon Linux de 64 bits	• /var/log/eb-activity.log • /var/log/nginx/access.log • /var/log/nginx/error.log • /var/log/web-1.error.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/nginx/error.log • /var/log/nginx/access.log • /var/log/httpd/error.log • /var/log/httpd/access.log
PHP	• /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log

Plataforma/ramificação da plataforma	Logs
Python	• /var/log/eb-activity.log • /var/log/httpd/error_log • /var/log/httpd/access_log • /opt/python/log/supervisord.log
Ruby/ ramificação da plataforma: Puma com Ruby em execução no Amazon Linux de 64 bits	• /var/log/eb-activity.log • /var/log/nginx/error.log • /var/log/puma/puma.log • /var/log/nginx/access.log
Ruby/ ramificação da plataforma: Passenger com Ruby em execução no Amazon Linux de 64 bits	• /var/log/eb-activity.log • /var/app/support/logs/passenger.log • /var/app/support/logs/access.log • /var/app/support/logs/error.log

O Elastic Beanstalk configura CloudWatch grupos de log em Logs para os vários arquivos de log que ele transmite. Para recuperar arquivos de log específicos do CloudWatch Logs, você precisa saber o nome do grupo de log correspondente. O esquema de nomenclatura do grupo de logs depende do sistema operacional da plataforma.

Para plataformas Linux, prefixe o local do arquivo de log na instância com `/aws/elasticbeanstalk/`*`environment_name`* para obter o nome do grupo de logs. Por exemplo, para recuperar o arquivo `/var/log/nginx/error.log`, especifique o grupo de logs `/aws/elasticbeanstalk/`*`environment_name`*`/var/log/nginx/error.log`.

Para plataformas Windows, consulte a tabela a seguir referente ao grupo de logs correspondente a cada arquivo de log.

Arquivo de log na instância	Grupo de logs
`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`
`C:\inetpub\logs\LogFiles` (todo o diretório)	`/aws/elasticbeanstalk/<environment-name>/IIS-Log`

Transmitir registros de instâncias para CloudWatch Logs

Você pode habilitar o streaming de registros de instâncias para o CloudWatch Logs usando o console do Elastic Beanstalk, a CLI do EB ou as opções de configuração.

Antes de habilitá-lo, configure as permissões do IAM para usar com o agente do CloudWatch Logs. Você pode anexar a seguinte política personalizada ao perfil da instância atribuído a seu ambiente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Streaming de logs de instâncias usando o console do Elastic Beanstalk

Para transmitir registros de instâncias para CloudWatch Logs

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Em Transmissão de log de instância para CloudWatch Logs:
   - Ative Log streaming (Streaming de logs).
   - Defina Retention (Retenção) com o número de dias para salvar os logs.
   - Selecione a configuração Lifecycle (Ciclo de vida) que determina se os logs devem ser salvos depois que o ambiente é encerrado.
6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Depois de ativar o streaming de logs, você poderá voltar para a categoria ou a página de configuração Software para encontrar o link Log Groups (Grupos de logs). Clique neste link para ver seus registros no CloudWatch console.

Streaming de logs de instância usando a EB CLI

Para ativar o streaming de registros da instância para o CloudWatch Logs usando a CLI do EB, use o eb logs comando.

```
$ eb logs --cloudwatch-logs enable
```

Você também pode usar eb logs para recuperar registros do CloudWatch Logs. Você pode recuperar todos os logs de instância do ambiente ou usar as várias opções do comando para especificar subconjuntos de logs que devem ser recuperados. Por exemplo, o comando a seguir recupera o conjunto completo de logs de instância para seu ambiente e salva todos eles em um diretório em `.elasticbeanstalk/logs`.

```
$ eb logs --all
```

Especificamente, a opção `--log-group` permite que você recupere logs da instância de um grupo de logs específico, correspondente a um arquivo de log específico na instância. Para fazer isso, você precisa saber o nome do grupo de logs que corresponde ao arquivo de log que você deseja recuperar. Você pode encontrar essas informações em Como o Elastic Beanstalk configura os registros CloudWatch .

Streaming de logs de instância usando arquivos de configuração

Ao criar ou atualizar um ambiente, você pode usar um arquivo de configuração para instalar e configurar o streaming de registros da instância para o CloudWatch Logs. O exemplo de arquivo de configuração a seguir ativa o streaming de logs de instância padrão. O Elastic Beanstalk faz streaming do conjunto padrão de arquivos de log para a plataforma do ambiente. Para usar o exemplo, copie o texto em um arquivo com a extensão `.config` no diretório `.ebextensions`, no nível superior do pacote de origem do aplicativo.

```
option_settings:
  - namespace: aws:elasticbeanstalk:cloudwatch:logs
    option_name: StreamLogs
    value: true
```

Streaming do arquivo de log personalizado

A CloudWatch integração do Elastic Beanstalk com o Logs não oferece suporte direto ao streaming de arquivos de log personalizados que seu aplicativo gera. Para transmitir registros personalizados, use um arquivo de configuração para instalar diretamente o CloudWatch agente e configurar os arquivos a serem enviados. Para obter um arquivo de configuração demonstrativo, consulte `logs-streamtocloudwatch-linux.config`.

> **Note**
>
> O exemplo não funciona na plataforma Windows.

Para obter mais informações sobre a configuração de CloudWatch registros, consulte a referência do arquivo de configuração do CloudWatch agente no Guia do CloudWatch usuário da Amazon.

Solução de problemas da integração de CloudWatch registros

Não é possível localizar os registros da instância do ambiente

Se você não conseguir encontrar alguns dos registros de instância do ambiente que você espera em CloudWatch Logs, investigue os seguintes problemas comuns:

- Sua função do IAM não tem as permissões do IAM necessárias.
- Você lançou seu ambiente em um ambiente Região da AWS que não é compatível com CloudWatch Logs.
- Um de seus arquivos de log personalizados não existe no caminho especificado.

Registros de aplicativos ausentes ou intermitentes

Se os registros do aplicativo do Elastic Beanstalk`/var/log/web.stdout.log`, (), parecerem ausentes ou intermitentes, isso pode ser devido às configurações padrão de limitação de taxa no rsyslog e no journald. Embora a desativação total da limitação de taxa possa resolver esse problema, ela não é recomendada, pois pode levar ao uso excessivo do disco, potencial negação de serviço ou degradação do desempenho do sistema durante picos inesperados de registros. Em vez disso, você pode ajustar os limites de taxa usando o seguinte `.ebextensions configuration`. Essa configuração aumenta o intervalo do limite de taxa para 600 segundos com limites de intermitência mais altos, fornecendo um equilíbrio entre o registro adequado e a proteção do sistema.

Problemas de controle de utilização

Se uma operação do Elastic Beanstalk que executa simultaneamente um grande número de instâncias retornar uma `Error: fail to create log stream: ThrottlingException: Rate exceeded` mensagem como a de que está limitando muitas chamadas para a API. CloudWatch

Para resolver o problema de limitação, execute uma das seguintes ações:

- Use um tamanho de lote menor com implantações contínuas para reduzir as atualizações simultâneas.
- Solicite um aumento na cota de serviço limite de transação por segundo (TPS) da sua AWS conta para. CreateLogStream Para obter mais informações, consulte Cotas de CloudWatch registros e Gerenciamento de suas cotas do serviço de CloudWatch registros no Guia do usuário do Amazon CloudWatch Logs.

Streaming de informações sobre a saúde do ambiente do Elastic Beanstalk para o Amazon Logs CloudWatch

Se você habilitar relatórios de integridade aprimorados para seu ambiente, poderá configurar o ambiente para transmitir informações de integridade para o CloudWatch Logs. Esse streaming é independente do streaming de registros de EC2 instâncias da Amazon. Este tópico descreve o processo de streaming de informações de integridade do ambiente. Para obter mais informações sobre streaming de logs de instância, consulte Usando o Elastic CloudWatch Beanstalk com o Amazon Logs.

Quando você configura o streaming de integridade do ambiente, o Elastic Beanstalk CloudWatch cria um grupo de registros de registros para a integridade do ambiente. O nome do grupo de logs é `/aws/elasticbeanstalk/`*`environment-name`*`/environment-health.log`. Dentro desse grupo de logs, o Elastic Beanstalk cria streams de logs denominados *`YYYY-MM-DD`*#*`<hash-suffix>`* (pode haver mais de um stream de logs por data).

Quando o status de integridade do ambiente é alterado, o Elastic Beanstalk adiciona um registro ao stream de logs de integridade. O registro representa a transição do status de integridade – o novo status e uma descrição do motivo da alteração. Por exemplo, o status de um ambiente pode se alterar para Grave porque o load balancer está falhando. Para obter uma descrição do status de integridade avançada, consulte Status e cores de integridade.

Pré-requisitos para o streaming da integridade do ambiente para o Logs CloudWatch

Para habilitar o streaming de integridade do ambiente para o CloudWatch Logs, você deve atender às seguintes condições:

- Plataforma: você precisa usar uma versão de plataforma compatível com os relatórios de integridade aprimorada.
- Permissões: é necessário conceder determinadas permissões relacionadas ao registro em log ao Elastic Beanstalk para que ele possa atuar em seu nome com o objetivo de fazer streaming de informações de integridade do ambiente. Se o ambiente não estiver usando uma função de serviço criada pelo Elastic Beanstalk para ele, `aws-elasticbeanstalk-service-role` ou a função vinculada ao serviço de sua conta, `AWSServiceRoleForElasticBeanstalk`, lembre-se de adicionar as permissões a seguir à sua função de serviço personalizada.

```
{
      "Effect": "Allow",
```

```
      "Action": [
        "logs:DescribeLogStreams",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/elasticbeanstalk/*:log-stream:*"
}
```

Transmissão de registros de integridade do ambiente para CloudWatch Logs

Você pode habilitar o streaming de integridade do ambiente para CloudWatch Logs usando o console do Elastic Beanstalk, a CLI do EB ou as opções de configuração.

Fazer streaming de logs de integridade do ambiente usando o console do Elastic Beanstalk

Para transmitir registros de integridade do ambiente para o CloudWatch Logs

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Monitoring (Monitoramento), escolha Edit (Editar).
5. Em Health reporting (Relatório de integridade), verifique se System (Sistema) está definido como Enhanced (Avançado).
6. Transmissão de eventos Under Health para CloudWatch Logs
   - Ative Log streaming (Streaming de logs).
   - Defina Retention (Retenção) com o número de dias para salvar os logs.
   - Selecione a configuração Lifecycle (Ciclo de vida) que determina se os logs devem ser salvos depois que o ambiente é encerrado.
7. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Depois de ativar o streaming de logs, você poderá voltar para a categoria ou a página de configuração Monitoring (Monitoramento) para encontrar o link Log Group (Grupo de logs). Clique neste link para ver os registros de integridade do seu ambiente no CloudWatch console.

Streaming de logs de integridade do ambiente usando a EB CLI

Para habilitar o streaming de registros de integridade do ambiente para CloudWatch registros usando a CLI do EB, use o eb logs comando.

```
$ eb logs --cloudwatch-logs enable --cloudwatch-log-source environment-health
```

Você também pode usar eb logs para recuperar registros do CloudWatch Logs. Por exemplo, o comando a seguir recupera todos os logs de integridade para seu ambiente e os salva em um diretório em `.elasticbeanstalk/logs`.

```
$ eb logs --all --cloudwatch-log-source environment-health
```

Streaming de logs de integridade do ambiente usando arquivos de configuração

Ao criar ou atualizar um ambiente, você pode usar um arquivo de configuração para configurar e configurar o streaming de registros de integridade do ambiente para o CloudWatch Logs. Para usar o exemplo abaixo, copie o texto em um arquivo com a extensão `.config` no diretório `.ebextensions`, no nível superior do pacote de origem de seu aplicativo. O exemplo configura o Elastic Beanstalk para habilitar o streaming de logs de integridade do ambiente, manter os logs depois de encerrar o ambiente e salvá-los por 30 dias.

Example Arquivo de configuração de streaming de integridade

```
#############################################################################
##  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
```

Sobre opções padrão e valores válidos, consulte `aws:elasticbeanstalk:cloudwatch:logs:health`.

Usando o Elastic Beanstalk com a Amazon EventBridge

Usando a Amazon EventBridge, você pode configurar regras orientadas por eventos que monitoram seus recursos do Elastic Beanstalk e iniciam ações direcionadas que usam outros serviços. AWS Por exemplo, você pode definir uma regra para enviar notificações por e-mail sinalizando um tópico do Amazon SNS sempre que a integridade de um ambiente de produção mudar para um status de Warning (Aviso). . Ou você pode definir uma função do Lambda para passar uma notificação ao Slack sempre que a integridade do seu ambiente mudar para um status Degradated (Degradado) ou Severe (Grave).

Você pode criar regras na Amazon EventBridge para atuar em qualquer um dos seguintes eventos do Elastic Beanstalk:

- Alterações de estado para operações de ambiente (incluindo operações de criação, atualização e encerramento). O evento especificará se a alteração de estado foi iniciada, bem-sucedida ou falhou.
- Alterações de estado para outros recursos. Além dos ambientes, outros recursos monitorados incluem balanceadores de carga, grupos de escalabilidade automática e instâncias.
- Transição de integridade para ambientes. O evento indica onde a integridade do ambiente passou de um determinado estado para outro.
- Alteração de estado para atualizações gerenciadas. O evento especificará se a alteração de estado foi iniciada, bem-sucedida ou falhou.

Para capturar eventos específicos do Elastic Beanstalk nos quais você está interessado, defina EventBridge padrões específicos de eventos que possam ser usados para detectar os eventos. Os padrões de eventos têm a mesma estrutura que os eventos aos quais correspondem. O padrão menciona os campos com os quais você deseja fazer a correspondência e fornece os valores que você está procurando. Os eventos são emitidos com base no melhor esforço. Eles são entregues pelo Elastic EventBridge Beanstalk para quase em tempo real em circunstâncias operacionais normais. No entanto, podem surgir situações que podem atrasar ou impedir a entrega de um evento.

Para obter uma lista de campos incluídos nos eventos do Elastic Beanstalk e seus possíveis valores de string, consulte Mapeamento de campo de evento do Elastic Beanstalk. Para obter informações

sobre como EventBridge as regras funcionam com padrões de eventos, consulte Eventos e padrões de eventos em EventBridge.

Monitore um recurso do Elastic Beanstalk com EventBridge

Com EventBridge, você pode criar regras que definem ações a serem tomadas quando o Elastic Beanstalk emite eventos para seus recursos. Por exemplo, você pode criar uma regra que envia uma mensagem de e-mail sempre que o status de um ambiente for alterado.

O EventBridge console tem uma opção de padrão predefinida para criar padrões de eventos do Elastic Beanstalk. Se você selecionar essa opção no EventBridge console ao criar uma regra, poderá criar rapidamente um padrão de eventos do Elastic Beanstalk. Você só precisa selecionar os campos e os valores do evento. Ao fazer seleções, o console constrói e exibe o padrão do evento. Como alternativa, você pode editar manualmente o padrão do evento que você criar e pode salvá-lo como um padrão personalizado. O console também fornece a opção para exibir um Sample Event (Exemplo de evento) detalhado que você pode copiar e colar no padrão de evento que está criando.

Se você preferir digitar ou copiar e colar um padrão de evento no EventBridge console, poderá optar por usar a opção Padrão personalizado no console. Ao fazer isso, você não precisará passar pelas etapas de seleção de campos e valores descritos anteriormente. Este tópico oferece exemplos de padrões de correspondência de eventos e eventos do Elastic Beanstalk que você pode usar.

Para criar uma regra para um evento de recurso

1. Faça login AWS usando uma conta que tenha permissões de uso EventBridge e do Elastic Beanstalk.
2. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
3. No painel de navegação, escolha Regras.
4. Escolha Criar Regra.
5. Informe um Name (Nome) para a regra e, opcionalmente, uma descrição.
6. Em Barramento de eventos, escolha padrão. Quando um AWS serviço em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
7. Em Tipo de Regra, escolha Regra com Padrão de Evento.
8. Escolha Próximo.
9. Em Origem do evento, escolha AWS eventos ou eventos de EventBridge parceiros.
10. (Opcional) Para Exemplo de evento, selecioneEventos da AWS . Insira Elastic Beanstalk no campo de pesquisa. Isso fornecerá uma lista de exemplos de eventos do Elastic Beanstalk

que é possível escolher para exibição. Essa etapa simplesmente exibe um exemplo de evento que você pode referenciar. Isso não afeta o resultado da criação da regra. A seção Eventos demonstrativos do Elastic Beanstalk mais adiante neste tópico fornecerá exemplos do mesmo tipo de evento.

11. Na seção Event patter (Padrão de evento), selecione Event pattern form (Formulário de padrão de evento).

> **Note**
>
> Se você já tem texto para um padrão de evento e não precisa que o EventBridge console o crie para você, selecione Padrão personalizado (editor JSON). Você pode inserir manualmente ou copiar e colar texto na caixa Event pattern (Padrão de evento). Selecione Next (Seguinte) e acesse a etapa sobre como inserir um destino.

12. Em Fonte do evento, selecione Serviços da AWS .
13. Em Serviço da AWS , selecione Elastic Beanstalk.
14. Em Event type (Tipo de evento), selecione Status Change (Alteração de status).
15. Essa etapa aborda como você pode trabalhar com os campos de evento detail type (tipo de detalhe), statuse severity (gravidade) do Elastic Beanstalk. À medida que você escolhe esses campos e os valores que deseja corresponder, o console cria e exibe o padrão do evento.

    - Se você selecionar somente um valor para Tipos de detalhes específicos, poderá escolher um ou mais valores para o próximo campo na hierarquia.
    - Se você escolher mais de um valor para Tipos de detalhes específicos, não escolha valores específicos para os próximos campos na hierarquia. Isso evita uma lógica de correspondência ambígua entre campos no seu padrão de evento.

    O campo de evento environment (ambiente) não é afetado por essa hierarquia. Portanto, ele é exibido conforme descrito na próxima etapa.

16. Para o ambiente, selecione Any environment (Qualquer ambiente) ou Specific environment(s) (Ambientes específicos).

    - Se você selecionar Ambientes específicos, poderá escolher um ou mais ambientes na lista suspensa. EventBridge adiciona todos os ambientes que você seleciona na lista EnvironmentName[] na seção de detalhes do padrão do evento. Em seguida, sua regra filtra todos os eventos para incluir apenas os ambientes específicos que você escolher.

- Se você selecionar Any environment (Qualquer ambiente), nenhum ambiente será adicionado ao seu padrão de evento. Por isso, sua regra não filtra qualquer dos eventos do Elastic Beanstalk com base no ambiente.

17. Escolha Next (Avançar).
18. Em Tipos de destino, escolha Serviço da AWS .
19. Em Select a target (Selecionar um destino), escolha a ação de destino a ser executada quando um evento de alteração de estado de recurso for recebido do Elastic Beanstalk.

    Por exemplo, é possível usar um tópico do Amazon Simple Notification Service (SNS) para enviar um e-mail ou mensagem de texto quando ocorrer um evento. Para fazer isso, você precisa criar um tópico do Amazon SNS usando o console do Amazon SNS. Para saber mais, consulte Usar o Amazon SNS para notificações de usuários.

    > **Important**
    >
    > Algumas ações de destino podem exigir o uso de outros serviços e gerar cobranças adicionais, como o Amazon SNS ou o serviço Lambda. Para obter mais informações sobre AWS preços, consultehttps://aws.amazon.com/pricing/. Alguns serviços fazem parte do nível de uso AWS gratuito. Se for um cliente novo, você pode testar esses serviços gratuitamente. Consulte https://aws.amazon.com/free/ para obter mais informações.

20. (Opcional) Escolha Add another target (Adicionar outro destino) para especificar uma ação de destino adicional para a regra de evento.
21. Escolha Próximo.
22. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte as EventBridge tags da Amazon no Guia EventBridge do usuário da Amazon.
23. Escolha Próximo.
24. Analise os detalhes da regra e selecione Criar regra.

Eventos demonstrativos de padrões de eventos do Elastic Beanstalk

Os padrões de eventos têm a mesma estrutura que os eventos aos quais correspondem. O padrão menciona os campos com os quais você deseja fazer a correspondência e fornece os valores que você está procurando.

- Health status change (Alteração de estado de integridade) para todos os ambientes

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Health status change"
    ]
}
```

- Health status change (Alteração do estado de integridade) para os seguintes ambientes: `myEnvironment1` e `myEnvironment2`. Esse padrão de evento filtra para esses dois ambientes específicos, enquanto o exemplo anterior de Health status change (Alteração do estado de integridade) que não filtra envia eventos para todos os ambientes.

```
{"source": [
    "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Health status change"
    ],
    "detail": {
        "EnvironmentName": [
            "myEnvironment1",
            "myEnvironment2"
        ]
    }
}
```

- Elastic Beanstalk resource status change (Alteração de status do recurso do Elastic Beanstalk) para todos os ambientes

```
{
  "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Elastic Beanstalk resource status change"
    ]
}
```

- Elastic Beanstalk resource status change (Alteração de status do recurso do Elastic Beanstalk) com `Status` a atualização do ambiente falhou e `Severity` ERRO para os seguintes ambientes: `myEnvironment1` e `myEnvironment2`

```
{"source": [
    "aws.elasticbeanstalk"
    ],
    "detail-type": [
        "Elastic Beanstalk resource status change"
    ],
    "detail": {
        "Status": [
            "Environment update failed"
            ],
        "Severity": [
            "ERROR"
            ],
        "EnvironmentName": [
            "myEnvironment1",
            "myEnvironment2"
        ]
    }
}
```

- Other resource status change (Alteração de status de outros recursos) para balanceadores de carga, grupos de Auto Scaling e instâncias

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
  "detail-type": [
    "Other resource status change"
    ]
}
```

- Managed update status change (Alteração de status de atualização gerenciada) para ambientes

```
{
   "source": [
    "aws.elasticbeanstalk"
  ],
```

```
  "detail-type": [
    "Managed update status change"
    ]
}
```

- Para capturar all events (todos os eventos) do Elastic Beanstalk (exclua a seção `detail-type`)

```
{
  "source": [
    "aws.elasticbeanstalk"
  ]
}
```

Eventos demonstrativos do Elastic Beanstalk

Veja a seguir um evento demonstrativo do Elastic Beanstalk para uma alteração de status de recurso:

```
{
   "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"
   }
}
```

Veja a seguir um evento demonstrativo do Elastic Beanstalk para uma alteração de status de integridade:

```
{
   "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"
   }
}
```

Mapeamento de campo de evento do Elastic Beanstalk

A tabela a seguir mapeia os campos de eventos do Elastic Beanstalk e seus possíveis valores de string para o campo. EventBridge `detail-type` Para obter mais informações sobre como EventBridge funciona com padrões de eventos para um serviço, consulte Eventos e padrões de eventos em EventBridge.

EventBridge tipo de detalhe do campo	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
Alteração de status do recurso do Elastic Beanstalk	Criação de ambiente iniciada	INFO	createEnvironment está sendo iniciada.
	Criação de ambiente bem-sucedida	INFO	createEnvironment concluída com êxito.
	Criação de ambiente bem-sucedida	INFO	Ambiente executado: <Environment Name>. No entanto, houve problemas durante a execução. Consulte o log de eventos para obter detalhes.
	Assistente de criação de ambiente	ERRO	Falha ao executar o ambiente.
	Atualização do ambiente iniciada	INFO	A atualização do ambiente está sendo iniciada.
	Atualização do ambiente bem-sucedida	INFO	Atualização do ambiente concluída com êxito.

EventBridge tipo de detalhe do campo	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
	Falha na atualização do ambiente	ERRO	Falha ao implantar a configuração.
	Encerramento do ambiente iniciado	INFO	terminateEnvironment está sendo iniciado.
	Encerramento do ambiente bem-sucedido	INFO	terminateEnvironment concluído com êxito.
	Falha no encerramento do ambiente	INFO	A etapa de encerramento do ambiente falhou porque pelo menos um dos fluxos de trabalho de encerramento do ambiente falhou.
Alteração de status de outros recursos	Grupo de Auto Scaling criado	INFO	createEnvironment está sendo iniciada.
	Grupo de Auto Scaling excluído	INFO	createEnvironment está sendo iniciada.

EventBridge tipo de detalhe do campo	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
	Instância adicionada	INFO	Instância adicionada [i-123456789a12b1234] ao ambiente.
	Instância removida	INFO	Instância removida [i-123456789a12b1234] do ambiente.
	Balanceador de carga criado	INFO	Balanceador de carga criado chamado: <LB Name>
	Balanceador de carga excluído	INFO	Balanceador de carga excluído chamado: <LB Name>
Alteração do estado de integridade	A integridade do ambiente mudou	INFO/AVISO	A integridade do ambiente passou para <healthStatus>.
	A integridade do ambiente mudou	INFO/AVISO	A integridade do ambiente passou de <healthStatus> para <healthStatus>.
Alteração de status de atualização gerenciada	Atualização gerenciada iniciada	INFO	A atualização da plataforma gerenciada está em andamento.

EventBridge tipo de detalhe do campo	Campo do Elastic Beanstalk Status	Campo do Elastic Beanstalk Severity (Gravidade)	Campo do Elastic Beanstalk Message (Mensagem)
	Falha na atualização gerenciada	INFO	Falha na atualização gerenciada, tentando novamente em%s minutos.

Encontrando e monitorando recursos do Elastic Beanstalk com AWS Config

AWS Configfornece uma visão detalhada da configuração dos AWS recursos em sua AWS conta. Você pode ver como os recursos estão relacionados, obter um histórico de alterações de configuração e ver como os relacionamentos e as configurações foram alterados ao longo do tempo. Você pode usar AWS Config para definir regras que avaliam as configurações de recursos para fins de conformidade de dados.

Vários tipos de recursos do Elastic Beanstalk são integrados com: AWS Config

- Aplicativos
- Versões do aplicativo
- Ambientes do

A seção a seguir mostra como configurar AWS Config para registrar recursos desses tipos.

Para obter mais informações sobre AWS Config, consulte o Guia do AWS Config desenvolvedor. Para obter informações sobre preços, consulte a página de informações de de preços do AWS Config.

Conf AWS Config iguração

Para configurar inicialmente AWS Config, consulte os tópicos a seguir no Guia do AWS Config desenvolvedor.

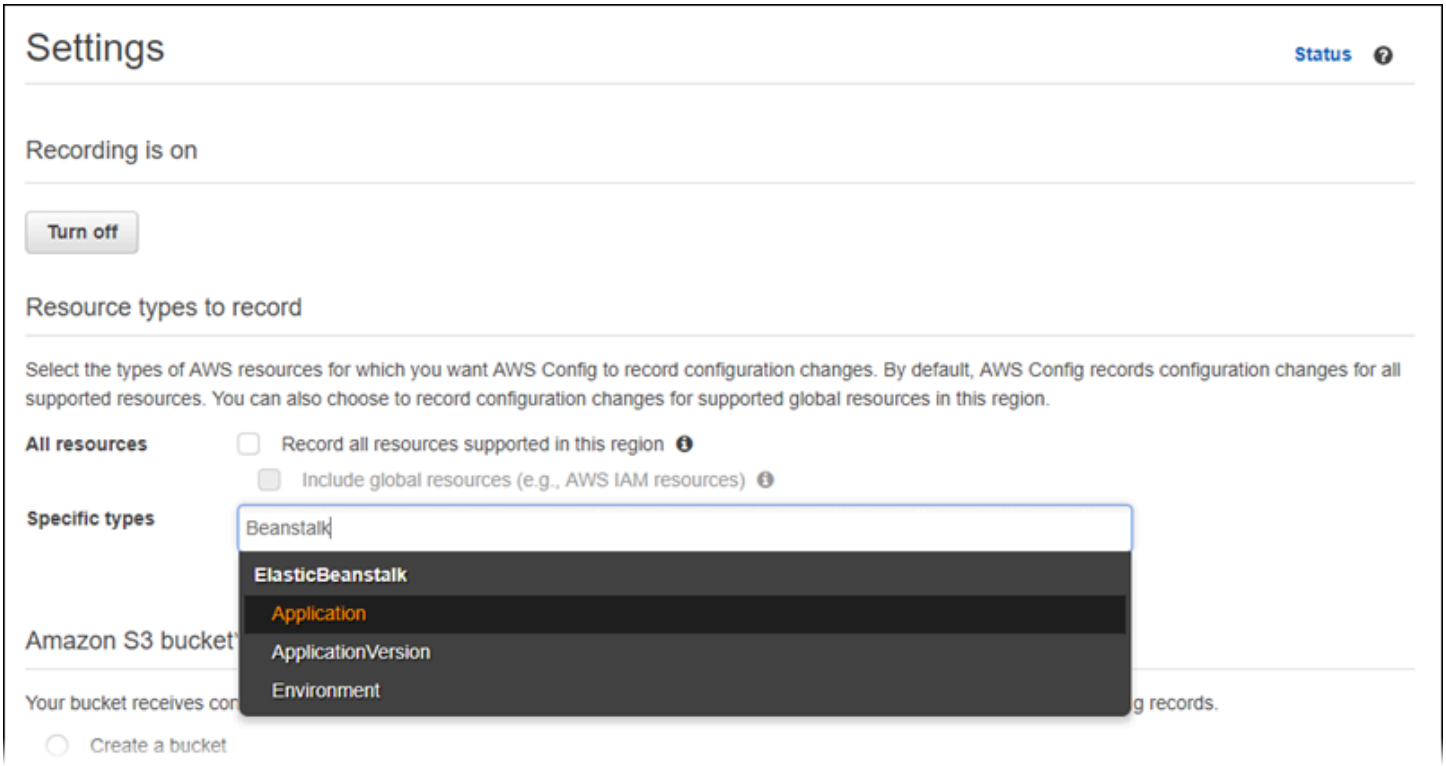
- Configurando AWS Config com o console
- Configurando AWS Config com o AWS CLI

Configuração AWS Config para registrar recursos do Elastic Beanstalk

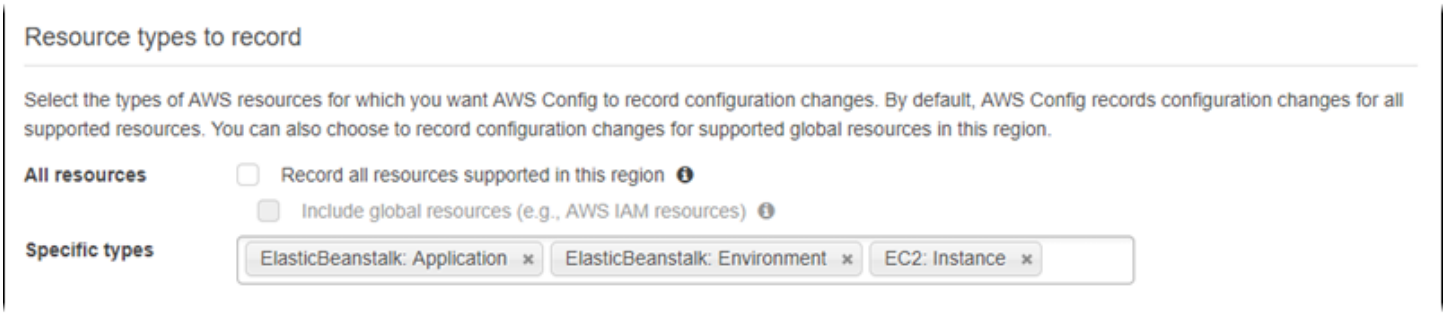
Por padrão, AWS Config registra as alterações de configuração de todos os tipos suportados de recursos regionais que ele descobre na região em que seu ambiente está sendo executado. Você pode personalizar AWS Config para registrar alterações somente para tipos de recursos específicos ou alterações em recursos globais.

Por exemplo, você pode configurar AWS Config para registrar alterações nos recursos do Elastic Beanstalk e em um subconjunto de outros AWS recursos que o Elastic Beanstalk inicia para você. Usando o AWS Config console, você pode selecionar o Elastic Beanstalk como um recurso AWS Config na página Configurações do campo Tipos específicos. A partir daí, você pode optar por registrar qualquer um dos tipos de recursos do Elastic Beanstalk: ApplicationVersionAplicativo e Ambiente.

A figura a seguir mostra a página AWS Config Configurações, com os tipos de recursos do Elastic Beanstalk que você pode escolher registrar: ApplicationVersionAplicativo e Ambiente.



Depois de selecionar alguns tipos de recursos, esta é a forma que a lista dos Specific types (Tipos específicos) é exibida.



Para saber mais sobre recursos regionais versus globais e para conhecer o procedimento completo de personalização, consulte Seleção de quais AWS Config registros de recursos.

Visualizando detalhes de configuração do Elastic Beanstalk no console AWS Config

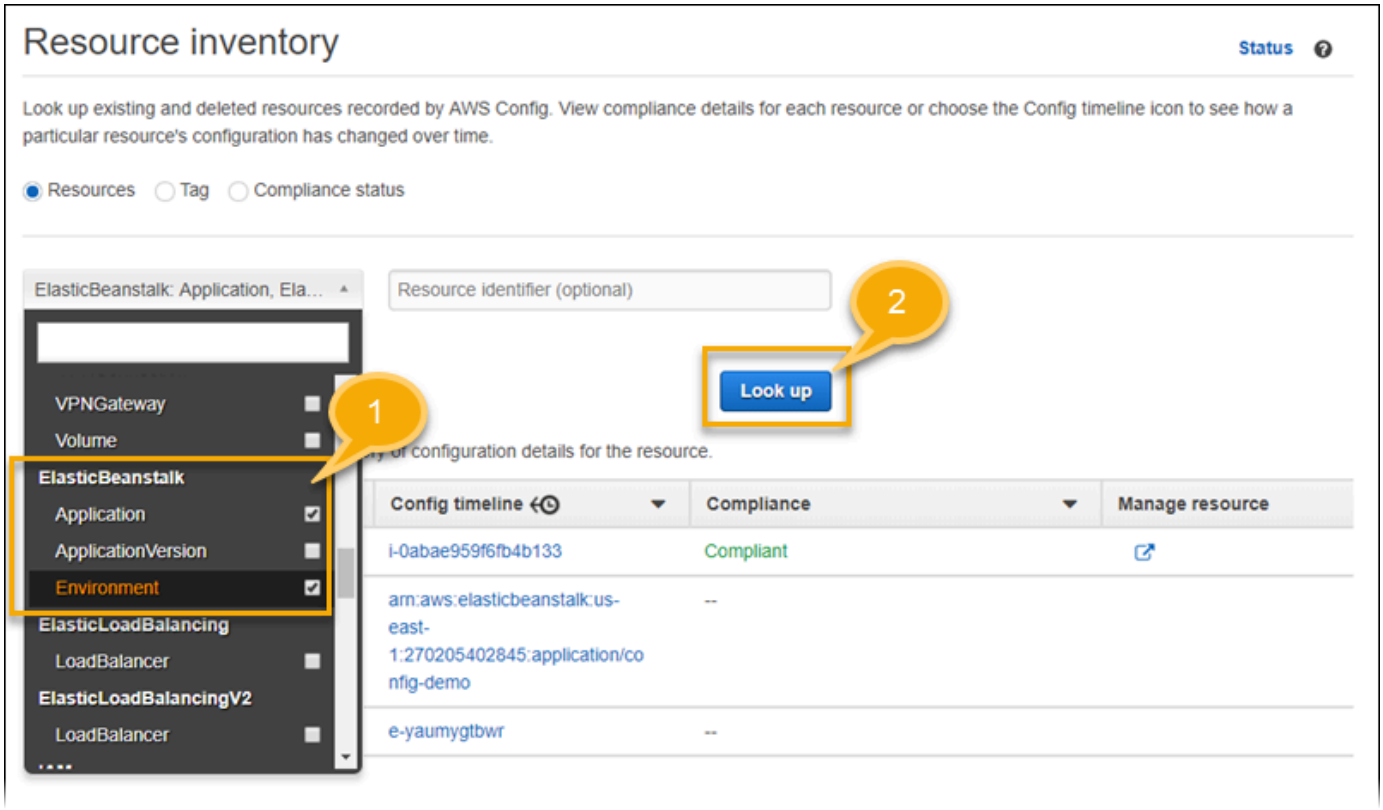
Você pode usar o AWS Config console para procurar recursos do Elastic Beanstalk e obter detalhes atuais e históricos sobre suas configurações. O exemplo a seguir mostra como encontrar informações sobre um ambiente do Elastic Beanstalk.

Para encontrar um ambiente do Elastic Beanstalk no console AWS Config

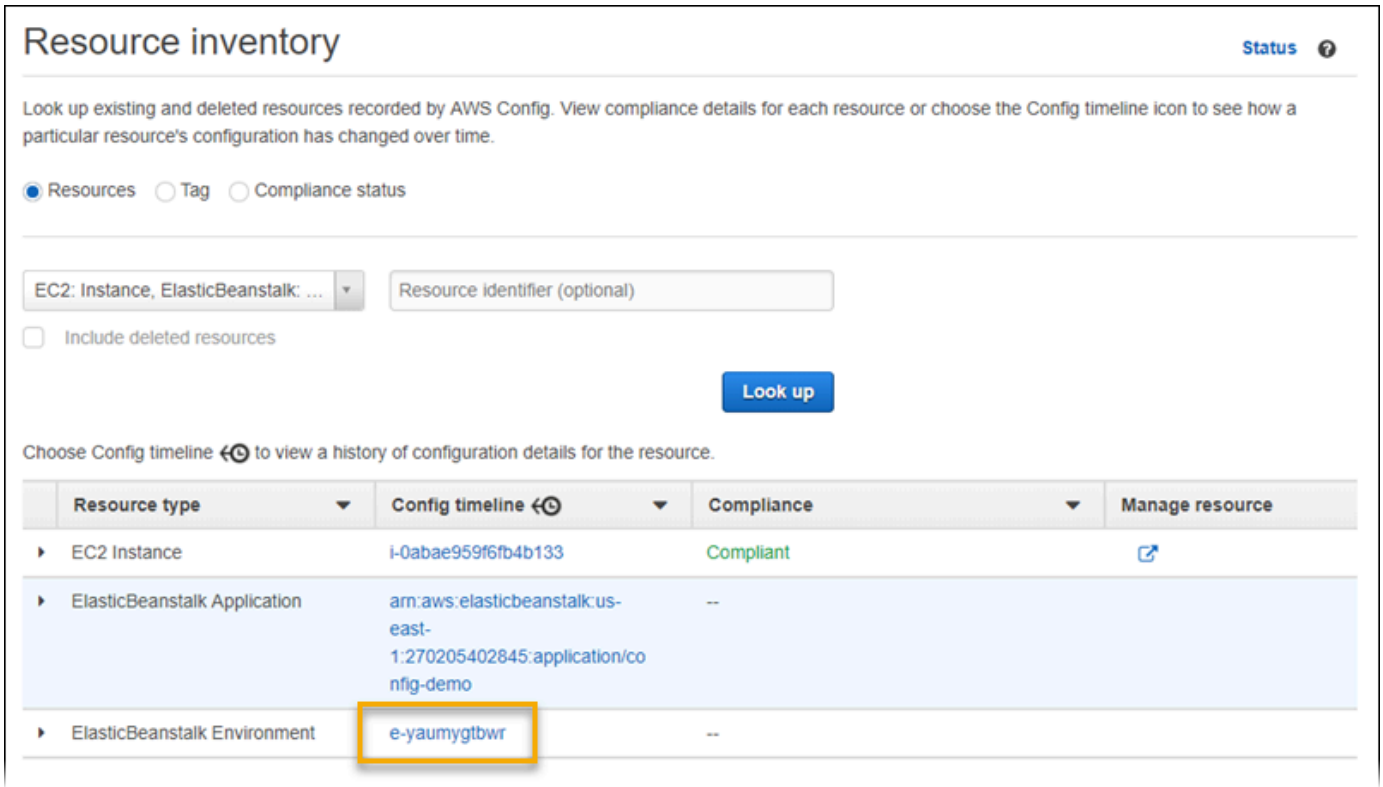
1. Abra o console do AWS Config.
2. Escolha Resources (Recursos).
3. Na página de inventário Resource (Recurso), selecione Resources (Recursos).
4. Abra o menu Tipo de recurso, vá até ElasticBeanstalke escolha um ou mais dos tipos de recursos do Elastic Beanstalk.

> **Note**
>
> Para visualizar detalhes de configuração de outros recursos que o Elastic Beanstalk criou para a aplicação, escolha outros tipos de recursos. Por exemplo, você pode escolher Instância em EC2.

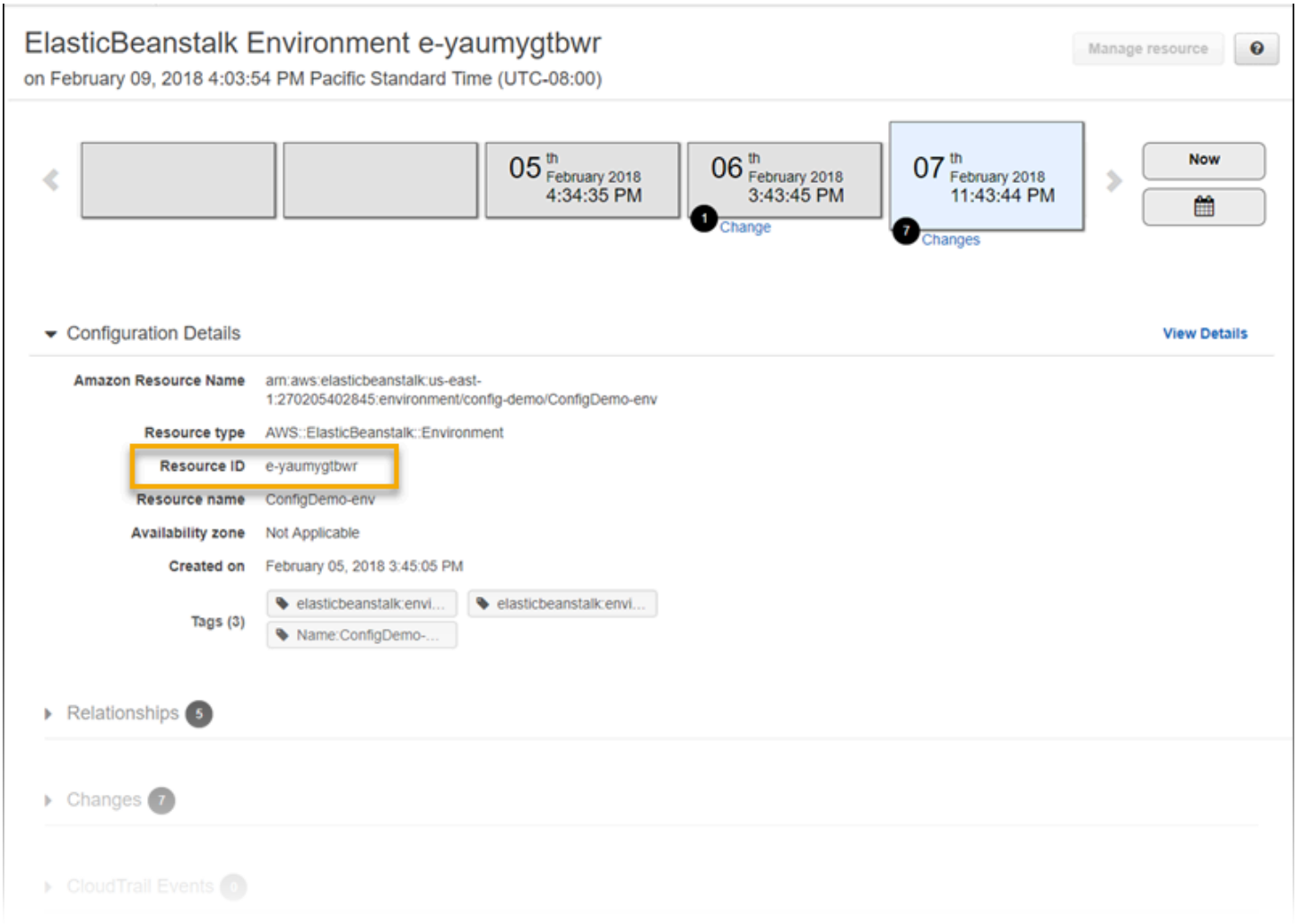
5. Escolha Look up. Veja 2 na figura a seguir.



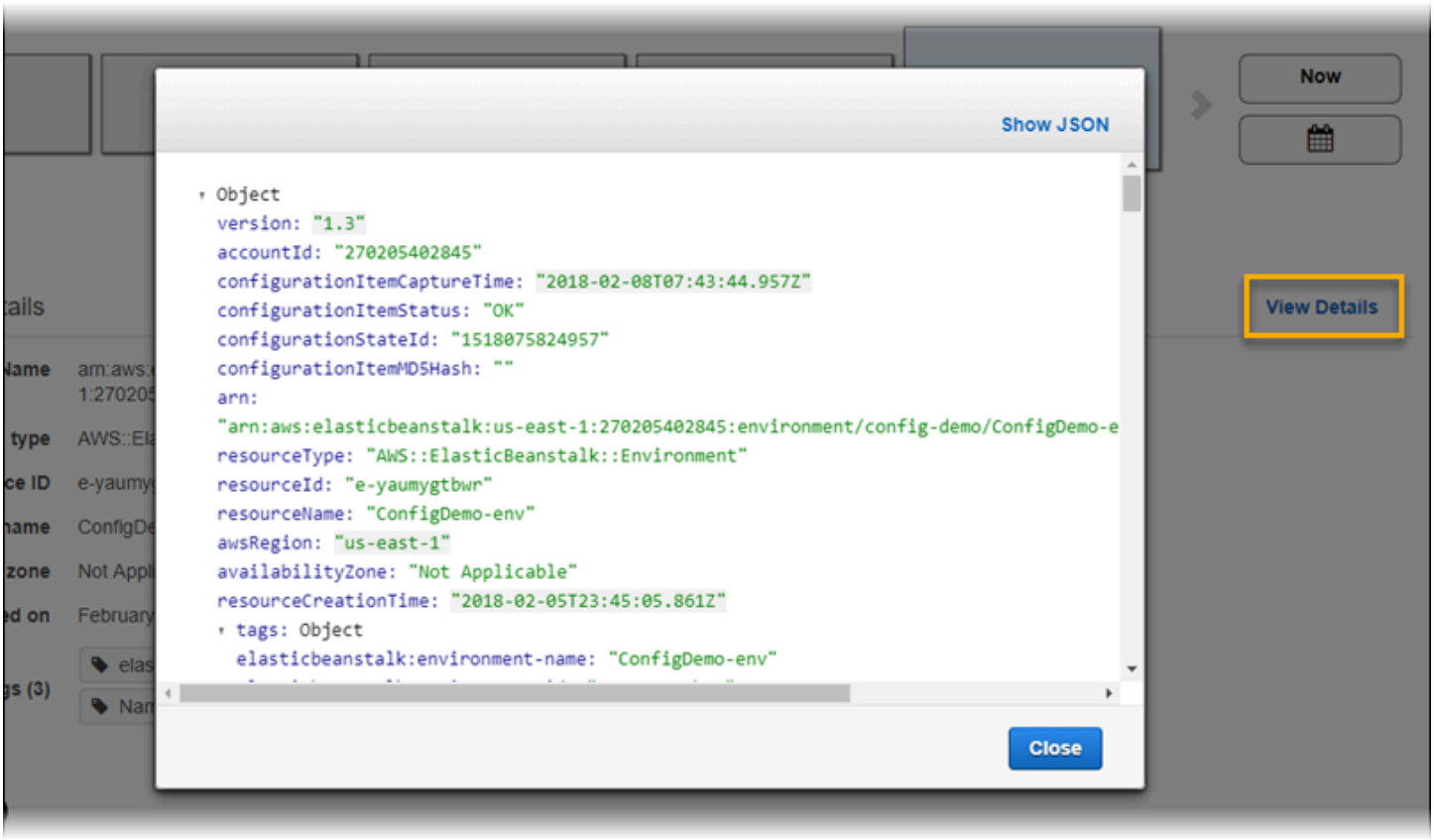
6. Escolha uma ID de recurso na lista de recursos AWS Config exibida.



AWS Config exibe detalhes da configuração e outras informações sobre o recurso selecionado.



Para ver os detalhes completos da configuração registrada, selecione View Details (Exibir detalhes).



Para saber mais maneiras de encontrar um recurso e visualizar as informações nesta página, consulte Exibindo configurações e histórico de AWS recursos no Guia do AWS Config desenvolvedor.

Avaliação dos recursos do Elastic Beanstalk usando regras AWS Config

Você pode criar AWS Config regras, que representam as configurações ideais para seus recursos do Elastic Beanstalk. Você pode usar regras de configuração AWS gerenciada predefinidas ou definir regras personalizadas. AWS Config rastreia continuamente as alterações na configuração de seus recursos para determinar se essas alterações violam alguma das condições em suas regras. O AWS Config console mostra o status de conformidade de suas regras e recursos.

Se um recurso violar uma regra e for sinalizado como não compatível, AWS Config poderá alertá-lo usando um tópico do Amazon Simple Notification Service (Amazon SNS). Para consumir programaticamente os dados desses AWS Config alertas, use uma fila do Amazon Simple Queue Service (Amazon SQS) como ponto final de notificação para o tópico do Amazon SNS. Por exemplo, convém gravar um código que inicia um fluxo de trabalho quando alguém modifica a configuração do grupo de Auto Scaling do seu ambiente.

Para saber mais sobre como configurar e usar regras, consulte Avaliação de recursos com AWS Config regras no Guia do AWS Config desenvolvedor.

Usar o Elastic Beanstalk com o Amazon DynamoDB

O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado que fornece uma performance rápida e previsível com escalabilidade integrada. Se você for um desenvolvedor, poderá usar o DynamoDB para criar uma tabela de banco de dados para armazenar e recuperar qualquer quantidade de dados, bem como atender a todos os níveis de tráfego de solicitação. O DynamoDB distribui automaticamente dados e tráfego para a tabela entre um número suficiente de servidores para controlar a capacidade de solicitação especificada pelo cliente e a quantidade de dados armazenados, enquanto mantém uma performance consistente e rápida. Todos os itens de dados são armazenados em unidades de estado sólido (SSDs) e replicados automaticamente em várias zonas de disponibilidade em uma AWS região para fornecer alta disponibilidade e durabilidade de dados incorporadas.

Se você usa tarefas periódicas em um ambiente de operador, o Elastic Beanstalk cria uma tabela do DynamoDB e a usa para executar a eleição de líder e armazenar informações sobre a tarefa. Cada instância no ambiente tenta gravar na tabela em intervalos de poucos segundos para se tornar líder e realizar a tarefa quando estiver programada.

É possível usar arquivos de configuração para criar uma tabela do DynamoDB para sua aplicação. eb-node-express-sampleVeja um GitHub exemplo de aplicativo Node.js que cria uma tabela com um arquivo de configuração e se conecta a ela com o AWS SDK JavaScript em Node.js. Para obter uma demonstração usando o DynamoDB com PHP, consulte Exemplo: DynamoDB CloudWatch e SNS. Para ver um exemplo que usa o AWS SDK para Java, consulte Gerenciar o estado da sessão do Tomcat com o DynamoDB na documentação. AWS SDK para Java

Ao criar uma tabela do DynamoDB usando os arquivos de configuração, a tabela não se vincula ao ciclo de vida de seu ambiente e não é excluída quando seu ambiente é encerrado. Para que as informações pessoais não sejam retidas desnecessariamente, exclua todos os registros que você não precisa mais ou exclua a tabela.

Para obter mais informações sobre o DynamoDB, consulte o Guia do desenvolvedor do DynamoDB.

Usando o Elastic Beanstalk com a Amazon ElastiCache

ElastiCache A Amazon é um serviço web que permite configurar, gerenciar e escalar ambientes distribuídos de cache na memória na nuvem. Ele fornece um cache na memória de alto

desempenho, escalável e econômico, ao mesmo tempo em que remove a complexidade associada à implantação e ao gerenciamento de um ambiente de cache distribuído. ElastiCache é compatível com o protocolo Redis e Memcached, portanto, o código, os aplicativos e as ferramentas mais populares que você usa atualmente em seus ambientes Redis e Memcached existentes funcionarão perfeitamente com o serviço. Para obter mais informações sobre ElastiCache, acesse a página ElastiCache do produto Amazon.

Para usar o Elastic Beanstalk com a Amazon ElastiCache

1. Crie um ElastiCache cluster.

   - Para obter instruções sobre como criar um ElastiCache cluster com o Redis, acesse Getting Started with Amazon ElastiCache for Redis no Guia do usuário do ElastiCachefor Redis.
   - Para obter instruções sobre como criar um ElastiCache cluster com o Memcached, acesse Getting Started with Amazon ElastiCache for Memcached no Guia do usuário do ElastiCache Memcached.

2. Configure seu grupo ElastiCache de segurança para permitir o acesso do grupo de EC2 segurança da Amazon usado pelo seu aplicativo Elastic Beanstalk. Para obter instruções sobre como encontrar o nome do seu grupo de EC2 segurança usando o AWS Management Console, consulte a EC2 grupos de segurança página do documento EC2 Instâncias.

   - Para obter mais informações sobre o Redis, acesse Autorizar acesso no Guia do usuário do ElastiCache para Redis.
   - Para obter mais informações sobre o Memcached, acesse Autorizar acesso no Guia do usuário do ElastiCache para Memcached.

Você pode usar arquivos de configuração para personalizar seu ambiente do Elastic Beanstalk para uso. ElastiCache Para exemplos de arquivos de configuração que se integram ElastiCache ao Elastic Beanstalk, consulte. Exemplo: ElastiCache.

Usar o Elastic Beanstalk com o Amazon Elastic File System

Com o Amazon Elastic File System (Amazon EFS), você pode criar sistemas de arquivos de rede que podem ser montados por instâncias em várias zonas de disponibilidade. Um sistema de arquivos Amazon EFS é um AWS recurso que usa grupos de segurança para controlar o acesso pela rede que está em sua VPC padrão ou personalizada.

Em um ambiente do Elastic Beanstalk, você pode usar o Amazon EFS para criar um diretório compartilhado que armazena arquivos para a sua aplicação que os usuários carregam e modificam. Sua aplicação pode tratar um volume montado do Amazon EFS como armazenamento local. Assim, você não precisa alterar o código da aplicação para aumentar a escala vertical para várias instâncias.

Para obter mais informações sobre o Amazon EFS, consulte o Guia do usuário do Amazon Elastic File System.

> **Note**
>
> O Elastic Beanstalk cria um usuário de webapp que você pode configurar como proprietário dos diretórios de aplicativos nas instâncias da Amazon. EC2 Para obter mais informações, consulte Armazenamento persistente no tópico Considerações sobre design deste guia.

Seções

- Arquivos de configuração
- Sistemas de arquivos criptografados
- Aplicações de exemplo
- Limpeza dos sistemas de arquivos

Arquivos de configuração

O Elastic Beanstalk fornece arquivos de configuração que pode ser usado para criar e montar sistemas de arquivos do Amazon EFS. Crie um volume do Amazon EFS como parte de seu ambiente ou monte um volume do Amazon EFS que você criou independentemente do Elastic Beanstalk.

- storage-efs-createfilesystem.config — usa a `Resources` chave para criar um novo sistema de arquivos e pontos de montagem no Amazon EFS. Todas as instâncias do ambiente podem se conectar ao mesmo sistema de arquivos para armazenamento escalável compartilhado. Use `storage-efs-mountfilesystem.config` para montar o sistema de arquivos em cada instância.

> **Recursos internos**
>
> Todos os recursos que você cria com os arquivos de configuração são vinculados ao ciclo de vida do ambiente. Se você terminar o ambiente ou remover o arquivo de configuração, esses recursos serão perdidos.

- storage-efs-mountfilesystem.config — Monte um sistema de arquivos Amazon EFS em um caminho local nas instâncias do seu ambiente. Você pode criar o volume como parte do ambiente com `storage-efs-createfilesystem.config`. Ou você pode montá-lo em seu ambiente usando o console ou AWS SDK do Amazon EFS. AWS CLI

Para usar os arquivos de configuração, comece criando o sistema de arquivos do Amazon EFS com `storage-efs-createfilesystem.config`. Siga as instruções do arquivo de configuração e o adicione ao diretório .ebextensions em seu código-fonte para criar o sistema de arquivos na VPC.

Implante o código-fonte atualizado no ambiente do Elastic Beanstalk. Isso é para confirmar se o sistema de arquivos foi criado com êxito. Em seguida, adicione o `storage-efs-mountfilesystem.config` para montar o sistema de arquivos nas instâncias do ambiente. Faça isso em duas implantações separadas para garantir que, se a operação de montagem falhar, o sistema de arquivos permaneça intacto. Se você fizer as duas coisas na mesma implantação, um problema em uma das etapas fará com que o sistema de arquivos seja encerrado quando a implantação falhar.

Sistemas de arquivos criptografados

O Amazon EFS é compatível com sistemas de arquivos criptografados. O arquivo de configuração `storage-efs-createfilesystem.config` discutido neste tópico define duas opções personalizadas. Você pode usar essas opções para criar um sistema de arquivos criptografado do Amazon EFS. Para obter mais informações, consulte as instruções no arquivo de configuração.

Aplicações de exemplo

O Elastic Beanstalk também fornece aplicações demonstrativas que usam o Amazon EFS para armazenamento compartilhado. Os dois projetos têm arquivos de configuração que você pode usar com um instalador padrão WordPress ou do Drupal para executar um blog ou outro sistema de gerenciamento de conteúdo em um ambiente com balanceamento de carga. Quando um usuário carrega fotos ou outras mídias, o arquivo é armazenado em um sistema de arquivos do Amazon

EFS. Isso evita ter que usar a alternativa, que é usar um plugin para armazenar arquivos carregados no Amazon S3.

- Balanceamento de carga WordPress — Isso inclui os arquivos de configuração para instalação WordPress segura e execução em um ambiente Elastic Beanstalk com balanceamento de carga.
- Drupal com carga balanceada: arquivos de configuração e instruções para instalar com segurança o Drupal e executá-lo em um ambiente do Elastic Beanstalk com carga balanceada.

Limpeza dos sistemas de arquivos

Se você criou um sistema de arquivos do Amazon EFS usando um arquivo de configuração como parte do seu ambiente do Elastic Beanstalk, o Elastic Beanstalk removerá o sistema de arquivos quando o ambiente for terminado. Para minimizar os custos de armazenamento de uma aplicação em execução, exclua rotineiramente os objetos dos quais a aplicação não precisa. Ou certifique-se de que o código da aplicação mantenha o ciclo de vida do arquivo corretamente.

> **Important**
>
> Se você criou um sistema de arquivos do Amazon EFS fora de um ambiente do Elastic Beanstalk e montou-o nas instâncias do ambiente, lembre-se de que o Elastic Beanstalk não removerá o sistema de arquivos quando o ambiente for terminado. Para garantir que suas informações pessoais não sejam retidas desnecessariamente e evitar custos de armazenamento, exclua os arquivos que sua aplicação armazenou se não precisar mais deles. Como alternativa, é possível remover todo o sistema de arquivos.

Usando o Elastic Beanstalk com AWS Identity and Access Management

AWS Identity and Access Management (IAM) ajuda você a controlar com segurança o acesso aos seus AWS recursos. Esta seção inclui materiais de referência para trabalhar com políticas do IAM, perfis de instância e funções de serviço.

Para obter uma visão geral de permissões, consulte Perfis do Elastic Beanstalk Service, perfis de instância e políticas de usuário. Na maioria dos ambientes, a função de serviço e o perfil de instância cuja criação é solicitada pelo console do Elastic Beanstalk ao iniciar seu primeiro ambiente têm todas as permissões necessárias. Da mesma maneira, as políticas gerenciadas fornecidas pelo

Elastic Beanstalk para acesso total e acesso somente leitura contêm todas as permissões de usuário necessárias ao uso diário.

O Guia do usuário do IAM fornece uma cobertura detalhada das AWS permissões.

Tópicos

- Gerenciar perfis de instância do Elastic Beanstalk
- Gerenciar funções de serviço do Elastic Beanstalk
- Usar funções vinculadas ao serviço para o Elastic Beanstalk
- Gerenciar políticas de usuário do Elastic Beanstalk
- Formato de nome de recurso da Amazon para o Elastic Beanstalk
- Recursos e condições para ações do Elastic Beanstalk
- Usar tags para controlar o acesso aos recursos do Elastic Beanstalk
- Exemplo de políticas baseadas em políticas gerenciadas
- Políticas de exemplo baseadas em permissões de recursos
- Impedir o acesso ao bucket do Amazon S3 entre ambientes

Gerenciar perfis de instância do Elastic Beanstalk

Um perfil de instância é um contêiner para uma função AWS Identity and Access Management (IAM) que você pode usar para passar informações da função para uma EC2 instância da Amazon quando a instância é iniciada.

Se sua AWS conta não tiver um perfil de EC2 instância, você deverá criar um usando o serviço IAM. Em seguida, você pode atribuir o perfil da EC2 instância aos novos ambientes criados por você. As etapas de criação do ambiente no console do Elastic Beanstalk fornecem acesso ao console do IAM, para que você possa EC2 criar um perfil de instância com as permissões necessárias.

> **Note**
>
> Anteriormente, o Elastic Beanstalk criava EC2 um `aws-elasticbeanstalk-ec2-role` perfil de instância padrão chamado na primeira AWS vez que uma conta criava um ambiente. Esse perfil de instância incluía as políticas gerenciadas padrão. Se sua conta já tiver esse perfil de instância, ele permanecerá disponível para você atribuí-lo aos seus ambientes. No entanto, as diretrizes de AWS segurança recentes não permitem que um AWS serviço crie automaticamente funções com políticas de confiança para outros AWS serviços, EC2

nesse caso. Por causa dessas diretrizes de segurança, o Elastic Beanstalk não cria mais um perfil de instância padrão `aws-elasticbeanstalk-ec2-role`.

Políticas gerenciadas

O Elastic Beanstalk fornece várias políticas gerenciadas para permitir que seu ambiente atenda a diferentes casos de uso. Para atender aos casos de uso padrão de um ambiente, essas políticas devem ser anexadas à função do perfil da EC2 instância.

- AWSElasticBeanstalkWebTier— Concede permissões para que o aplicativo faça upload de registros para o Amazon S3 e informações de depuração para. AWS X-Ray Para ver o conteúdo da política gerenciada, consulte AWSElasticBeanstalkWebTiero Guia de referência de políticas AWS gerenciadas.
- AWSElasticBeanstalkWorkerTier— Concede permissões para upload de registros, depuração, publicação de métricas e tarefas de instância de trabalho, incluindo gerenciamento de filas, eleição de líderes e tarefas periódicas. Para ver o conteúdo da política gerenciada, consulte AWSElasticBeanstalkWorkerTiero Guia de referência de políticas AWS gerenciadas.
- AWSElasticBeanstalkMulticontainerDocker— Concede permissões para o Amazon Elastic Container Service coordenar tarefas de cluster para ambientes Docker. Para ver o conteúdo da política gerenciada, consulte AWSElasticBeanstalkMulticontainerDockero Guia de referência de políticas AWS gerenciadas.

Important

As políticas gerenciadas do Elastic Beanstalk não fornecem permissões granulares, elas concedem todas as permissões que são potencialmente necessárias para trabalhar com aplicações Elastic Beanstalk. Em alguns casos, talvez você queira restringir ainda mais as permissões das nossas políticas gerenciadas. Para um exemplo de um caso de uso, consulte Impedir o acesso ao bucket do Amazon S3 entre ambientes.
Nossas políticas gerenciadas também não abrangem permissões para recursos personalizados que você pode adicionar à sua solução e que não são gerenciados pelo Elastic Beanstalk. Para implementar permissões mais granulares, permissões mínimas necessárias ou permissões de recursos personalizadas, use políticas personalizadas.

Política de relacionamento de confiança para EC2

Para permitir que as EC2 instâncias em seu ambiente assumam a função necessária, o perfil da instância deve especificar a Amazon EC2 como uma entidade confiável na política de relacionamento de confiança, da seguinte forma.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para personalizar as permissões, você pode adicionar políticas à função anexada ao perfil da instância padrão ou criar seu próprio perfil de instância com um conjunto restrito de permissões.

Seções

- Criar um perfil da instância
- Adicionar permissões ao perfil da instância padrão
- Verificar as permissões atribuídas ao perfil de instância
- Atualização de um perfil de instância out-of-date padrão

Criar um perfil da instância

Um perfil de instância é um invólucro em torno de uma função padrão do IAM que permite que uma EC2 instância assuma a função. Você pode criar um perfil de instância com as políticas gerenciadas padrão do Elastic Beanstalk. Você também pode criar perfis de instância adicionais para personalizar permissões para diferentes aplicativos. Ou você pode criar um perfil de instância que não inclua as duas políticas gerenciadas que concedem permissões para ambientes Docker gerenciados por ECS ou de nível de trabalhador, se você não usar esses recursos.

Para criar um perfil de instância com as políticas gerenciadas padrão

1. Abra a página Roles (Funções) no console do IAM.
2. Selecione Create role (Criar função).
3. Em Tipo de entidade confiável, escolha Serviços da AWS .
4. Para serviço ou caso de uso, escolha Elastic Beanstalk.
5. Para o caso de uso, escolha Elastic Beanstalk — Compute.
6. Escolha Próximo.
7. Insira um nome em Nome da função.

   Você pode inserir o nome da função padrão que o console do Elastic Beanstalk sugere:. `aws-elasticbeanstalk-ec2-role`
8. Verifique se as políticas de permissões incluem o seguinte e escolha Avançar:

   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
9. Selecione Criar perfil.

Para criar um perfil de instância com sua escolha específica de políticas gerenciadas

1. Abra a página Roles (Funções) no console do IAM.
2. Selecione Create role (Criar função).
3. Em Tipo de entidade confiável, selecione Serviço da AWS .
4. Em Use case (Caso de uso), escolha EC2.
5. Escolha Próximo.
6. Associe as políticas gerenciadas adequadas disponibilizadas pelo Elastic Beanstalk e quaisquer políticas adicionais que fornecem as permissões necessárias à sua aplicação.
7. Escolha Próximo.
8. Insira um nome para a função.
9. (Opcional) Adicione tags à função.
10. Selecione Criar perfil.

Adicionar permissões ao perfil da instância padrão

Se seu aplicativo acessa recursos AWS APIs ou recursos aos quais as permissões não são concedidas no perfil de instância padrão, adicione políticas que concedam permissões no console do IAM.

Para adicionar políticas à função anexadas ao perfil de instância padrão

1. Abra a página Roles (Funções) no console do IAM.
2. Escolha a função atribuída como seu perfil de EC2 instância.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Selecione a política gerenciada para os serviços adicionais que o seu aplicativo utiliza. Por exemplo, `AmazonS3FullAccess` ou `AmazonDynamoDBFullAccess`.
5. Escolha Anexar política.

Verificar as permissões atribuídas ao perfil de instância

As permissões atribuídas ao seu perfil da instância padrão pode variar de acordo com a data de criação, a última vez em que você iniciou um ambiente e qual cliente você usou. É possível verificar as permissões do perfil da instância padrão no console do IAM.

Para verificar as permissões do perfil da instância padrão

1. Abra a página Roles (Funções) no console do IAM.
2. Escolha a função atribuída como seu perfil de EC2 instância.
3. Na guia Permissions, (Permissões) revise a lista de políticas associadas à função.
4. Para ver as permissões que uma política concede, selecione a política.

Atualização de um perfil de instância out-of-date padrão

Se o perfil de instância padrão não tiver as permissões necessárias, você poderá adicionar manualmente as políticas gerenciadas à função atribuída como seu perfil de EC2 instância.

Para adicionar políticas gerenciadas à função anexadas ao perfil de instância padrão

1. Abra a página Roles (Funções) no console do IAM.
2. Escolha a função atribuída como seu perfil de EC2 instância.

3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Digite **`AWSElasticBeanstalk`** para filtrar as políticas.
5. Selecione as seguintes políticas e selecione Attach policy (Associar política):

   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`

Gerenciar funções de serviço do Elastic Beanstalk

Para gerenciar e monitorar seu ambiente, AWS Elastic Beanstalk executa ações nos recursos do ambiente em seu nome. O Elastic Beanstalk precisa de certas permissões para realizar essas ações e AWS Identity and Access Management assume funções de serviço (IAM) para obter essas permissões.

O Elastic Beanstalk precisa usar credenciais de segurança temporárias sempre que assumir uma função de serviço. Para obter essas credenciais, o Elastic Beanstalk envia uma solicitação para o AWS Security Token Service (AWS STS) em um endpoint específico da região. Para obter mais informações, consulte Credenciais de segurança temporárias no Manual do usuário do IAM.

> **Note**
>
> Se o AWS STS endpoint da região em que seu ambiente está localizado estiver desativado, o Elastic Beanstalk enviará a solicitação em um endpoint alternativo que não pode ser desativado. Este endpoint está associado a uma região diferente. Assim, trata-se de uma solicitação entre regiões. Para obter mais informações, consulte Ativação e desativação AWS STS em uma AWS região no Guia do usuário do IAM.

Gerenciar funções de serviço usando o console do Elastic Beanstalk e a CLI do EB

Você pode usar o console do Elastic Beanstalk e a CLI do EB para configurar funções de serviço para o seu ambiente com um conjunto suficiente de permissões. Eles criam uma função de serviço padrão e usam políticas gerenciadas nela.

Políticas gerenciadas de função de serviços

O Elastic Beanstalk fornece uma política gerenciada para o monitoramento avançado de integridade e outra com permissões adicionais necessárias para atualizações de plataforma gerenciada. O console e a CLI do EB atribuem as políticas à função de serviço padrão criada para você. Essas políticas devem ser usadas somente para esta função de serviço padrão. Elas não devem ser usadas com outros usuários ou funções em suas contas.

`AWSElasticBeanstalkEnhancedHealth`

Esta política concede permissões ao Elastic Beanstalk para monitorar a integridade do ambiente e da instância. Ela também inclui ações do Amazon SQS para permitir que o Elastic Beanstalk monitore a atividade das filas nos ambientes de operador. Para ver o conteúdo dessa política gerenciada, consulte a AWSElasticBeanstalkEnhancedHealthpágina no Guia de referência da política AWS gerenciada.

`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Esta política concede permissões ao Elastic Beanstalk para atualizar ambientes em seu nome a fim de realizar atualizações gerenciadas de plataforma. Para ver o conteúdo dessa política gerenciada, consulte a AWSElasticBeanstalkManagedUpdatesCustomerRolePolicypágina no Guia de referência da política AWS gerenciada.

Agrupamentos de permissão no nível de serviço

Esta política é agrupada em declarações com base no conjunto de permissões fornecidas.

- *`ElasticBeanstalkPermissions`*— Esse grupo de permissões serve para chamar as ações de serviço do Elastic Beanstalk (Elastic Beanstalk). APIs
- *`AllowPassRoleToElasticBeanstalkAndDownstreamServices`* - Esse grupo de permissões permite que qualquer função seja transferida para o Elastic Beanstalk e para outros serviços downstream, como o AWS CloudFormation.
- *`ReadOnlyPermissions`*: esse grupo de permissões destina-se a coletar informações sobre o ambiente em execução.
- *`*OperationPermissions`*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma.
- *`*BroadOperationPermissions`*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma. Eles também incluem permissões amplas para oferecer suporte a ambientes herdados.

- *`*TagResource`*— Os grupos com esse padrão de nomenclatura são para chamadas que usam o tag-on-create APIs para anexar tags em recursos que estão sendo criados em um ambiente do Elastic Beanstalk.

Para visualizar o conteúdo de uma política gerenciada, você também pode usar a página Políticas no console do IAM.

> **Important**
>
> As políticas gerenciadas do Elastic Beanstalk não fornecem permissões granulares, elas concedem todas as permissões que são potencialmente necessárias para trabalhar com aplicações Elastic Beanstalk. Em alguns casos, talvez você queira restringir ainda mais as permissões das nossas políticas gerenciadas. Para um exemplo de um caso de uso, consulte Impedir o acesso ao bucket do Amazon S3 entre ambientes.
> Nossas políticas gerenciadas também não abrangem permissões para recursos personalizados que você pode adicionar à sua solução e que não são gerenciados pelo Elastic Beanstalk. Para implementar permissões mais granulares, permissões mínimas necessárias ou permissões de recursos personalizadas, use políticas personalizadas.

Políticas gerenciadas pela obsoletas

No passado, o Elastic Beanstalk AWSElasticBeanstalkServiceapoiava a política de função de serviço gerenciado. Esta política foi substituída por AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy. Talvez você ainda consiga ver e usar a política anterior no console do IAM.

Para ver o conteúdo da política gerenciada, consulte AWSElasticBeanstalkServiceo Guia de referência de políticas AWS gerenciadas.

No entanto, recomendamos que você faça a transição para o uso da nova política gerenciada (AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy). Adicione políticas personalizadas para conceder permissões a recursos personalizados, se tiver alguma.

Usar o console do Elastic Beanstalk

Ao executar um ambiente no console do Elastic Beanstalk, o console cria uma função de serviço padrão chamada `aws-elasticbeanstalk-service-role` e anexa políticas gerenciadas com permissões padrão a essa função de serviço.

Para permitir que o Elastic Beanstalk assuma a função `aws-elasticbeanstalk-service-role`, a função de serviço especifica o Elastic Beanstalk como uma entidade confiável na política de relacionamento de confiança.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
          "Service": "elasticbeanstalk.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
            "sts:ExternalId": "elasticbeanstalk"
          }
        }
      }
    ]
}
```

Ao habilitar atualizações gerenciadas de plataforma para o ambiente, o Elastic Beanstalk assume uma função de serviço separada para executar atualizações gerenciadas. Por padrão, o console do Elastic Beanstalk usa a mesma função de serviço gerada, `aws-elasticbeanstalk-service-role`, para a função de serviço de atualizações gerenciadas. Se você alterar a função de serviço padrão, o console definirá a função de serviço de atualizações gerenciadas para usar a função vinculada ao serviço de atualizações gerenciadas, `AWSServiceRoleForElasticBeanstalkManagedUpdates`. Para obter mais informações sobre funções vinculadas ao serviço, consulte the section called “Uso de perfis vinculados ao serviço”.

> **Note**
>
> Em função de problemas de permissão, o serviço do Elastic Beanstalk nem sempre consegue criar essa função vinculada ao serviço para você com sucesso. Portanto, o console tenta criá-lo explicitamente. Para garantir que sua conta tenha essa função vinculada ao serviço, crie um ambiente pelo menos uma vez usando o console e configure as atualizações gerenciadas para serem habilitadas antes da criação do ambiente.

Uso da EB CLI

Se você executar um ambiente usando o comando the section called “eb create” da Interface de linhas de comando do Elastic Beanstalk (CLI do EB) e não especificar uma função de serviço por meio da opção `--service-role`, o Elastic Beanstalk cria uma função de serviço padrão `aws-elasticbeanstalk-service-role`. Se a função de serviço padrão já existir, o Elastic Beanstalk a usará para o novo ambiente. Nessas situações, o console do Elastic Beanstalk também executa ações semelhantes.

Diferente de como é no console, você não pode especificar uma função de serviço de atualizações gerenciadas usando uma opção de comando da CLI do EB. Se você habilitar atualizações gerenciadas para o seu ambiente, deverá definir a função de serviço de atualizações gerenciadas nas opções de configuração. O exemplo a seguir habilita atualizações gerenciadas e usa a função de serviço padrão como uma função de serviço de atualizações gerenciadas.

Example .ebextensions/ .config managed-platform-update

```
option_settings:
  aws:elasticbeanstalk:managedactions:
    ManagedActionsEnabled: true
    PreferredStartTime: "Tue:09:00"
    ServiceRoleForManagedUpdates: "aws-elasticbeanstalk-service-role"
  aws:elasticbeanstalk:managedactions:platformupdate:
    UpdateLevel: patch
    InstanceRefreshEnabled: true
```

Gerenciar funções de serviço usando a API do Elastic Beanstalk

Ao usar a ação `CreateEnvironment` da API do Elastic Beanstalk para criar um ambiente, especifique uma função de serviço usando a opção de configuração `ServiceRole` no namespace `aws:elasticbeanstalk:environment`. Para obter mais informações sobre como usar o monitoramento avançado de integridade com a API do Elastic Beanstalk, consulte Usar relatórios de integridade aprimorada com a API do Elastic Beanstalk.

Além disso, se você habilitar as atualizações gerenciadas de plataforma para o ambiente, poderá especificar uma função de serviço de atualizações gerenciadas usando a opção `ServiceRoleForManagedUpdates` do namespace `aws:elasticbeanstalk:managedactions`.

Uso de perfis vinculados ao serviço

Uma função vinculada ao serviço é um tipo exclusivo de função de serviço predefinida pelo Elastic Beanstalk para incluir todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS A função vinculada a serviço é associada à sua conta. O Elastic Beanstalk cria essa função uma única vez e depois a reutiliza ao criar ambientes adicionais. Para obter mais informações sobre o uso de funções vinculadas a serviços com ambientes do Elastic Beanstalk, consulte Usar funções vinculadas ao serviço para o Elastic Beanstalk.

Se você criar um ambiente usando a API do Elastic Beanstalk e não especificar uma função de serviço, o Elastic Beanstalk cria uma função vinculada ao serviço de monitoramento para a sua conta, se ainda não houver uma. O Elastic Beanstalk usa essa função para o novo ambiente. Você também pode usar o IAM para criar uma função vinculada ao serviço de monitoramento para a sua conta com antecedência. Depois que a sua conta tiver essa função, você pode usá-la para criar um ambiente usando a API do Elastic Beanstalk, o console do Elastic Beanstalk ou a CLI do EB.

Se você habilitar atualizações de plataforma gerenciadas para o ambiente e especificar `AWSServiceRoleForElasticBeanstalkManagedUpdates` como o valor da opção `ServiceRoleForManagedUpdates` do namespace `aws:elasticbeanstalk:managedactions`, o Elastic Beanstalk criará uma função vinculada ao serviço de atualizações gerenciadas para a conta, se ainda não houver uma. O Elastic Beanstalk usa a função para executar atualizações gerenciadas para o novo ambiente.

> **Note**
>
> Quando o Elastic Beanstalk tenta criar funções vinculadas ao serviço de monitoramento e atualizações gerenciadas para sua conta quando um ambiente é criado, é necessário ter a permissão `iam:CreateServiceLinkedRole`. Se você não tiver essa permissão, haverá falhas na criação do ambiente e uma mensagem explicando o problema será exibida. Como alternativa, outro usuário com permissão para criar funções vinculadas ao serviço pode usar o IAM para criá-las com antecedência. Com esse método, você não precisa da permissão `iam:CreateServiceLinkedRole` para criar o seu ambiente.

Como verificar as permissões de funções de serviço padrão

As permissões concedidas pela sua função de serviço padrão variam com base na sua data de criação, na última vez que você executou um ambiente e em qual cliente você usou. No console do IAM, você pode verificar as permissões concedidas pela função de serviço padrão.

Para verificar as permissões da função de serviço padrão

1. No console do IAM, abra a página Roles (Funções).
2. Selecione aws-elasticbeanstalk-service-role.
3. Na guia Permissions, (Permissões) revise a lista de políticas associadas à função.
4. Para visualizar as permissões que uma política concede, selecione a política.

Atualização de uma função de serviço out-of-date padrão

Se a função de serviço padrão não tiver as permissões necessárias, você poderá atualizá-la criando um novo ambiente no console de gerenciamento de ambiente do Elastic Beanstalk.

Como alternativa, você pode adicionar manualmente as políticas gerenciadas à função de serviço padrão.

Para adicionar políticas gerenciadas à função de serviço padrão

1. No console do IAM, abra a página Roles (Funções).
2. Selecione aws-elasticbeanstalk-service-role.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Insira **`AWSElasticBeanstalk`** para filtrar as políticas.
5. Selecione as seguintes políticas e selecione Attach policy (Associar política):
   - `AWSElasticBeanstalkEnhancedHealth`
   - `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Adicionar permissões à função de serviço padrão

Se seu aplicativo incluir arquivos de configuração que se referem a AWS recursos cujas permissões não estão incluídas na função de serviço padrão, o Elastic Beanstalk pode precisar de permissões adicionais. Tais permissões extras são necessárias para resolver essas referências quando elas processam os arquivos de configuração durante uma atualização gerenciada. Em caso de ausência das permissões, a atualização falha e o Elastic Beanstalk retorna uma mensagem indicando quais permissões ele precisa. Siga estas etapas para adicionar permissões para serviços adicionais à função de serviço padrão no console do IAM.

Para adicionar outras políticas à função de serviço padrão

1. No console do IAM, abra a página Roles (Funções).
2. Selecione aws-elasticbeanstalk-service-role.
3. Na guia Permissions (Permissões), escolha Attach policies (Anexar políticas).
4. Selecione a política gerenciada para os serviços adicionais que o seu aplicativo utiliza. Por exemplo, `AmazonAPIGatewayAdministrator` ou `AmazonElasticFileSystemFullAccess`.
5. Escolha Anexar política.

Criar um perfil de serviço

Se você não pode usar a função de serviço padrão, crie uma função de serviço.

Para criar um perfil de serviço

1. No console do IAM, abra a página Roles (Funções).
2. Selecione Criar perfil.
3. Em Produto da AWS , escolha AWS Elastic Beanstalk, e selecione o caso de uso.
4. Escolha Próximo: Permissões.
5. Associe as políticas gerenciadas `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` e `AWSElasticBeanstalkEnhancedHealth` e quaisquer políticas adicionais que fornecem as permissões das quais seu aplicativo precisa.
6. Escolha Próximo: tags.
7. (Opcional) Adicione tags à função.
8. Escolha Próximo: revisar.
9. Insira um nome para a função.
10. Selecione Criar perfil.

Aplique a sua função de serviço personalizada ao criar um ambiente usando o assistente de criação de ambiente ou com a opção `--service-role` para o comando `eb create`.

Usar funções vinculadas ao serviço para o Elastic Beanstalk

AWS Elastic Beanstalk usa funções vinculadas ao serviço AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

O Elastic Beanstalk define alguns tipos de funções vinculadas ao serviço:

- Monitoramento da função vinculada ao serviço: permite que o Elastic Beanstalk monitore a integridade dos ambientes em execução e publique notificações de eventos de integridade.
- Função vinculada ao serviço de manutenção: permite que o Elastic Beanstalk realize atividades regulares de manutenção para seus ambientes em execução.
- Função vinculada ao serviço de atualizações gerenciadas: permite que o Elastic Beanstalk realize atualizações agendadas da plataforma dos ambientes em execução.

Tópicos

- A função de função vinculada ao serviço de monitoramento
- A função vinculada ao serviço de manutenção
- A função vinculada ao serviço de atualizações gerenciadas

A função de função vinculada ao serviço de monitoramento

AWS Elastic Beanstalk usa funções vinculadas ao serviço AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

Uma função vinculada ao serviço facilita a configuração do Elastic Beanstalk, já que não é preciso adicionar as permissões necessárias manualmente. O Elastic Beanstalk define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido em contrário, somente o Elastic Beanstalk pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Elastic Beanstalk porque não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Permissões de função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk usa a função vinculada ao serviço AWSServiceRoleForElasticBeanstalkchamada — Permite que o Elastic Beanstalk monitore a integridade dos ambientes em execução e publique notificações de eventos de saúde.

A função AWSService RoleForElasticBeanstalk vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `elasticbeanstalk.amazonaws.com`

A política de permissões da função AWSService RoleForElasticBeanstalk vinculada ao serviço contém todas as permissões que o Elastic Beanstalk precisa para concluir ações em seu nome:

AllowCloudformationReadOperationsOnElasticBeanstalkStacks

```
{
    "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": [
                "*"
            ]
        }
    ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Como alternativa, você pode usar uma política AWS gerenciada para fornecer acesso total ao Elastic Beanstalk.

Criar uma função vinculada ao serviço para o Elastic Beanstalk

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhuma função de serviço é especificada, o Elastic Beanstalk cria a função vinculada ao serviço para você.

> **Important**
>
> Se você estava usando o serviço Elastic Beanstalk antes de 27 de setembro de 2017, quando ele começou a AWSService RoleForElasticBeanstalk oferecer suporte à função vinculada ao serviço, e sua conta precisava dele, o Elastic Beanstalk criou a função em

sua conta. AWSService RoleForElasticBeanstalk Para saber mais, consulte A New Role Appeared in My IAM Account.

Quando o Elastic Beanstalk tenta AWSService RoleForElasticBeanstalk criar a função vinculada ao serviço para sua conta ao criar um ambiente, você deve ter a permissão. `iam:CreateServiceLinkedRole` Se você não tiver essa permissão, ocorrerá falha na criação do ambiente e você verá uma mensagem explicando o problema.

Como alternativa, outro usuário com permissão para criar funções vinculadas ao serviço pode usar o IAM para pré-criá-las com antecedência. Você poderá criar seu ambiente, mesmo sem ter a permissão `iam:CreateServiceLinkedRole`.

Você (ou outro usuário) pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso do Elastic Beanstalk. Na CLI ou na API do IAM, crie uma função vinculada ao serviço com o nome de serviço `elasticbeanstalk.amazonaws.com`. Para obter mais informações, consulte Criar uma função vinculada ao serviço no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhuma função de serviço é especificada, o Elastic Beanstalk cria novamente a função vinculada ao serviço para você.

Editar uma função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk não permite que você edite a função vinculada ao serviço. AWSService RoleForElasticBeanstalk Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada a serviço para o Elastic Beanstalk

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

Limpar um perfil vinculado ao serviço

Para poder usar o IAM para excluir uma função vinculada ao serviço, primeiro é necessário confirmar se todos os ambientes do Elastic Beanstalk estão usando uma função de serviço diferente ou se foram encerrados.

> **Note**
>
> Se o serviço Elastic Beanstalk estiver usando a função vinculada ao serviço quando você tentar encerrar os ambientes, o encerramento poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para encerrar um ambiente do Elastic Beanstalk que usa o (console) AWSService RoleForElasticBeanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Consulte eb terminate para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a CLI do EB.

Consulte TerminateEnvironmentpara obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a API.

Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a CLI do IAM ou a API do IAM para excluir a função vinculada ao AWSService RoleForElasticBeanstalk serviço. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do Elastic Beanstalk

O Elastic Beanstalk é compatível com funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Elastic Beanstalk.

A função vinculada ao serviço de manutenção

AWS Elastic Beanstalk usa funções vinculadas ao serviço AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

Uma função vinculada ao serviço facilita a configuração do Elastic Beanstalk, já que não é preciso adicionar as permissões necessárias manualmente. O Elastic Beanstalk define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido em contrário, somente o Elastic Beanstalk pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Elastic Beanstalk porque não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Permissões de função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk usa a função vinculada ao serviço AWSServiceRoleForElasticBeanstalkMaintenancechamada — Permite que o Elastic Beanstalk realize atividades regulares de manutenção em seus ambientes em execução.

A função AWSService RoleForElasticBeanstalkMaintenance vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `maintenance.elasticbeanstalk.amazonaws.com`

A política de permissões da função AWSService RoleForElasticBeanstalkMaintenance vinculada ao serviço contém todas as permissões que o Elastic Beanstalk precisa para concluir ações em seu nome:

```
{
    "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-*"
            ]
        }
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Como alternativa, você pode usar uma política AWS gerenciada para fornecer acesso total ao Elastic Beanstalk.

Criar uma função vinculada ao serviço para o Elastic Beanstalk

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhum perfil de instância é especificado, o Elastic Beanstalk cria a função vinculada ao serviço para você.

> **Important**
>
> Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Se você estava usando o serviço Elastic Beanstalk antes de 18 de abril de 2019, quando ele começou a AWSService RoleForElasticBeanstalkMaintenance oferecer suporte à função vinculada ao serviço, e sua conta precisava dele, o Elastic Beanstalk criou a função em sua conta. AWSService RoleForElasticBeanstalkMaintenance Para saber mais, consulte Uma Nova Função Apareceu na minha Conta do IAM.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você pode usar esse mesmo processo para recriar a função na sua conta. Quando um ambiente do Elastic Beanstalk é criado usando a API do Elastic Beanstalk e nenhum perfil de instância é especificado, o Elastic Beanstalk cria a função novamente vinculada ao serviço para você.

Editar uma função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk não permite que você edite a função vinculada ao serviço. AWSService RoleForElasticBeanstalkMaintenance Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada a serviço para o Elastic Beanstalk

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir uma função vinculada ao serviço, primeiro é necessário encerrar todos os ambientes do Elastic Beanstalk que usam a função.

> **Note**
>
> Se o serviço Elastic Beanstalk estiver usando a função vinculada ao serviço quando você tentar encerrar os ambientes, o encerramento poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para encerrar um ambiente do Elastic Beanstalk que usa o (console) AWSService RoleForElasticBeanstalkMaintenance

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Terminate Environment (Encerrar ambiente).

4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Consulte eb terminate para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a CLI do EB.

Consulte TerminateEnvironmentpara obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a API.

Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a CLI do IAM ou a API do IAM para excluir a função vinculada ao AWSService RoleForElasticBeanstalkMaintenance serviço. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do Elastic Beanstalk

O Elastic Beanstalk é compatível com funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Elastic Beanstalk.

A função vinculada ao serviço de atualizações gerenciadas

AWS Elastic Beanstalk usa funções vinculadas ao serviço AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Elastic Beanstalk. As funções vinculadas ao serviço são predefinidas pelo Elastic Beanstalk e incluem todas as permissões que o serviço exige para chamar outros serviços em seu nome. AWS

Uma função vinculada ao serviço facilita a configuração do Elastic Beanstalk, já que não é preciso adicionar as permissões necessárias manualmente. O Elastic Beanstalk define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido em contrário, somente o Elastic Beanstalk pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Elastic Beanstalk porque não é possível remover por engano a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contenham Sim na

coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Permissões de função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk usa a função vinculada ao serviço AWSServiceRoleForElasticBeanstalkManagedUpdateschamada — Permite que o Elastic Beanstalk realize atualizações programadas da plataforma de seus ambientes em execução.

A função AWSService RoleForElasticBeanstalkManagedUpdates vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `managedupdates.elasticbeanstalk.amazonaws.com`

A política gerenciada AWSElasticBeanstalkManagedUpdatesServiceRolePolicyconcede AWSService RoleForElasticBeanstalkManagedUpdates à função vinculada ao serviço todas as permissões que o Elastic Beanstalk precisa para concluir as ações de atualização gerenciada em seu nome. Para ver o conteúdo da política gerenciada, consulte a AWSElasticBeanstalkManagedUpdatesServiceRolePolicypágina no Guia de referência de políticas AWS gerenciadas.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Como alternativa, você pode usar uma política AWS gerenciada para fornecer acesso total ao Elastic Beanstalk.

Criar uma função vinculada ao serviço para o Elastic Beanstalk

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria um ambiente do Elastic Beanstalk com a sua respectiva API, habilita atualizações gerenciadas e especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como o valor para a opção `ServiceRoleForManagedUpdates` do namespace `aws:elasticbeanstalk:managedactions`, o Elastic Beanstalk cria a função novamente vinculada ao serviço para você.

Quando o Elastic Beanstalk tenta AWSService RoleForElasticBeanstalkManagedUpdates criar a função vinculada ao serviço para sua conta ao criar um ambiente, você deve ter a permissão.

`iam:CreateServiceLinkedRole` Se você não tiver essa permissão, ocorrerá falha na criação do ambiente e você verá uma mensagem explicando o problema.

Como alternativa, outro usuário com permissão para criar funções vinculadas ao serviço pode usar o IAM para pré-criá-las com antecedência. Você poderá criar seu ambiente, mesmo sem ter a permissão `iam:CreateServiceLinkedRole`.

Você (ou outro usuário) pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso Elastic Beanstalk Managed Updates (Atualizações gerenciadas do Elastic Beanstalk). Na CLI ou na API do IAM, crie uma função vinculada ao serviço com o nome de serviço `managedupdates.elasticbeanstalk.amazonaws.com`. Para obter mais informações, consulte Criar uma função vinculada ao serviço no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um ambiente do Elastic Beanstalk com a sua respectiva API, habilita atualizações gerenciadas e especifica `AWSServiceRoleForElasticBeanstalkManagedUpdates` como o valor para a opção `ServiceRoleForManagedUpdates` do namespace `aws:elasticbeanstalk:managedactions`, o Elastic Beanstalk cria novamente a função novamente vinculada ao serviço para você.

Editar uma função vinculada ao serviço para o Elastic Beanstalk

O Elastic Beanstalk não permite que você edite a função vinculada ao serviço. AWSService RoleForElasticBeanstalkManagedUpdates Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada a serviço para o Elastic Beanstalk

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

Limpar um perfil vinculado ao serviço

Para poder usar o IAM para excluir uma função vinculada ao serviço, primeiro é necessário confirmar se os ambientes com atualizações gerenciadas habilitadas do Elastic Beanstalk estão usando uma função de serviço diferente ou se foram encerrados.

> **Note**
>
> Se o serviço Elastic Beanstalk estiver usando a função vinculada ao serviço quando você tentar encerrar os ambientes, o encerramento poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para encerrar um ambiente do Elastic Beanstalk que usa o (console) AWSService RoleForElasticBeanstalkManagedUpdates

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions e, em seguida, Terminate Environment.
4. Use a caixa de diálogo na tela para confirmar o encerramento do ambiente.

Consulte eb terminate para obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a CLI do EB.

Consulte TerminateEnvironmentpara obter detalhes sobre como encerrar um ambiente do Elastic Beanstalk usando a API.

Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a CLI do IAM ou a API do IAM para excluir a função vinculada ao AWSService RoleForElasticBeanstalkManagedUpdates serviço. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do Elastic Beanstalk

O Elastic Beanstalk é compatível com funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Endpoints e cotas do AWS Elastic Beanstalk.

Gerenciar políticas de usuário do Elastic Beanstalk

AWS Elastic Beanstalk fornece duas políticas gerenciadas que permitem que você atribua acesso total ou acesso somente de leitura a todos os recursos gerenciados pelo Elastic Beanstalk. Você pode anexar as políticas a usuários ou grupos AWS Identity and Access Management (IAM) ou a funções assumidas por seus usuários.

Políticas de usuário gerenciadas

- AdministratorAccess- AWSElastic Beanstalk — concede ao usuário permissões administrativas completas para criar, modificar e excluir aplicativos, versões de aplicativos, definições de configuração, ambientes e seus recursos subjacentes do Elastic Beanstalk. Para ver o conteúdo da política gerenciada, consulte a página AdministratorAccess- AWSElastic Beanstalk no Guia de referência de políticas AWS gerenciadas.
- AWSElasticBeanstalkReadOnly— Permite que o usuário visualize aplicativos e ambientes, mas não realize operações que os modifiquem. Ele fornece acesso somente de leitura a todos os recursos do Elastic Beanstalk e a outros recursos AWS que o console do Elastic Beanstalk recupera. Observe que o acesso somente leitura não permite ações como fazer download de logs do Elastic Beanstalk para que você possa lê-los. Isso ocorre porque os logs são preparados no bucket do Amazon S3, em que o Elastic Beanstalk exige permissão para gravação. Consulte o exemplo no fim deste tópico para obter informações sobre como habilitar o acesso aos logs do Elastic Beanstalk. Para ver o conteúdo da política gerenciada, consulte a AWSElasticBeanstalkReadOnlypágina no Guia de referência de políticas AWS gerenciadas.

> **Important**
>
> As políticas gerenciadas do Elastic Beanstalk não fornecem permissões granulares, elas concedem todas as permissões que são potencialmente necessárias para trabalhar com aplicações Elastic Beanstalk. Em alguns casos, talvez você queira restringir ainda mais as permissões das nossas políticas gerenciadas. Para um exemplo de um caso de uso, consulte Impedir o acesso ao bucket do Amazon S3 entre ambientes.
> Nossas políticas gerenciadas também não abrangem permissões para recursos personalizados que você pode adicionar à sua solução e que não são gerenciados pelo Elastic Beanstalk. Para implementar permissões mais granulares, permissões mínimas necessárias ou permissões de recursos personalizadas, use políticas personalizadas.

Políticas gerenciadas pela obsoletas

Anteriormente, o Elastic Beanstalk dava suporte a duas outras políticas de usuários gerenciados, e. AWSElasticBeanstalkFullAccessAWSElasticBeanstalkReadOnlyAccess Planejamos aposentar essas políticas anteriores. Talvez você ainda consiga vê-las e usá-las no console do IAM. No entanto, recomendamos que você faça a transição para o uso das novas políticas de usuário gerenciadas e adicione políticas personalizadas para conceder permissões a recursos personalizados, se você tiver algum.

Políticas de integração com outros serviços

Também fornecemos políticas mais granulares que permitem integrar seu ambiente a outros serviços, se você preferir usá-los.

- AWSElasticBeanstalkRoleCWL — Permite que um ambiente gerencie grupos de CloudWatch logs do Amazon Logs.
- AWSElasticBeanstalkRoleRDS — Permite que um ambiente integre uma instância do Amazon RDS.
- AWSElasticBeanstalkRoleWorkerTier— Permite que um nível de ambiente de trabalho crie uma tabela do Amazon DynamoDB e uma fila do Amazon SQS.
- AWSElasticBeanstalkRoleECS — Permite que um ambiente Docker de vários contêineres gerencie clusters do Amazon ECS.
- AWSElasticBeanstalkRoleCore— Permite as operações principais de um ambiente de serviços web.
- AWSElasticBeanstalkRoleSNS — Permite que um ambiente habilite a integração de tópicos do Amazon SNS.

Para ver a origem JSON de uma política gerenciada específica, consulte o Guia de referência de políticas gerenciadas do AWS .

Controlar o acesso com políticas gerenciadas

É possível usar as políticas gerenciadas para conceder acesso total ou somente leitura ao Elastic Beanstalk. O Elastic Beanstalk atualiza essas políticas automaticamente quando outras permissões são necessárias para acessar novos recursos.

Como aplicar uma política gerenciada a grupos ou usuários do IAM

1. Abra a página Policies (Políticas) no console do IAM.
2. Na caixa de pesquisa, digite **`AWSElasticBeanstalk`** para filtrar as políticas.
3. Na lista de políticas, marque a caixa de seleção ao lado de AWSElasticBeanstalkReadOnlyou AdministratorAccess- AWSElastic Beanstalk.
4. Selecione Ações da política e escolha Anexar.
5. Selecione um ou mais usuários e grupos aos quais associar a política. Você pode usar o menu Filter (Filtro) e a caixa de pesquisa para filtrar a lista de entidades principais.
6. Escolha Anexar política.

Criar uma política de usuário personalizada

Você pode criar sua própria política do IAM para permitir ou negar ações específicas da API do Elastic Beanstalk em recursos específicos do Elastic Beanstalk e controlar o acesso a recursos personalizados que não são gerenciados pelo Elastic Beanstalk. Para obter mais informações sobre como anexar uma política a um usuário ou grupo, consulte Trabalhar com políticas no Guia do usuário do IAM. Para obter detalhes sobre a criação de uma política personalizada, consulte Criar políticas do IAM no Guia do usuário do IAM.

> **Note**
>
> Embora você possa restringir a forma como um usuário interage com o APIs Elastic Beanstalk, atualmente não há uma forma eficaz de impedir que usuários com permissão para criar os recursos subjacentes necessários criem outros recursos na Amazon e em outros serviços. EC2
> Considere essas políticas como uma forma eficiente para distribuir as responsabilidades do Elastic Beanstalk, e não como uma maneira de proteger todos os recursos subjacentes.

> **Important**
>
> Se você tiver políticas personalizadas atribuídas a uma função de serviço do Elastic Beanstalk, é importante atribuir a ela as permissões adequadas para os modelos de lançamento. Caso contrário, talvez você não tenha as permissões necessárias para atualizar

um ambiente ou iniciar um novo. Para obter mais informações, consulte Permissões necessárias para modelos de execução do .

Uma política do IAM contém declarações que descrevem as permissões a serem concedidas. Ao criar uma declaração de política para o Elastic Beanstalk, você precisa entender como usar as seguintes quatro partes:

- Efeito especifica se é para permitir ou negar as ações na declaração.
- Ação especifica as operações da API que você deseja controlar. Por exemplo, use `elasticbeanstalk:CreateEnvironment` para especificar a operação `CreateEnvironment`. Algumas operações, como criação de ambiente, exigem permissões adicionais para executar essas ações. Para obter mais informações, consulte Recursos e condições para ações do Elastic Beanstalk.

  > **Note**
  >
  > Para usar a operação de API `UpdateTagsForResource`, especifique uma das duas ações virtuais a seguir (ou ambas) em vez do nome de operação da API:
  >
  > `elasticbeanstalk:AddTags`
  >
  > Controla a permissão para chamar `UpdateTagsForResource` e passar uma lista de tags a serem adicionadas ao parâmetro `TagsToAdd`.
  >
  > `elasticbeanstalk:RemoveTags`
  >
  > Controla a permissão para chamar `UpdateTagsForResource` e passar uma lista de chaves de tags a serem removidas do parâmetro `TagsToRemove`.

- Recurso especifica os recursos aos quais você deseja controlar o acesso. Para especificar recursos do Elastic Beanstalk, liste o Nome de recurso da Amazon (ARN) de cada recurso.
- (opcional) Condição especifica restrições sobre a permissão concedida na declaração. Para obter mais informações, consulte Recursos e condições para ações do Elastic Beanstalk.

As seções a seguir demonstram alguns casos em que você pode considerar uma política de usuário personalizada.

Habilitar a criação limitada do ambiente do Elastic Beanstalk

A política no exemplo a seguir permite que um usuário chame a ação `CreateEnvironment` para criar um ambiente cujo nome começa com **`Test`** com o aplicativo especificado e a versão do aplicativo.

```
{
  "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":[
        "*"
      ]
    }
  ]
}
```

A política acima mostra como conceder acesso limitado às operações do Elastic Beanstalk. Para realmente iniciar um ambiente, o usuário também deve ter permissão para criar os AWS recursos que alimentam o ambiente. Por exemplo, a seguinte política concede acesso ao conjunto padrão de recursos para um ambiente de servidor Web:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "ecs:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "cloudformation:*",
        "sqs:*"
        ],
      "Resource": "*"
    }
  ]
}
```

Habilitar o acesso aos logs do Elastic Beanstalk armazenados no Amazon S3

A política no exemplo a seguir permite que um usuário extraia logs do Elastic Beanstalk, prepare-os no Amazon S3 e os recupere.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:DeleteObject",
        "s3:GetObjectAcl",
        "s3:PutObjectAcl"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::elasticbeanstalk-*"
```

```
    }
  ]
}
```

> **Note**
>
> Para restringir essas permissões a somente caminho de logs, use o seguinte formato de recursos.
>
> ```
> "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/resources/environments/logs/*"
> ```

Permitir o gerenciamento de uma aplicação específica do Elastic Beanstalk

A política no exemplo a seguir permite que um usuário gerencie ambientes e outros recursos dentro de uma aplicação específica do Elastic Beanstalk. A política nega ações do Elastic Beanstalk nos recursos de outras aplicações e também nega a criação e exclusão de aplicações do Elastic Beanstalk.

> **Note**
>
> A política não nega acesso a nenhum recurso por meio de outros serviços. Ela se revela uma forma eficiente de distribuir responsabilidades para gerenciar aplicações do Elastic Beanstalk entre diferentes usuários, e não para proteger os recursos subjacentes.

```
{
  "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 de nome de recurso da Amazon para o Elastic Beanstalk

Especifique um recurso para uma política do IAM usando o Nome de recurso da Amazon (ARN) desse recurso. Para o Elastic Beanstalk, o ARN tem o formato a seguir.

```
arn:aws:elasticbeanstalk:region:account-id:resource-type/resource-path
```

Em que:

- *region* é a região na qual o recurso reside (por exemplo, **us-west-2**).
- *account-id*é o ID da AWS conta, sem hífens (por exemplo,) **123456789012**
- *resource-type* identifica o tipo de recurso do Elastic Beanstalk, por exemplo, `environment`. Consulte a tabela a seguir para obter uma lista de todos os tipos de recursos do Elastic Beanstalk.
- *resource-path* é a parte que identifica o recurso específico. Um recurso do Elastic Beanstalk tem um caminho que identifica exclusivamente esse recurso. Veja na tabela abaixo o formato do caminho do recurso para cada tipo de recurso. Por exemplo, um ambiente está sempre associado a um aplicativo. O caminho do recurso para o ambiente **myEnvironment** no aplicativo **myApp** seria assim:

  ```
  myApp/myEnvironment
  ```

O Elastic Beanstalk tem vários tipos de recursos que podem ser especificados em uma política. A tabela a seguir mostra o formato de ARN para cada tipo de recurso e um exemplo.

Tipo de recurso	Formato do ARN
`application`	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:application/` *application-name* Exemplo: **arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App**
`applicationversion`	`arn:aws:elasticbeanstalk:` *region*`:`*account-id* `:applicationversion/` *application-name* `/`*version-label*

Tipo de recurso	Formato do ARN
	Exemplo: **arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version**
configurationtemplate	arn:aws:elasticbeanstalk: *region*:*account-id* :configurationtemplate/ *application-name* /*template-name* Exemplo: **arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template**
environment	arn:aws:elasticbeanstalk: *region*:*account-id* :environment/ *application-name* /*environment-name* Exemplo: **arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/MyEnvironment**
platform	arn:aws:elasticbeanstalk: *region*:*account-id* :platform/ *platform-name* /*platform-version* Exemplo: **arn:aws:elasticbeanstalk:us-east-2:123456789012:platform/MyPlatform/1.0**
solutionstack	arn:aws:elasticbeanstalk: *region*::solutionstack/ *solutionstack-name* Example: **arn:aws:elasticbeanstalk:us-east-2::solutionstack/32bit Amazon Linux running Tomcat 7**

Um ambiente, versão do aplicativo e modelo de configuração estão sempre contidos em um aplicativo específico. Você verá que esses recursos têm um nome de aplicativo no caminho do recurso, de forma que eles sejam identificados exclusivamente pelo nome do recurso e pelo aplicativo que os contêm. Embora as pilhas de soluções sejam usadas por modelos e ambientes de configuração, as pilhas de soluções não são específicas de um aplicativo ou AWS conta e não contêm o aplicativo ou a AWS conta em si. ARNs

Recursos e condições para ações do Elastic Beanstalk

Esta seção descreve os recursos e as condições que podem ser usadas em declarações de política a fim de conceder permissões para que ações específicas sejam executadas em recursos específicos do Elastic Beanstalk.

As condições permitem que você especifique permissões para recursos necessários para a conclusão da ação. Por exemplo, quando você chama a ação `CreateEnvironment`, também deve especificar a versão do aplicativo para implantar, bem como o aplicativo que contém o nome do aplicativo. Ao definir permissões para a ação `CreateEnvironment`, você especifica o aplicativo e a versão do aplicativo nos quais deseja que a ação tenha efeito usando as condições `InApplication` e `FromApplicationVersion`.

Além disso, você pode especificar a configuração do ambiente com um stack de solução (`FromSolutionStack`) ou com um modelo de configuração (`FromConfigurationTemplate`). A declaração de política a seguir permite que a ação `CreateEnvironment` crie um ambiente com o nome **`myenv`** (especificado por `Resource`) no aplicativo **`My App`** (especificado pela condição `InApplication`) usando a versão do aplicativo **`My Version`**(`FromApplicationVersion`) com uma configuração **`32bit Amazon Linux running Tomcat 7`** (`FromSolutionStack`):

```
{
  "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**
>
> A maioria das chaves de condição mencionadas neste tópico é específica do Elastic Beanstalk, e seus nomes contêm o prefixo `elasticbeanstalk:`. Em resumo, nós omitimos este prefixo dos nomes de chave de condição quando nós os mencionamos nas seções a seguir. Por exemplo, mencionamos `InApplication` ao invés de seu nome completo `elasticbeanstalk:InApplication`.
> Por outro lado, mencionamos algumas chaves de condição usadas em todos AWS os serviços e incluímos seu `aws:` prefixo para destacar a exceção.
> Os exemplos de política sempre mostram nomes de chave de condição completos, incluindo o prefixo.

Seções

- Informações de políticas para ações do Elastic Beanstalk
- Chaves de condição para ações do Elastic Beanstalk

Informações de políticas para ações do Elastic Beanstalk

A tabela a seguir lista todas as ações do Elastic Beanstalk, o recurso afetado por cada ação e as informações contextuais adicionais que podem ser fornecidas usando condições.

Informações sobre política para ações do Elastic Beanstalk, incluindo recursos, condições, exemplos e dependências

Recurso	Condições	Exemplo de declaração
Ação: `AbortEnvironmentUpdate`		

`application`
 	
`environment`
	`aws:ResourceTag/` *key-name* (opcional)	A política a seguir permite que um usuário interrompa as operações de atualização de ambiente em ambientes de um aplicativo chamado `My App`.
		
		`{`

Recurso	Condições	Exemplo de declaração

	`aws:TagKeys` (opcional)	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		    {
		      "Action": [
		        "elasticbeanstalk:AbortEnvi
		ronmentUpdate"
		      ],
		      "Effect": "Allow",
		      "Resource": [
		        "arn:aws:elasticbeanstalk:us-
		east-2:123456789012:application/My App"
		      ]
		    }
		  ]
		}</pre>

Ação: `CheckDNSAvailability`		

`"*"`	N/D	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		    {
		      "Action": [
		        "elasticbeanstalk:CheckDNSA
		vailability"
		      ],
		      "Effect": "Allow",
		      "Resource": "*"
		    }
		  ]
		}</pre>

Ação: `ComposeEnvironments`		

Recurso	Condições	Exemplo de declaração
`application`	`aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que um usuário componha ambientes que pertencem a um aplicativo chamado `My App`. `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:ComposeEnvironments" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App" ] } ] }`
Ação: `CreateApplication`		

Recurso	Condições	Exemplo de declaração

`application`	`aws:RequestTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		Este exemplo permite que a ação `CreateApplication` crie aplicativos cujos nomes começam com **`DivA`**:
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:CreateApplication"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/DivA*"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `CreateApplicationVersion`		

Recurso	Condições	Exemplo de declaração

`applicationversion`	`InApplication`
	
	`aws:RequestTag/` *`key-name`* (opcional)
	
	`aws:TagKeys` (opcional)
		Este exemplo permite que a ação `CreateApplicationVersion` crie versões do aplicativo com qualquer nome (**`*`**) no aplicativo **`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"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: `CreateConfigurationTemplate`		

Recurso	Condições	Exemplo de declaração

`configurationtemplate`	`InApplication`
	
	`FromApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`FromEnvironment`
	
	`FromSolutionStack`
	
	`aws:RequestTag/` *`key-name`* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `CreateConfigurationTemplate` crie modelos de configuração cujos nomes comecem com **`My Template`** (`My Template*`) no aplicativo **`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"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: `CreateEnvironment`		

Recurso	Condições	Exemplo de declaração

`environment`	`InApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`FromSolutionStack`
	
	`aws:RequestTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `CreateEnvironment` crie um ambiente cujo nome é **`myenv`** no aplicativo **`My App`** usando o stack de solução **`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"] } } }] }`

Ação: `CreatePlatformVersion`

Recurso	Condições	Exemplo de declaração
`platform`	`aws:RequestTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	Esse exemplo permite que a ação `CreatePlatformVersion` crie versões de plataforma direcionando a região `us-east-2` , cujos nomes começam com **`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_*" ] } ] }`
Ação: `CreateStorageLocation`		
`"*"`	N/D	`{ "Version": "2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:CreateStorageLocation" ], "Effect": "Allow", "Resource": "*" } ] }`

Recurso	Condições	Exemplo de declaração
Ação: DeleteApplication		
application	aws:ResourceTag/ *key-name* (opcional) aws:TagKeys (opcional)	A política a seguir permite que a ação DeleteApplication exclua o aplicativo **My App**: `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DeleteApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Ação: DeleteApplicationVersion		

Recurso	Condições	Exemplo de declaração
`applicationversion`	`InApplication` `aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `DeleteApplicationVersion` exclua uma versão do aplicativo cujo nome é **`My Version`** no aplicativo **`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"] } } }] }`

Ação: [DeleteConfigurationTemplate](#)

Recurso	Condições	Exemplo de declaração

`configurationtemplate`	`InApplication` (opcional)
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DeleteConfigurationTemplate` exclua um modelo de configuração cujo nome é **`My Template`** no aplicativo **`My App`**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:DeleteConfigurationTemplate"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:configurationtemplate/My App/My Template"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DeleteEnvironmentConfiguration`		

Recurso	Condições	Exemplo de declaração

`environment`	`InApplication` (opcional)	A política a seguir permite que a ação `DeleteEnvironmentConfiguration` exclua um modelo de configuração de rascunho do ambiente **`myenv`** no aplicativo **`My App`**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:DeleteEnvironmentConfiguration"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DeletePlatformVersion`		

Recurso	Condições	Exemplo de declaração

`platform`	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DeletePlatformVersion` exclua versões de plataforma direcionando a região `us-east-2` , cujos nomes começam com **`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_*"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DescribeApplications`		

Recurso	Condições	Exemplo de declaração
`application`	`aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `DescribeApplications` descreva o aplicativo My App. `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:DescribeApplications" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Ação: `DescribeApplicationVersions`		

Recurso	Condições	Exemplo de declaração

`applicationversion`	`InApplication` (opcional)
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DescribeApplicationVersions` descreva a versão do aplicativo **`My Version`** no aplicativo **`My App`**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:DescribeApplicationVersions"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:applicationversion/My App/My Version"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DescribeConfigurationOptions`

Recurso	Condições	Exemplo de declaração

`environment`
 	
`configurationtemplate`
 	
`solutionstack`
	`InApplication` (opcional)
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DescribeConfigurationOptions` descreva as opções de configuração do ambiente **myenv** no aplicativo **My App**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{ "Version": "2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeConfigurationOptions", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`

Ação: `DescribeConfigurationSettings`		

Recurso	Condições	Exemplo de declaração

`environment`
 	
`configurationtemplate`
	`InApplication` (opcional)
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DescribeConfigurationSettings` descreva as definições de configuração do ambiente **`myenv`** no aplicativo **`My App`**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": "elasticbeanstalk:DescribeConfigurationSettings",`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DescribeEnvironmentHealth`		

Recurso	Condições	Exemplo de declaração
`environment`	`aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite o uso de `DescribeEnvironmentHealth` para recuperar informações de integridade de um ambiente chamado **`myenv`**. `{ "Version": "2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeEnvironmentHealth", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Ação: `DescribeEnvironmentResources`		

Recurso	Condições	Exemplo de declaração

`environment`	`InApplication` (opcional)
	
	`aws:ResourceTag/` *`key-name`* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a `DescribeEnvironmentResources` ação retorne uma lista de AWS recursos para o ambiente **`myenv`** no aplicativo **`My App`**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": "elasticbeanstalk:DescribeEnvironmentResources",`
		`  "Effect": "Allow",`
		`  "Resource": [`
		`  "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DescribeEnvironments`		

Recurso	Condições	Exemplo de declaração

`environment`	`InApplication` (opcional)
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DescribeEnvironments` descreva os ambientes **`myenv`** e **`myotherenv`** no aplicativo **`My App`**. A especificação do nome do aplicativo como uma condição é opcional.
		
		`{`
		`  "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"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `DescribeEvents`		

Recurso	Condições	Exemplo de declaração

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
	`InApplication`
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DescribeEvents` liste descrições de evento para o ambiente **`myenv`** e a versão do aplicativo **`My Version`** no aplicativo **`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"] } } }] }`

Ação: `DescribeInstancesHealth`		

Recurso	Condições	Exemplo de declaração
`environment`	N/D	A política a seguir permite o uso de `DescribeInstancesHealth` para recuperar informações de integridade de instâncias em um ambiente chamado **`myenv`**. `{ "Version": "2012-10-17", "Statement": [ { "Action": "elasticbeanstalk:DescribeInstancesHealth", "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:environment/My App/myenv" ] } ] }`
Ação: `DescribePlatformVersion`		

Recurso	Condições	Exemplo de declaração

`platform`	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `DescribePlatformVersion` descreva as versões de plataforma direcionando a região `us-east-2` , cujos nomes começam com **`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_*"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `ListAvailableSolutionStacks`		

Recurso	Condições	Exemplo de declaração

`solutionstack`	N/D	A política a seguir permite que a ação `ListAvailableSolutionStacks` retorne somente a pilha de solução **`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"`
		`  }`
		`  ]`
		`}`

Ação: `ListPlatformVersions`

Recurso	Condições	Exemplo de declaração

`platform`	`aws:RequestTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		Esse exemplo permite que a ação `CreatePlatformVersion` crie versões de plataforma direcionando a região `us-east-2` , cujos nomes começam com **`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_*"`
		`  ]`
		`  }`
		`  ]`
		`}`

Ação: `ListTagsForResource`		

Recurso	Condições	Exemplo de declaração

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `ListTagsForResource` liste as tags de recursos existentes somente se elas tiverem uma tag chamada `stage` com o valor `test`:
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:ListTagsForResource"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": "*",`
		`  "Condition": {`
		`  "StringEquals": {`
		`  "aws:ResourceTag/stage": ["test"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: `RebuildEnvironment`		

Recurso	Condições	Exemplo de declaração
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `RebuildEnvironment` recrie o ambiente **`myenv`** no aplicativo **`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"] } } }] }`
Ação: `RequestEnvironmentInfo`		

Recurso	Condições	Exemplo de declaração
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `RequestEnvironmentInfo` compile informações sobre o ambiente **`myenv`** no aplicativo **`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"] } } }] }`
Ação: `RestartAppServer`		

Recurso	Condições	Exemplo de declaração

`environment`	`InApplication`	A política a seguir permite que a ação `RestartAppServer` reinicie o servidor do contêiner do aplicativo para o ambiente **`myenv`** no aplicativo **`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"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: [RetrieveEnvironmentInfo](#)		

Recurso	Condições	Exemplo de declaração
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `RetrieveEnvironmentInfo` recupere as informações compiladas sobre o ambiente **`myenv`** no aplicativo **`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"] } } }] }`
Ação: `SwapEnvironmentCNAMEs`		

Recurso	Condições	Exemplo de declaração
`environment`	`InApplication` (opcional) `FromEnvironment` (opcional)	A política a seguir permite que a `SwapEnvironmentCNAMEs` ação troque o CNAMEs pelos ambientes **`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" ] } ] }`
Ação: `TerminateEnvironment`		

Recurso	Condições	Exemplo de declaração
`environment`	`InApplication` `aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `TerminateEnvironment` termine o ambiente **`myenv`** no aplicativo **`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"] } } }] }`
Ação: UpdateApplication		

Recurso	Condições	Exemplo de declaração
`application`	`aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `UpdateApplication` atualize as propriedades do aplicativo **My App**. `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:UpdateApplication" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`

Ação: UpdateApplicationResourceLifecycle

Recurso	Condições	Exemplo de declaração
`application`	`aws:ResourceTag/` *`key-name`* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `UpdateApplicationResourceLifecycle` atualize as configurações de ciclo de vida do aplicativo **`My App`**. `{ "Version": "2012-10-17", "Statement": [ { "Action": [ "elasticbeanstalk:UpdateApplicationResourceLifecycle" ], "Effect": "Allow", "Resource": [ "arn:aws:elasticbeanstalk:us-east-2:123456789012:application/My App" ] } ] }`
Ação: `UpdateApplicationVersion`		

Recurso	Condições	Exemplo de declaração

`applicationversion`	`InApplication`
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `UpdateApplicationVersion` atualize as propriedades da versão do aplicativo **`My Version`** no aplicativo **`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"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: [UpdateConfigurationTemplate](#)		

Recurso	Condições	Exemplo de declaração
`configurationtemplate`	`InApplication` `aws:ResourceTag/` *key-name* (opcional) `aws:TagKeys` (opcional)	A política a seguir permite que a ação `UpdateConfigurationTemplate` atualize as propriedades ou as opções do modelo de configuração **`My Template`** no aplicativo **`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"] } } }] }`
Ação: `UpdateEnvironment`		

Recurso	Condições	Exemplo de declaração

`environment`	`InApplication`
	
	`FromApplicationVersion`
	
	`FromConfigurationTemplate`
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `UpdateEnvironment` atualize o ambiente **`myenv`** no aplicativo **`My App`** implantando a versão do aplicativo **`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"] } } }] }`

Ação: `UpdateTagsForResource` – `AddTags`		

Recurso	Condições	Exemplo de declaração

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:RequestTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A ação `AddTags` é uma das duas ações virtuais associadas à API `UpdateTagsForResource`.
		
		A política a seguir permite que a ação `AddTags` modifique as tags de recursos existentes somente se elas tiverem uma tag chamada `stage` com o valor `test`:
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:AddTags"`
		`  ],`
		`  "Effect": "Allow",`
		`  "Resource": "*",`
		`  "Condition": {`
		`  "StringEquals": {`
		`  "aws:ResourceTag/stage": ["test"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: `UpdateTagsForResource` – `RemoveTags`		

Recurso	Condições	Exemplo de declaração

`application`
 	
`applicationversion`
 	
`configurationtemplate`
 	
`environment`
 	
`platform`
	`aws:ResourceTag/` *`key-name`* (opcional)
	
	`aws:TagKeys` (opcional)
		A ação `RemoveTags` é uma das duas ações virtuais associadas à API `UpdateTagsForResource`.
		
		A política a seguir nega a ação `RemoveTags` para solicitar a remoção de uma tag chamada `stage` dos recursos existentes:
		
		`{`
		`  "Version": "2012-10-17",`
		`  "Statement": [`
		`  {`
		`  "Action": [`
		`  "elasticbeanstalk:RemoveTags"`
		`  ],`
		`  "Effect": "Deny",`
		`  "Resource": "*",`
		`  "Condition": {`
		`  "ForAnyValue:StringEquals": {`
		`  "aws:TagKeys": ["stage"]`
		`  }`
		`  }`
		`  }`
		`  ]`
		`}`

Ação: `ValidateConfigurationSettings`		

Recurso	Condições	Exemplo de declaração

`template`
 	
`environment`
	`InApplication`
	
	`aws:ResourceTag/` *key-name* (opcional)
	
	`aws:TagKeys` (opcional)
		A política a seguir permite que a ação `ValidateConfigurationSettings` valide as configurações no ambiente **`myenv`** no aplicativo **`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"] } } }] }`

Chaves de condição para ações do Elastic Beanstalk

As chaves permitem que você especifique condições que expressam dependências, restringem permissões ou especificam restrições aos parâmetros de entrada de uma ação. O Elastic Beanstalk é compatível com as chaves a seguir.

`InApplication`

Especifica o aplicativo que contém o recurso afetado pela ação.

O exemplo a seguir permite que a ação `UpdateApplicationVersion` atualize as propriedades da versão do aplicativo **`My Version`**. A condição `InApplication` especifica **`My App`** como o contêiner de **`My Version`**.

```
{
  "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`

Especifica uma versão do aplicativo como uma dependência ou uma restrição a um parâmetro de entrada.

O exemplo a seguir permite que a ação `UpdateEnvironment` atualize o ambiente **`myenv`** no aplicativo **`My App`**. A condição `FromApplicationVersion` restringe o parâmetro `VersionLabel` para permitir que somente a versão do aplicativo **`My Version`** atualize o ambiente.

```
{
  "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`

Especifica um modelo de configuração como uma dependência ou uma restrição a um parâmetro de entrada.

O exemplo a seguir permite que a ação `UpdateEnvironment` atualize o ambiente **`myenv`** no aplicativo **`My App`**. A condição `FromConfigurationTemplate` restringe o parâmetro `TemplateName` para permitir que somente o modelo de configuração **`My Template`** atualize o ambiente.

```
{
  "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`

Especifica um ambiente como uma dependência ou uma restrição a um parâmetro de entrada.

O exemplo a seguir permite que a `SwapEnvironmentCNAMEs` ação troque o CNAMEs in **`My App`** por todos os ambientes cujos nomes começam com **`mysrcenv`** e**`mydestenv`**, mas não pelos ambientes cujos nomes começam com **`mysrcenvPROD*`** e. **`mydestenvPROD*`**

```
{
  "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": {
        "StringNotLike": {
          "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`

Especifica um stack de solução como uma dependência ou uma restrição a um parâmetro de entrada.

A política a seguir permite que a ação `CreateConfigurationTemplate` crie modelos de configuração cujos nomes comecem com **`My Template`** (`My Template*`) no aplicativo **`My App`**. A condição `FromSolutionStack` restringe que o parâmetro `solutionstack` permita apenas a pilha de solução **`32bit Amazon Linux running Tomcat 7`** como o valor de entrada do parâmetro.

```
{
  "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`

Especifique as condições com base na tag. Para obter mais detalhes, consulte Usar tags para controlar o acesso aos recursos do Elastic Beanstalk.

Usar tags para controlar o acesso aos recursos do Elastic Beanstalk

Este tópico explica como o controle de acesso baseado em tags pode ajudar a criar e gerenciar políticas do IAM.

Podemos usar condições em declarações de política de usuário do IAM para configurar permissões de acesso do Elastic Beanstalk aos recursos. Para saber mais sobre condições de declarações de política, consulte Recursos e condições para ações do Elastic Beanstalk. O uso de tags em condições é uma forma de controlar o acesso a recursos e solicitações. Para obter informações sobre como marcar recursos do Elastic Beanstalk, consulte Marcar recursos da aplicação do Elastic Beanstalk.

Ao criar políticas do IAM, você pode definir permissões granulares concedendo acesso a recursos específicos. À medida que o número de recursos que você gerencia aumenta, essa tarefa se torna mais difícil. Atribuir etiquetas a recursos e usá-las em condições de declaração de política pode facilitar essa tarefa. Você concede acesso em massa a qualquer recurso utilizando determinada etiqueta. Depois, você a aplica repetidamente a recursos relevantes durante a criação ou posteriormente.

As tags podem ser anexadas ao recurso ou passadas na solicitação para serviços compatíveis com tags. No Elastic Beanstalk, os recursos podem ter tags, e algumas ações podem incluir tags. Ao criar uma política do IAM, você pode usar chaves de condição de etiqueta para controlar as seguintes condições:

- Quais usuários podem executar ações em um ambiente, com base nas tags que ele já tem.
- Quais tags podem ser transmitidas na solicitação de uma ação.
- Se chaves de tags específicas podem ser usadas em uma solicitação.

Para obter a sintaxe e a semântica completas das chaves de condição de tag, consulte Controlar o acesso usando tags no Guia do usuário do IAM.

Exemplos de condições de etiquetas em políticas

Os exemplos a seguir demonstram como especificar condições de tag em políticas para usuários do Elastic Beanstalk.

Example 1: Limitar ações com base em tags na solicitação

A política de usuário gerenciado do AdministratorAccessElastic AWSElastic Beanstalk - Beanstalk dá aos usuários permissão ilimitada para realizar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

A política a seguir limita esse poder e nega a usuários não autorizados permissão para criar ambientes de produção do Elastic Beanstalk. Para fazer isso, ela negará a ação `CreateEnvironment` se a solicitação especificar uma tag chamada `stage` com um dos valores `gamma` ou `prod`. Além disso, para impedir que esses usuários não autorizados interfiram na fase de ambiente de produção, a política não permite que ações de modificação de etiquetas incluam esses mesmos valores de etiqueta ou removam completamente a etiqueta `stage`. O administrador de um cliente deve anexar essa política do IAM a usuários não autorizados do IAM, além da política de usuário gerenciada.

```
{
  "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: Limitar ações com base em tags de recursos

A política de usuário gerenciado do AdministratorAccessElastic AWSElastic Beanstalk - Beanstalk dá aos usuários permissão ilimitada para realizar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

A política a seguir limita esse poder e nega a usuários não autorizados permissão para realizar ações em ambientes de produção do Elastic Beanstalk. Para fazer isso, ela negará ações específicas se o ambiente tiver uma tag chamada `stage` com um dos valores `gamma` ou `prod`. O administrador de um cliente deve anexar essa política do IAM a usuários não autorizados do IAM, além da política de usuário gerenciada.

```
{
  "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: Permitir ações com base em tags na solicitação

A política a seguir concede aos usuários permissão para criar aplicações de desenvolvimento do Elastic Beanstalk.

Para fazer isso, ela permitirá as ações `AddTags` e `CreateApplication` se a solicitação especificar uma tag chamada `stage` com o valor `development`. A condição `aws:TagKeys` garante que o usuário não possa adicionar outras chaves de tag. Especificamente, ele garante a diferenciação entre maiúsculas e minúsculas da chave da tag `stage`. Observe que essa política é útil para usuários do IAM que não têm a política de usuário gerenciada do Elastic Beanstalk AdministratorAccess — AWSElasticBeanstalk anexada. A política gerenciada oferece aos usuários permissão ilimitada para executar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

```
{
  "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: Permitir ações com base em tags de recursos

A política a seguir concede aos usuários permissão para executar ações e obter informações sobre as aplicações de desenvolvimento do Elastic Beanstalk.

Para fazer isso, ela permitirá ações específicas se o aplicativo tiver uma tag chamada `stage` com o valor `development`. A condição `aws:TagKeys` garante que o usuário não possa adicionar outras chaves de tag. Especificamente, ele garante a diferenciação entre maiúsculas e minúsculas da chave da tag `stage`. Observe que essa política é útil para usuários do IAM que não têm a política de usuário gerenciada do Elastic Beanstalk AdministratorAccess — AWSElasticBeanstalk anexada. A política gerenciada oferece aos usuários permissão ilimitada para executar qualquer ação do Elastic Beanstalk em qualquer recurso gerenciado pelo Elastic Beanstalk.

```
{
  "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"]
        }
      }
    }
  ]
}
```

Exemplo de políticas baseadas em políticas gerenciadas

Esta seção demonstra como controlar o acesso do usuário AWS Elastic Beanstalk e inclui exemplos de políticas que fornecem o acesso necessário para cenários comuns. Essas políticas são derivadas das políticas gerenciadas do Elastic Beanstalk. Para obter informações sobre como anexar políticas gerenciadas a usuários e grupos, consulte Gerenciar políticas de usuário do Elastic Beanstalk.

Neste cenário, Exemplo Corp. é uma empresa de software com três equipes responsáveis pelo site da empresa: os administradores que gerenciam a infraestrutura, os desenvolvedores que criam o software para o site e uma equipe de controle de qualidade que testa o site. Para ajudar a gerenciar

permissões para os recursos do Elastic Beanstalk, a Exemplo Corp. cria três grupos aos quais os membros de cada equipe pertencem: administradores, desenvolvedores e testadores. A Exemplo Corp. deseja que o grupo de administradores tenha acesso total a todas as aplicações, ambientes e seus recursos subjacentes, para que possam criar, solucionar problemas e excluir todos os ativos do Elastic Beanstalk. Os desenvolvedores precisam de permissões para visualizar todos os ativos do Elastic Beanstalk e para criar e implantar versões da aplicação. Os desenvolvedores não devem ter permissão para criar novos aplicativos ou ambientes ou para encerrar ambientes em execução. Os testadores precisam visualizar todos os recursos do Elastic Beanstalk para monitorar e testar aplicações. Os testadores não devem ser capazes de fazer alterações em nenhum recurso do Elastic Beanstalk.

O seguinte exemplo de políticas fornece as permissões necessárias para cada grupo.

Exemplo 1: Grupo de administradores — Todo o Elastic Beanstalk e serviços relacionados APIs

A política a seguir oferece aos usuários permissões para todas as ações necessárias para usar o Elastic Beanstalk. Ela também permite que o Elastic Beanstalk provisione e gerencie recursos em seu nome nos serviços a seguir. O Elastic Beanstalk depende desses serviços adicionais para provisionar recursos subjacentes ao criar um ambiente.

- Amazon Elastic Compute Cloud
- Elastic Load Balancing
- Auto Scaling
- Amazon CloudWatch
- Amazon Simple Storage Service
- Amazon Simple Notification Service
- Amazon Relational Database Service
- AWS CloudFormation

Observe que esta política é um exemplo. Ela fornece um amplo conjunto de permissões aos serviços da AWS que o Elastic Beanstalk usa para gerenciar aplicações e ambientes. Por exemplo, `ec2:*` permite que um usuário AWS Identity and Access Management (IAM) realize qualquer ação em qualquer EC2 recurso da Amazon na AWS conta. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticbeanstalk:*",
        "ec2:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "cloudwatch:*",
        "s3:*",
        "sns:*",
        "rds:*",
        "cloudformation:*"
      ],
      "Resource" : "*"
    }
  ]
}
```

Exemplo 2: Grupo de desenvolvedores: todas as operações, exceto altamente privilegiadas

A política a seguir nega permissão para criar aplicações e ambientes e permite todas as demais ações do Elastic Beanstalk.

Observe que esta política é um exemplo. Ela fornece um amplo conjunto de permissões para produtos da AWS que o Elastic Beanstalk usa para gerenciar aplicações e ambientes. Por exemplo, `ec2:*` permite que um usuário do IAM realize qualquer ação em qualquer EC2 recurso da Amazon na AWS conta. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```
{
  "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" : "*"
    }
  ]
}
```

Exemplo 3: Testadores: somente visualização

A seguinte política permite acesso somente leitura a todos os aplicativos, versões de aplicativos, eventos e ambientes. Não permite realizar nenhuma ação.

```
{
  "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" : "*"
    }
  ]
}
```

Políticas de exemplo baseadas em permissões de recursos

Esta seção aborda um caso de uso para controlar as permissões de usuário para ações do Elastic Beanstalk que acessam recursos específicos do Elastic Beanstalk. Apresentaremos as políticas de exemplo compatíveis com o caso de uso. Para obter mais informações sobre os recursos do Elastic Beanstalk, consulte Criar uma política de usuário personalizada. Para obter informações sobre como anexar políticas a usuários e grupos, acesse Gerenciamento de políticas do IAM em Uso do AWS Identity and Access Management.

No nosso caso de uso, Exemplo Corp. é uma pequena empresa de consultoria que desenvolve aplicativos para dois clientes diferentes. John é o gerente de desenvolvimento que supervisiona o desenvolvimento das duas aplicações do Elastic Beanstalk, app1 e app2. John faz o desenvolvimento e alguns testes nos dois aplicativos, e apenas ele pode atualizar o ambiente de produção dos dois aplicativos. Estas são as permissões de que ele precisa para o app1 e o app2:

- Visualização do aplicativo, versões de aplicativos, ambientes e modelos de configuração
- Criação das versões do aplicativo e implantação no ambiente de preparação
- Atualização do ambiente de produção
- Criação e encerramento de ambientes

Jill é uma testadora que precisa de acesso para visualizar os seguintes recursos a fim de monitorar e testar os dois aplicativos: aplicativos, versões de aplicativos, ambientes e modelos de configuração. No entanto, ela não deve ser capaz de fazer alterações em nenhum recurso do Elastic Beanstalk.

Jack é o desenvolvedor do app1 que precisa de acesso para visualizar todos os recursos do app1 e também precisa criar versões de aplicativos para o app1 e implantá-las no ambiente de preparação.

Judy é administradora da AWS conta da Example Corp. Ela criou usuários do IAM para John, Jill e Jack e anexa as seguintes políticas a esses usuários para conceder as permissões apropriadas aos aplicativos app1 e app2.

Exemplo 1: John: gerente de desenvolvimento para app1, app2

Dividimos a política de John em três políticas separadas para facilitar a leitura e o gerenciamento. Juntas, elas oferecem a John as permissões de que precisa para executar as ações de desenvolvimento, teste e implantação nos dois aplicativos.

A primeira política especifica ações para Auto Scaling, Amazon S3, Amazon, Amazon SNS EC2, Elastic Load Balancing, CloudWatch Amazon RDS e. AWS CloudFormation O Elastic Beanstalk depende desses serviços adicionais para provisionar recursos subjacentes ao criar um ambiente.

Observe que esta política é um exemplo. Ele fornece um amplo conjunto de permissões aos AWS produtos que o Elastic Beanstalk usa para gerenciar aplicativos e ambientes. Por exemplo, `ec2:*` permite que um usuário do IAM realize qualquer ação em qualquer EC2 recurso da Amazon na AWS conta. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```
{
   "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:PassRole",
            "iam:ListRolePolicies",
            "iam:ListAttachedRolePolicies",
            "iam:ListInstanceProfiles",
            "iam:ListRoles",
            "iam:ListServerCertificates",
            "acm:DescribeCertificate",
            "acm:ListCertificates",
            "codebuild:CreateProject",
            "codebuild:DeleteProject",
            "codebuild:BatchGetBuilds",
            "codebuild:StartBuild"
         ],
         "Resource":"*"
      }
   ]
}
```

A segunda política especifica as ações do Elastic Beanstalk que John tem permissão para executar nos recursos do app1 e do app2. A declaração `AllCallsInApplications` permite que todas as ações do Elastic Beanstalk (`"elasticbeanstalk:*"`) sejam realizadas em todos os recursos no app1 e no app2 (por exemplo, `elasticbeanstalk:CreateEnvironment`). A declaração `AllCallsOnApplications` permite todas as ações do Elastic Beanstalk (`"elasticbeanstalk:*"`) nos recursos do app1 e do app2 (por exemplo, `elasticbeanstalk:DescribeApplications`, `elasticbeanstalk:UpdateApplication`, etc.). A declaração `AllCallsOnSolutionStacks` permite todas as ações do Elastic Beanstalk (`"elasticbeanstalk:*"`) para os recursos de pilha de solução (por exemplo, `elasticbeanstalk:ListAvailableSolutionStacks`).

```
{
   "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/*"
         ]
      }
   ]
}
```

A terceira política especifica as ações do Elastic Beanstalk para as quais a segunda política precisa de permissões para concluir essas ações do Elastic Beanstalk. A instrução `AllNonResourceCalls` permite a ação `elasticbeanstalk:CheckDNSAvailability`, que é necessária para chamar `elasticbeanstalk:CreateEnvironment` e outras ações. Ela

também permite a ação `elasticbeanstalk:CreateStorageLocation`, que é necessária para `elasticbeanstalk:CreateApplication`, `elasticbeanstalk:CreateEnvironment` e outras ações.

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability",
            "elasticbeanstalk:CreateStorageLocation"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Exemplo 2: Jill: testadora do app1, app2

Dividimos a política de Jill em três separadas para facilitar a leitura e o gerenciamento. Juntas, elas oferecem a Jill as permissões de que precisa para executar as ações de teste e monitoramento nos dois aplicativos.

A primeira política especifica `Describe*` `List*`, e `Get*` ações no Auto Scaling, Amazon S3, Amazon, Amazon EC2 SNS, Elastic Load Balancing, CloudWatch Amazon RDS AWS CloudFormation e (para tipos de contêineres não legados) para que as ações do Elastic Beanstalk possam recuperar as informações relevantes sobre os recursos subjacentes dos aplicativos app1 e app2.

```
{
   "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":"*"
      }
   ]
}
```

A segunda política especifica as ações do Elastic Beanstalk que Jill tem permissão para executar nos recursos do app1 e do app2. A instrução `AllReadCallsInApplications` permite que ela chame as ações `Describe*` e as ações de informações do ambiente. A instrução `AllReadCallsOnApplications` permite que ela chame as ações `DescribeApplications` and `DescribeEvents` nos recursos de aplicativos do app1 e do app2. A instrução `AllReadCallsOnSolutionStacks` permite visualizar as ações que envolvem recursos de pilha de soluções (`ListAvailableSolutionStacks`, `DescribeConfigurationOptions` e `ValidateConfigurationSettings`).

```
{
   "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/*"
         ]
      }
   ]
}
```

A terceira política especifica as ações do Elastic Beanstalk para as quais a segunda política precisa de permissões para concluir essas ações do Elastic Beanstalk. A instrução `AllNonResourceCalls` permite a ação `elasticbeanstalk:CheckDNSAvailability`, que é necessária para chamar ações de visualização.

```
{
```

```
   "Version": "2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Exemplo 3: Jack: desenvolvedor do app1

Dividimos a política de Jack em três separadas para facilitar a leitura e o gerenciamento. Juntas, elas oferecem a Jack as permissões de que precisa para executar as ações de teste, monitoramento e implantação no recurso do app1.

A primeira política especifica as ações no Auto Scaling, Amazon S3, Amazon, Amazon EC2 SNS, Elastic Load Balancing, CloudWatch Amazon RDS AWS CloudFormation e (para tipos de contêineres não legados) para que as ações do Elastic Beanstalk possam visualizar e trabalhar com os recursos subjacentes do app1. Para obter uma lista de tipos de contêiner não legados compatíveis, consulte the section called “Por que algumas versões de plataforma são marcadas como legadas?”.

Observe que esta política é um exemplo. Ele fornece um amplo conjunto de permissões aos AWS produtos que o Elastic Beanstalk usa para gerenciar aplicativos e ambientes. Por exemplo, `ec2:*` permite que um usuário do IAM realize qualquer ação em qualquer EC2 recurso da Amazon na AWS conta. Essas permissões não estão limitadas aos recursos utilizados com o Elastic Beanstalk. Como uma das melhores práticas, você deve conceder a indivíduos apenas as permissões de que precisam para executar suas tarefas.

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
```

```
            "ec2:*",
            "elasticloadbalancing:*",
            "autoscaling:*",
            "cloudwatch:*",
            "s3:*",
            "sns:*",
            "rds:*",
            "cloudformation:*"
         ],
         "Resource":"*"
      }
   ]
}
```

A segunda política especifica as ações do Elastic Beanstalk que Jack tem permissão para executar no recurso do app1.

```
{
   "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"
               ]
            },
            "StringLike":{
               "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/*"
```

```
         ]
      }
   ]
}
```

A terceira política especifica as ações do Elastic Beanstalk para as quais a segunda política precisa de permissões para concluir essas ações do Elastic Beanstalk. A instrução `AllNonResourceCalls` permite a ação `elasticbeanstalk:CheckDNSAvailability`, que é necessária para chamar `elasticbeanstalk:CreateEnvironment` e outras ações. Ela também permite a ação `elasticbeanstalk:CreateStorageLocation`, que é necessária para `elasticbeanstalk:CreateEnvironment` e outras ações.

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Sid":"AllNonResourceCalls",
         "Action":[
            "elasticbeanstalk:CheckDNSAvailability",
            "elasticbeanstalk:CreateStorageLocation"
         ],
         "Effect":"Allow",
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Impedir o acesso ao bucket do Amazon S3 entre ambientes

Este tópico explica como as políticas gerenciadas podem permitir o acesso ao bucket do S3 entre ambientes e como você pode criar políticas personalizadas para gerenciar esse tipo de acesso.

O Elastic Beanstalk fornece políticas gerenciadas para AWS lidar com os recursos exigidos pelos ambientes do Elastic Beanstalk em sua conta. AWS As permissões fornecidas por padrão para um aplicativo em sua AWS conta têm acesso aos recursos do S3 que pertencem a outros aplicativos na mesma AWS conta.

Se sua AWS conta executa vários aplicativos Beanstalk, você pode definir a segurança de suas políticas criando sua própria política personalizada para anexar à sua própria função de serviço

ou perfil de instância para cada ambiente. É possível limitar as permissões do S3 na política personalizada para um ambiente específico.

> **Note**
>
> Esteja ciente de que você é responsável por manter sua política personalizada. Se uma política gerenciada do Elastic Beanstalk na qual sua política personalizada se baseia for alterada, você precisará modificar sua política personalizada com as respectivas alterações na política de base. Para obter um histórico de alterações das políticas gerenciadas do Elastic Beanstalk, consulte Atualizações do Elastic AWS Beanstalk nas políticas gerenciadas.

Exemplo de permissões com escopo reduzido

O exemplo a seguir é baseado na política AWSElasticBeanstalkWebTiergerenciada.

A política padrão inclui as seguintes linhas para permissões de buckets do S3. Essa política padrão não limita as ações do bucket do S3 a ambientes ou aplicações específicos.

```
{
   "Sid" : "BucketAccess",
   "Action" : [
      "s3:Get*",
      "s3:List*",
      "s3:PutObject"
     ],
   "Effect" : "Allow",
   "Resource" : [
      "arn:aws:s3:::elasticbeanstalk-*",
      "arn:aws:s3:::elasticbeanstalk-*/*"
     ]
}
```

É possível restringir o acesso qualificando recursos específicos para um perfil de serviço especificado como `Principal`. O exemplo a seguir fornece as permissões personalizadas do perfil de serviço `aws-elasticbeanstalk-ec2-role-my-example-env` para os buckets do S3 no ambiente com ID `my-example-env-ID`.

Example Conceder permissões apenas para os buckets do S3 de um ambiente específico

```
{
```

```
   "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**
>
> O ARN do recurso deve incluir o ID do ambiente do Elastic Beanstalk (não o nome do ambiente). É possível obter o ID do ambiente no console do Elastic Beanstalk, na página Visão geral do ambiente. Você também pode usar o comando AWS CLI describe-environments para obter essas informações.

Para obter mais informações que o ajudem a atualizar as permissões de bucket do S3 para seus ambientes Elastic Beanstalk, consulte os seguintes recursos:

- Usar o Elastic Beanstalk com o Amazon S3 neste guia
- Tipos de recursos definidos pelo Amazon S3 no guia de referência de autorização de serviço
- Formato ARN no Guia do usuário do IAM

Usar o Elastic Beanstalk com o Amazon RDS

Esta seção explica como você pode usar o Elastic Beanstalk com o Amazon Relational Database Service (Amazon RDS) para configurar, operar e dimensionar um banco de dados relacional. Explicamos alguns conceitos sobre configuração e fazemos recomendações. Em seguida, vamos orientá-lo no processo de criação e conexão com um Amazon RDS.

Há duas opções para começar:

- Criar um novo banco de dados do Amazon RDS.
- Comece com um banco de dados que foi criado pelo Elastic Beanstalk anteriormente e que depois foi desacoplado de um ambiente do Beanstalk. Para obter mais informações, consulte the section called “Banco de dados”.

Selecione a abordagem

Você pode usar qualquer uma das duas abordagens para executar uma instância de banco de dados no Amazon RDS e configurar a sua aplicação para conectar-se a ela no lançamento. É possível conectar vários ambientes a um banco de dados.

> **Note**
>
> Se você nunca usou uma instância de banco de dados com sua aplicação, recomendamos que primeiro adicione um banco de dados a um ambiente de teste com o console do Elastic Beanstalk. Fazendo isso, você pode verificar se a aplicação pode ler as propriedades do ambiente, criar uma sequência de conexão e conectar-se a uma instância de banco de dados, sem o trabalho de configuração adicional necessário para um banco de dados autônomo. Para obter mais informações, consulte Adição de um banco de dados ao seu ambiente do Elastic Beanstalk.

Configurar um grupo de segurança

Para permitir que as EC2 instâncias da Amazon em seu ambiente se conectem a um banco de dados externo, configure um grupo de segurança adicional para o grupo Auto Scaling associado ao seu ambiente. Você pode anexar o mesmo grupo de segurança anexado à sua instância de banco de dados. Você também pode usar um grupo de segurança separado. Se você anexar um grupo de segurança diferente, deve configurar o grupo de segurança anexado ao banco de dados para permitir o acesso de entrada a partir desse grupo de segurança.

> **Note**
>
> É possível conectar seu ambiente a um banco de dados adicionando uma regra ao grupo de segurança anexado ao banco de dados. Essa regra deve permitir o acesso de entrada a partir do grupo de segurança gerado automaticamente que o Elastic Beanstalk anexa ao

grupo do Auto Scaling do seu ambiente. Mas saiba que, ao criar essa regra, você também cria uma dependência entre os dois grupos de segurança. Subsequentemente, quando você tentar terminar o ambiente, o Elastic Beanstalk não poderá excluir o grupo de segurança do ambiente, porque o grupo de segurança do banco de dados é dependente dele.

Configurar a conexão com o banco de dados

Após iniciar sua instância de banco de dados e configurar grupos de segurança, você pode passar as informações de conexão (endpoint, senha, etc.) para sua aplicação usando as propriedades do ambiente. Este é o mesmo mecanismo que o Elastic Beanstalk usa em segundo plano quando você executa uma instância de banco de dados no seu ambiente.

Para ter uma camada de segurança adicional, é possível armazenar suas informações de conexão no Amazon S3 e configurar o Elastic Beanstalk para recuperá-las durante a implantação. Com os arquivos de configuração (`.ebextensions`), é possível configurar as instâncias no ambiente para recuperar arquivos do Amazon S3 com segurança ao implantar a aplicação.

Tópicos

- Iniciar e conectar-se a uma instância externa do Amazon RDS em uma VPC padrão
- Armazenando as credenciais do Amazon RDS em AWS Secrets Manager
- Limpar uma instância externa do Amazon RDS

Iniciar e conectar-se a uma instância externa do Amazon RDS em uma VPC padrão

Os procedimentos a seguir descrevem o processo de conexão a uma instância externa do Amazon RDS em uma VPC padrão. O processo é o mesmo se você estiver usando uma VPC personalizada. Os únicos requisitos adicionais são que o seu ambiente e instância de banco de dados estejam na mesma sub-rede ou em sub-redes que possam se comunicar umas com as outras. Para obter mais informações sobre como configurar uma VPC personalizada para usar com o Elastic Beanstalk, consulte Usar o Elastic Beanstalk com Amazon VPC.

> **Note**
>
> - Uma alternativa para iniciar uma nova instância de banco de dados é começar com um banco de dados que foi criado anteriormente pelo Elastic Beanstalk e posteriormente

desacoplado de um ambiente do Beanstalk. Para obter mais informações, consulte the section called “Banco de dados”. Com essa opção, você não precisa realizar o procedimento para iniciar um novo banco de dados. Mas, você precisa realizar os procedimentos subsequentes descritos neste tópico.

- Se estiver começando com um banco de dados que foi criado pelo Elastic Beanstalk e posteriormente desacoplado de um ambiente do Beanstalk, você pode pular o primeiro grupo de etapas e continuar com as etapas em Para modificar as regras de entrada no grupo de segurança da instância do RDS.
- Se planeja usar o banco de dados que você desacopla para um ambiente de produção, verifique se o tipo de armazenamento usado pelo banco de dados é adequado para sua workload. Para obter mais informações, consulte DB Instance Storage (Armazenamento de instâncias de banco de dados) e Modifying a DB instance (Modificar uma instância de banco de dados) no Amazon RDS User Guide (Guia do usuário do Amazon RDS).

Para executar uma instância de banco de dados do RDS em uma VPC padrão

1. Abra o console do RDS.
2. No painel de navegação, escolha Databases (Bancos de dados).
3. Selecione Criar banco de dados.
4. Escolha Standard Create (Criação padrão).

> ⚠ Important
>
> Não escolha Easy Create (Criação fácil). Se o escolher, você não poderá definir as configurações necessárias para iniciar esse banco de dados do RDS.

5. Em Additional configuration (Configuração adicional), para Initial database name (Nome do banco de dados inicial), digite **ebdb**.
6. Revise as configurações padrão e ajuste essas configurações de acordo com seus requisitos específicos. Preste atenção às seguintes opções:

   - DB instance class (Classe de instância de banco de dados): escolha um tamanho de instância que tem uma quantidade apropriada de memória e energia de CPU para sua workload.
   - Implantação Multi-AZ — Para alta disponibilidade, defina isso como Criar um nó Replica/ Reader Aurora em uma AZ diferente.

- Master username (Nome do usuário principal) e Master password (Senha principal):o nome de usuário do banco de dados e senha. Anote essas configurações porque você vai usá-las mais tarde.

7. Verifique as configurações padrão para as opções restantes e selecione Create database (Criar banco de dados).

Em seguida, modifique o grupo de segurança anexado à sua instância de banco de dados para permitir tráfego de entrada na porta apropriada. Este é o mesmo grupo de segurança que você vai anexar ao ambiente do Elastic Beanstalk mais tarde. Como resultado, a regra que você adiciona concederá permissão de acesso de entrada a outros recursos no mesmo grupo de segurança.

Para modificar as regras de entrada no grupo de segurança anexado à instância do RDS

1. Abra o console do Amazon RDS.
2. Escolha Databases (Bancos de dados).
3. Escolha o nome da sua instância de banco de dados para visualizar detalhes.
4. Na seção Connectivity (Conectividade), anote as Subnets (Sub-redes), os Security groups (Grupos de segurança) e o Endpoint que são exibidos nesta página. Faça isso para poder usar essas informações mais tarde.
5. Em Security (Segurança), você pode ver o grupo de segurança associado à instância de banco de dados. Abra o link para ver o grupo de segurança no EC2 console da Amazon.
6. Nos detalhes do grupo de segurança, escolha a guia Inbound (Entrada).
7. Escolha Editar.
8. Escolha Add Rule.
9. Para Tipo, escolha o mecanismo de banco de dados usado pelo aplicativo.
10. Em Source (Origem), digite **`sg-`** para exibir uma lista dos grupos de segurança disponíveis. Escolha o grupo de segurança associado ao grupo do Auto Scaling usado com seu ambiente do Elastic Beanstalk. Isso é para que as EC2 instâncias da Amazon no ambiente possam ter acesso ao banco de dados.



11. Escolha Salvar.

Em seguida, adicione o grupo de segurança da instância de banco de dados ao seu ambiente em execução. Nesse procedimento, o Elastic Beanstalk reprograma todas as instâncias em seu ambiente com o grupo de segurança adicional anexado.

Para adicionar um security group ao seu ambiente

- Execute um destes procedimentos:
  - Como adicionar um grupo de segurança usando o console do Elastic Beanstalk
    a. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
    b. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
    c. No painel de navegação, escolha Configuration (Configuração).
    d. Na categoria de configuração Instances (Instâncias), escolha Edit (Editar).
    e. Em grupos EC2 de segurança, escolha o grupo de segurança a ser anexado às instâncias, além do grupo de segurança de instâncias que o Elastic Beanstalk cria.
    f. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.
    g. Leia o aviso e escolha Confirm.
  - Para adicionar um grupo de segurança usando um arquivo de configuração, use o exemplo de arquivo `securitygroup-addexisting.config`.

Em seguida, passe as informações de conexão para o seu ambiente usando as propriedades de ambiente. Quando você adiciona uma instância de banco de dados ao ambiente com o console do Elastic Beanstalk, o Elastic Beanstalk usa propriedades de ambiente, como RDS_HOSTNAME, para

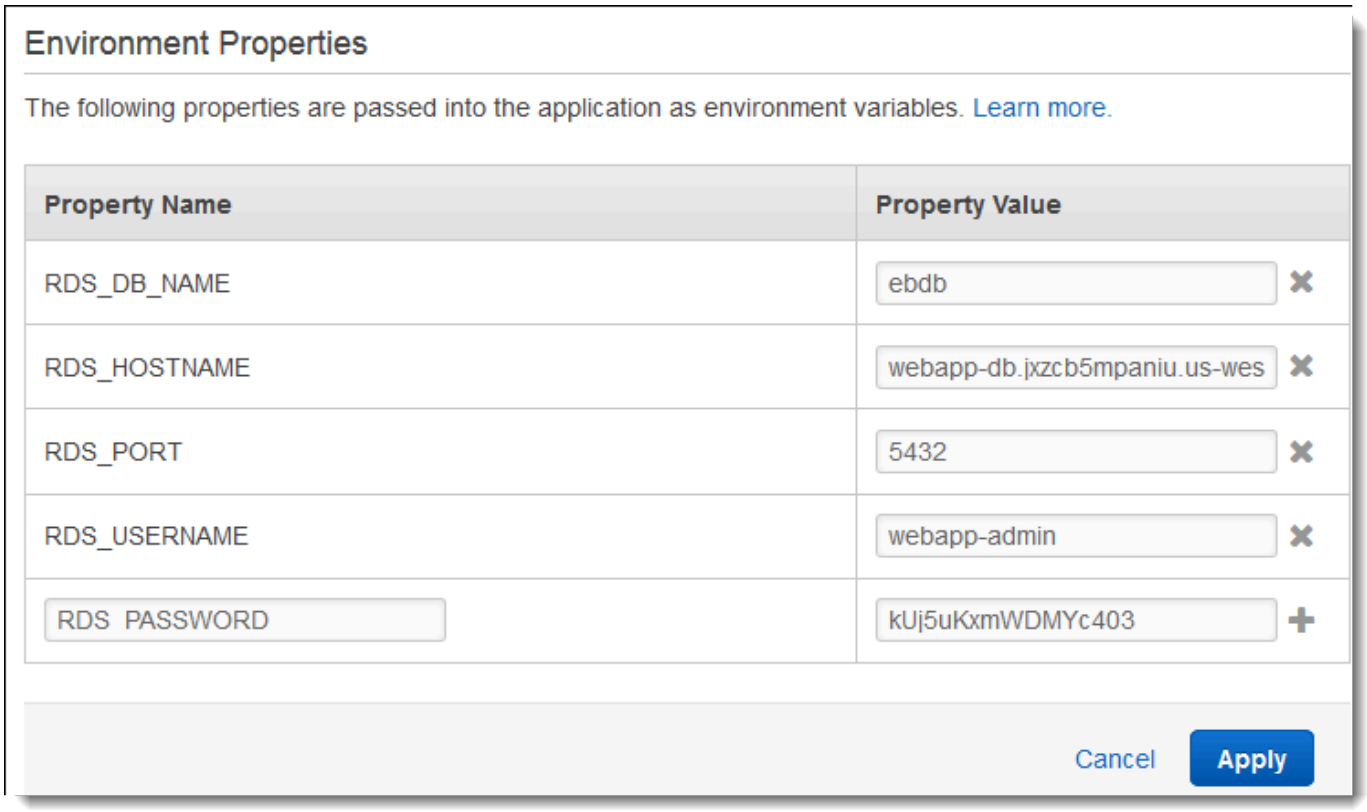
passar informações de conexão à aplicação. Você pode usar as mesmas propriedades. Fazendo isso, você usa o mesmo código de aplicação com as instâncias de banco de dados integradas e externas. Ou, alternativamente, você pode escolher seus próprios nomes de propriedade.

Como configurar as propriedades do ambiente para uma instância de banco de dados do Amazon RDS

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Na seção Environment properties, defina as variáveis que seu aplicativo lê para construir uma string de conexão. Para compatibilidade com ambientes que têm uma instância de banco de dados do RDS integrada, use os seguintes nomes e valores. É possível encontrar todos os valores, exceto sua senha, no console do RDS.

Nome da propriedade	Descrição	Valor da propriedade
`RDS_HOSTNAME`	O nome do host da instância de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) no console do Amazon RDS: Endpoint.
`RDS_PORT`	A porta na qual a instância de banco de dados aceita conexões. O valor padrão varia entre os mecanismos de banco de dados.	Na guia Connectivity & security (Conectividade e segurança) do console do Amazon RDS: Port (Porta).
`RDS_DB_NAME`	O nome do banco de dado, **ebdb**.	Na guia Configuration (Configuração) no console do Amazon RDS: DB Name (Nome do banco de dados).

Nome da propriedade	Descrição	Valor da propriedade
RDS_USERNAME	O nome de usuário que você configurou para seu banco de dados.	Na guia Configuration (Configuração) do console do Amazon RDS: Nome de usuário primário.
RDS_PASSWORD	A senha que você configurou para seu banco de dados.	Não disponível para referência no console do Amazon RDS.



6. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Se você ainda não programou a aplicação para ler as propriedades de ambiente e criar uma sequência de conexão, consulte os seguintes tópicos específicos de cada linguagem para obter instruções:

- Java SE: Conectar a um banco de dados (plataformas Java SE)

- Java com Tomcat: Conectar a um banco de dados (plataformas Tomcat)
- Node.js: Conectar-se a um banco de dados
- .NET: Conectar-se a um banco de dados
- PHP: Conectando-se a um banco de dados com um PDO ou My SQLi
- Python: Conectar-se a um banco de dados
- Ruby: Conectar-se a um banco de dados

Finalmente, dependendo de quando seu aplicativo lê as variáveis de ambiente, pode ser necessário reiniciar o servidor de aplicativo nas instâncias em seu ambiente.

Para reiniciar os servidores de aplicativos do ambiente

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. Escolha Actions (Ações) e depois Restart app server(s) (Reiniciar servidores de aplicações).

Armazenando as credenciais do Amazon RDS em AWS Secrets Manager

Este tópico explica como AWS Secrets Manager você pode melhorar sua postura de segurança para recuperação de credenciais com o Elastic Beanstalk. Ele também fornece referências a recursos específicos que podem ajudar você a configurar credenciais para sua aplicação Elastic Beanstalk.

AWS Secrets Manager ajuda você a melhorar sua postura de segurança, fornecendo a capacidade de armazenar e recuperar credenciais criptografadas. Armazenar as credenciais no Secrets Manager evita um possível comprometimento por qualquer pessoa que possa inspecionar sua aplicação ou os componentes relacionados. Seu código pode fazer uma chamada em runtime para o serviço Secrets Manager para recuperar credenciais dinamicamente. O Secrets Manager também oferece atributos como componentes de cache secreto do lado do cliente para linguagens de runtime, que incluem Python, Go e Java.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do AWS Secrets Manager :

- Como o Amazon RDS usa o AWS Secrets Manager
- Crie um segredo AWS Secrets Manager de banco de dados
- Recupere segredos de AWS Secrets Manager

Limpar uma instância externa do Amazon RDS

Quando você conecta uma instância externa do Amazon RDS ao ambiente do Elastic Beanstalk, a instância de banco de dados depende do ciclo de vida do ambiente e não é excluída quando o ambiente é terminado. Para garantir que as informações pessoais que você tenha armazenado na instância de banco de dados não sejam retidas desnecessariamente, exclua todos os registros que não deseja mais. Como alternativa, exclua a instância do banco de dados.

Usar o Elastic Beanstalk com o Amazon S3

Este tópico explica como o Elastic Beanstalk utiliza o Amazon Simple Storage Service (Amazon S3) e os tipos de objetos que ele armazena em buckets do S3. Ele também observa quais objetos você deve excluir manualmente depois de encerrar o ambiente do Elastic Beanstalk e fornece instruções para isso.

O bucket de contas de clientes do Amazon S3 do Elastic Beanstalk

O Elastic Beanstalk cria um bucket criptografado do Amazon S3 `elasticbeanstalk-`*`region`*`-`*`account-id` para cada região na qual você cria ambientes. Sua AWS conta é proprietária desse bucket. O Elastic Beanstalk armazena arquivos de configuração temporários e outros objetos para a operação adequada do seu aplicativo nesse bucket. O Elastic Beanstalk ACLs requer habilitação para buckets gerenciados por serviços e, portanto, habilita a Lista de Controle de Acesso (ACL) desse bucket.

Esteja ciente de que o Amazon S3 desativa as listas de controle de acesso do bucket (ACLs) por padrão. Além disso, o tópico de visão geral da ACL no Guia do usuário do Amazon S3 recomenda que você ACLs continue desativado, exceto em casos de uso específicos. Os buckets gerenciados por serviços do Elastic Beanstalk se enquadram em um caso de uso que precisa ser ativado. ACLs Para manter a segurança, as implantações do Elastic Beanstalk garantem que esse bucket seja de propriedade da conta que executa o aplicativo.

O Elastic Beanstalk retém a criptografia padrão fornecida pelos buckets do Amazon S3. Para obter mais informações sobre criptografia de bucket, consulte Criptografia padrão do Amazon S3 no Guia do usuário do Amazon Simple Storage Service.

Conteúdo do bucket de contas de clientes Amazon S3 do Elastic Beanstalk

A tabela a seguir lista alguns objetos que o Elastic Beanstalk armazena no bucket da sua conta de cliente. A tabela também mostra quais objetos devem ser excluídos manualmente. Para evitar custos

de armazenamento desnecessários, e para garantir que as informações pessoais não sejam retidas, exclua manualmente esses objetos quando não forem mais necessários.

Objeto	Quando é armazenado?	Quando é excluído?
Versões do aplicativo	Quando você cria um ambiente ou implanta o código da aplicação em um ambiente existente, o Elastic Beanstalk armazena uma versão da aplicação no Amazon S3 e associa-a ao ambiente.	Durante a exclusão do aplicativo e de acordo com o Ciclo de vida da versão.
Pacotes de origem	Quando você faz upload de uma nova versão da aplicação usando o console do Elastic Beanstalk ou a CLI do EB, o Elastic Beanstalk armazena uma cópia dela no Amazon S3 e a define como o pacote de origem do ambiente.	Manualmente. Ao excluir uma versão do aplicativo, você pode escolher Delete versions from Amazon S3 (Excluir versões do Amazon S3) e também excluir o pacote de origem relacionado. Para obter detalhes, consulte Gerenciar versões de aplicativos.
Plataformas personalizadas	Quando uma plataforma personalizada é criada, o Elastic Beanstalk armazena temporariamente dados relacionados no Amazon S3.	Após a conclusão bem-sucedida da criação da plataforma personalizada.
Arquivos de log	É possível solicitar ao Elastic Beanstalk para recuperar arquivos de log de instâncias (logs finais ou em pacote) e armazená-los no Amazon S3. Você também pode habilitar a alternância de logs e configurar o ambiente para publicá-los no Amazon S3 automaticamente depois de alternados.	Logs finais e em pacote: 15 minutos depois de serem criados. Logs alternados: manualmente.
Configurações salvas	Manualmente.	Manualmente.

Excluir objetos no bucket do Amazon S3 do Elastic Beanstalk

Quando você encerra um ambiente ou exclui uma aplicação, o Elastic Beanstalk exclui a maioria dos objetos relacionados do Amazon S3. Para minimizar os custos de armazenamento de um aplicativo em execução, exclua rotineiramente objetos dos quais seu aplicativo não precisa. Além disso, preste atenção aos objetos que você precisa excluir manualmente, conforme listado em Conteúdo do bucket de contas de clientes Amazon S3 do Elastic Beanstalk. Para que as informações privadas não sejam retidas desnecessariamente, exclua esses objetos quando não precisar mais deles.

- Exclua as versões de aplicativos que você não pretende mais usar em seu aplicativo. Ao excluir uma versão da aplicação, você pode selecionar Delete versions from Amazon S3 (Excluir versões do Amazon S3) para também excluir o pacote de origem relacionado: uma cópia do código-fonte da aplicação e os arquivos de configurações dos quais o Elastic Beanstalk fez upload para o Amazon S3 quando você implantou uma aplicação ou fez upload de uma versão da aplicação. Para saber como excluir uma versão do aplicativo, consulte Gerenciar versões de aplicativos.
- Excluir logs alternados dos quais não precisa. Como alternativa, baixe-os ou mova-os para o Amazon S3 Glacier para uma análise mais aprofundada.
- Exclua as configurações salvas que você não usará mais em nenhum ambiente.

Excluir o bucket do Amazon S3 do Elastic Beanstalk

Quando o Elastic Beanstalk cria um bucket, ele também cria uma política de bucket que se aplica ao novo bucket. Esta política atende a dois propósitos:

- Permitir que os ambientes escrevam no bucket.
- Evitar a exclusão acidental do bucket.

Devido à política aplicada pelo Elastic Beanstalk aos buckets que cria para seus ambientes, você não tem permissão para excluir esses buckets, a menos que exclua deliberadamente primeiro a política do bucket. É possível a política do bucket na seção Permissões das propriedades do bucket no console do Amazon S3.

> **Warning**
>
> Recomendamos excluir objetos desnecessários específicos do bucket do Amazon S3 do Elastic Beanstalk em vez de excluir o bucket inteiro.

Se você excluir um bucket que o Elastic Beanstalk criou em sua conta e ainda tiver aplicações existentes e ambientes em execução na região correspondente, as aplicações poderão parar de funcionar corretamente. Por exemplo:

- Quando um ambiente se expande, o Elastic Beanstalk deve ser capaz de encontrar a versão do aplicativo do ambiente no bucket do Amazon S3 e usá-la para iniciar novas instâncias da Amazon. EC2
- Quando uma plataforma personalizada é criada, o Elastic Beanstalk usa armazenamento temporário do Amazon S3 durante o processo de criação.

Para obter mais informações sobre as implicações da exclusão de um bucket do S3, consulte as considerações listadas em Exclusão de um bucket no Guia do usuário do Amazon S3.

Como excluir um bucket de armazenamento do Elastic Beanstalk (console)

O procedimento geral para excluir um bucket do S3 também é descrito em Exclusão de um bucket no Guia do usuário do Amazon S3. Como estamos excluindo um bucket criado pelo Elastic Beanstalk no procedimento a seguir, incluímos etapas adicionais para excluir primeiro a política do bucket.

1. Abra o console Amazon S3.
2. Abra a página de armazenamento do bucket do Elastic Beanstalk escolhendo o nome do bucket.
3. Escolha a aba Permissões.
4. Escolha Política do bucket.
5. Escolha Excluir.
6. Volte para a página principal do console do Amazon S3 e, depois, selecione o bucket de armazenamento do Elastic Beanstalk.
7. Escolha Excluir bucket.
8. Confirme se deseja excluir o bucket inserindo o nome do bucket no campo de texto e, em seguida, escolha Excluir bucket.

Usando o Elastic AWS Secrets ManagerAWS Systems Manager Beanstalk com um Parameter Store

Este tópico explica como você pode usar AWS Secrets Manager o AWS Systems Manager Parameter Store com seu ambiente do Elastic Beanstalk para armazenar e recuperar com segurança informações confidenciais, como credenciais e chaves de API. Seu aplicativo pode recuperar segredos e parâmetros armazenados diretamente desses armazenamentos, usando as ferramentas de linha de comando APIs ou as ferramentas de linha de comando desses serviços.

O Elastic Beanstalk também oferece a capacidade de referenciar dados do Secrets Manager e do Systems Manager Parameter Store em variáveis de ambiente. Essa é uma opção segura para que seu aplicativo acesse nativamente segredos e parâmetros armazenados por esses serviços sem precisar gerenciar chamadas de API para eles.

Tópicos

- Buscando segredos e parâmetros para variáveis de ambiente do Elastic Beanstalk
- Permissões do IAM necessárias para que o Elastic Beanstalk acesse segredos e parâmetros
- Usando um AWS Secrets Manager repositório de AWS Systems Manager parâmetros
- Solução de problemas: integração de segredos com variáveis de ambiente do Elastic Beanstalk

Buscando segredos e parâmetros para variáveis de ambiente do Elastic Beanstalk

O Elastic Beanstalk pode buscar AWS Secrets Manager valores de um Parameter Store durante a inicialização AWS Systems Manager da instância e atribuí-los às variáveis de ambiente para seu aplicativo usar.

Os pontos a seguir resumem a configuração, a sincronização e o acesso para usar variáveis de ambiente como segredos:

- Configure suas variáveis de ambiente para armazenar segredos especificando os Amazon Resource Names (ARNs) para os segredos e parâmetros que eles armazenarão.
- Quando os valores secretos são atualizados ou alternados no Secrets Manager ou no Systems Manager Parameter Store, você deve atualizar manualmente suas variáveis de ambiente.
- As variáveis de ambiente secrets estão disponíveis para comandos de contêiner ebextension e ganchos de plataforma.

Versões compatíveis da plataforma

As versões da plataforma lançadas em ou após 26 de março de 2025 oferecem suporte a AWS Secrets Manager segredos e AWS Systems Manager parâmetros do Parameter Store configurados como variáveis de ambiente.

> **Note**
>
> Com exceção das plataformas docker baseadas em Docker e ECS, as versões da plataforma Amazon Linux 2 não oferecem suporte a valores de variáveis de várias linhas. Para obter mais informações sobre o suporte a variáveis de várias linhas, consulteValores de várias linhas.

Tópicos

- Preços
- Configure segredos como variáveis de ambiente do Elastic Beanstalk
- Melhores práticas para sincronização de segredos com variáveis de ambiente do Elastic Beanstalk
- Valores de várias linhas nas variáveis de ambiente do Amazon Linux 2

Preços

Cobranças padrão se aplicam ao uso do Secrets Manager e do Systems Manager Parameter Store. Para obter mais informações sobre preços, consulte os seguintes sites:

- AWS Secrets Manager preços
- AWS Systems Manager preços (selecione Parameter Store na lista de conteúdo)

O Elastic Beanstalk não cobra que seu aplicativo faça referência a segredos do ambiente por meio de variáveis de ambiente. No entanto, cobranças padrão se aplicam às solicitações que o Elastic Beanstalk faz a esses serviços em seu nome.

Configure segredos como variáveis de ambiente do Elastic Beanstalk

Você pode usar o console do Elastic Beanstalk, os arquivos `.ebextensions` de configuração, AWS a CLI e o SDK para configurar segredos AWS e parâmetros como variáveis de ambiente.

Tópicos

- Pré-requisitos
- Utilizar o console
- Configuração usando arquivos em extensões.ebextensions
- Configuração usando o AWS CLI
- Configuração usando o AWS SDK

Pré-requisitos

Antes de configurar suas variáveis de ambiente para referenciar segredos, primeiro você precisará concluir as etapas a seguir.

Procedimento geral antes da configuração da variável de ambiente

1. Crie os segredos do Secrets Manager ou os parâmetros do Parameter Store para armazenar seus dados confidenciais. Para obter mais informações, consulte um ou os dois tópicos a seguir:
   - Criando segredos em the section called “Usando o Secrets Manager”
   - Criação de parâmetros em the section called “Usando Systems Manager Parameter Store”
2. Configure as permissões necessárias do IAM para as EC2 instâncias do seu ambiente para buscar os segredos e os parâmetros. Para obter mais informações, consulte Permissões obrigatórias do IAM.

Utilizar o console

Você pode usar o console do Elastic Beanstalk para configurar segredos como variáveis de ambiente.

Para configurar segredos como variáveis de ambiente no console do Elastic Beanstalk

1. Abra o console do Elastic Beanstalk e, na lista Regiões, selecione sua. Região da AWS
2. No painel de navegação, selecione Ambientes e selecione o nome do ambiente na lista.
3. No painel de navegação, escolha Configuration (Configuração).
4. Na categoria de configuração Updates, monitoring, and logging (Atualizações, monitoramento e logs), escolha Edit (Editar).
5. Role para baixo até Variáveis de ambiente de tempo de execução.
6. Selecione Adicionar variável de ambiente.

7. Em Source, selecione Secrets Manager ou SSM Parameter Store.

> **Note**
>
> Para obter mais informações sobre a opção Texto sem formatação no menu suspenso, consulte. Configurar propriedades do ambiente (variáveis de ambiente)

8. Em Nome da variável de ambiente, insira o nome da variável de ambiente para manter o valor secreto ou do parâmetro.
9. Em Valor da variável de ambiente, insira o ARN do parâmetro Systems Manager Parameter Store ou o segredo do Secrets Manager. Durante a inicialização da instância, o Elastic Beanstalk iniciará o valor da variável inserida na Etapa 8 com o valor armazenado nesse recurso ARN.

   O console valida se o valor inserido é um formato ARN válido para a loja selecionada na Etapa 7. No entanto, isso não valida a existência do recurso especificado pelo ARN ou se você tem as permissões necessárias do IAM para acessá-lo.
10. Se precisar adicionar mais variáveis, repita a Etapa 6 até a Etapa 9.
11. Para salvar as alterações, escolha Apply (Aplicar) na parte inferior da página.

Configuração usando arquivos em extensões.ebextensions

Você pode usar os arquivos de configuração do Elastic Beanstalk para configurar segredos como variáveis de ambiente. Use o aws: elasticbeanstalk: aplicação: segredos do meio ambiente namespace para definir as propriedades do ambiente.

Example .ebextensions/options.config para segredos do ambiente (sintaxe abreviada)

```
option_settings:
  aws:elasticbeanstalk:application:environmentsecrets:
    MY_SECRET: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret
    MY_PARAMETER: arn:aws:ssm:us-east-1:111122223333:parameter/myparam
```

Example .ebextensions/options.config para segredos do ambiente (sintaxe padrão)

```
option_settings:
  - namespace: aws:elasticbeanstalk:application:environmentsecrets
    option_name: MY_SECRET
    value: arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret
```

```
  - namespace: aws:elasticbeanstalk:application:environmentsecrets
    option_name: MY_PARAMETER
    value: arn:aws:ssm:us-east-1:111122223333:parameter/myparam
```

Configuração usando o AWS CLI

Você pode usar a interface de linha de AWS comando (AWS CLI) para configurar segredos como variáveis de ambiente do Elastic Beanstalk. Esta seção fornece exemplos dos comandos create-environment e update-environment com o namespace. aws: elasticbeanstalk: aplicação: segredos do meio ambiente Quando o Elastic Beanstalk EC2 inicializa as instâncias dos ambientes referenciados por esses comandos, ele inicializa as variáveis de ambiente com o segredo obtido e os valores dos parâmetros. Ele obtém esses valores dos respectivos repositórios de parâmetros ARNs do Secrets Manager e do Systems Manager.

Os dois exemplos a seguir usam o comando create-environment para adicionar um segredo e um parâmetro, configurados como variáveis de ambiente chamadas e. `MY_SECRET` `MY_PARAMETER`

Example de criar um ambiente com segredos configurados como variáveis de ambiente (opções de namespace embutidas)

```
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 20" \
--option-settings \
Namespace=aws:elasticbeanstalk:application:environmentsecrets,OptionName=MY_SECRET,Value=arn:aw
east-1:111122223333:secret:mysecret \
Namespace=aws:elasticbeanstalk:application:environmentsecrets,OptionName=MY_PARAMETER,Value=arn
east-1:111122223333:parameter/myparam
```

Outra opção é usar um arquivo `options.json` para especificar as opções de namespace em vez de incluí-las inline.

Example de criar ambiente com segredos configurados como variáveis de ambiente (opções de namespace no arquivo) **`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 20" \
--option-settings file://options.json
```

Example

```
### example options.json ###
[
  {
    "Namespace": "aws:elasticbeanstalk:application:environmentsecrets",
    "OptionName": "MY_SECRET",
    "Value": "arn:aws:secretsmanager:us-east-1:111122223333:secret:mysecret"
  },
  {
    "Namespace": "aws:elasticbeanstalk:application:environmentsecrets",
    "OptionName": "MY_PARAMETER",
    "Value": "arn:aws:ssm:us-east-1:111122223333:parameter/myparam"
  }
]
```

O próximo exemplo configura variáveis de ambiente, chamadas `MY_SECRET` e`MY_PARAMETER`, para armazenar um segredo e um parâmetro para um ambiente existente. O comando update-environment transmite opções com a mesma sintaxe do `create-environment` comando, em linha ou com um arquivo. `options.json` O exemplo a seguir demonstra o comando usando o mesmo `options.json` arquivo que também é usado no exemplo anterior.

Example do ambiente de atualização com segredos configurados como variáveis de ambiente (opções de namespace no arquivo) **`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 20" \
--option-settings file://options.json
```

Configuração usando o AWS SDK

Você pode configurar segredos e parâmetros como variáveis de ambiente usando AWS SDKso. Semelhante aos `create-environment` AWS CLI comandos `update-`

`environment` e mencionados na seção anterior, você pode usar as ações CreateEnvironmente UpdateEnvironmentda API. Use o parâmetro de `OptionSettings` solicitação para especificar as opções do aws: elasticbeanstalk: aplicação: segredos do meio ambiente namespace.

Melhores práticas para sincronização de segredos com variáveis de ambiente do Elastic Beanstalk

Este tópico recomenda as melhores práticas para seu aplicativo usar segredos de ambiente com o Secrets Manager ou o Systems Manager Parameter Store. Seu aplicativo Elastic Beanstalk não receberá automaticamente valores atualizados se os dados do armazenamento secreto forem atualizados ou alternados. O Elastic Beanstalk só extrai segredos para as variáveis de ambiente no momento da inicialização da instância.

Atualizando suas variáveis de ambiente

Para acionar seu ambiente do Elastic Beanstalk para recuperar os valores mais recentes dos segredos em seus repositórios secretos, recomendamos que você execute a operação ou. `UpdateEnvironment` `RestartAppServer` Você pode executar essas operações usando o console do Elastic Beanstalk AWS CLI, o ou a API do Elastic Beanstalk. Para obter mais informações, consulte AWS CLI exemplos do Elastic Beanstalk ou da API Reference.AWS Elastic Beanstalk

Gerenciando efeitos de escalonamento automático na sincronização secreta

Se um evento de escalabilidade ou substituição de instância ocorrer após as atualizações do armazenamento secreto, a nova instância que surgir terá os valores secretos mais recentes do Secrets Manager ou do Systems Manager Parameter Store. Esse evento pode ocorrer mesmo que nem todas as outras instâncias no ambiente tenham sido atualizadas para recuperar os novos segredos.

> **Important**
>
> Você deve garantir que seu aplicativo seja capaz de usar dois valores secretos diferentes para a mesma variável de ambiente. Isso acomoda eventos em que uma atualização secreta ocorre no Secrets Manager ou no Systems Manager Parameter Store, seguida por uma expansão horizontal ou substituição de instância em seu ambiente, enquanto as outras instâncias estão pendentes de atualização da variável de ambiente. Durante o período de espera pela atualização, nem todas as instâncias do ambiente terão os mesmos valores para as variáveis de ambiente do armazenamento secreto.

Um exemplo desse caso de uso é a rotação de credenciais do banco de dados. Quando um evento de expansão segue a rotação de credenciais, os segredos do ambiente referenciados pelas instâncias recém-inicializadas contêm as credenciais atualizadas do banco de dados. No entanto, os segredos do ambiente referenciados pelas instâncias existentes mantêm o valor antigo até serem atualizados pelas `UpdateEnvironment` operações ou. `RestartAppServer`

Valores de várias linhas nas variáveis de ambiente do Amazon Linux 2

Os valores de várias linhas são compostos por mais de uma linha e incluem um caractere de nova linha. Com exceção das plataformas Docker e Docker baseadas em ECS, as plataformas executadas no Amazon Linux 2 não oferecem suporte a valores de várias linhas para variáveis de ambiente

> **Note**
>
> O Elastic Beanstalk falhará na implantação dos ambientes afetados se detectar um valor de várias linhas.

As opções a seguir podem servir como soluções alternativas ou soluções para o problema de várias linhas:

- Atualize seu ambiente Amazon Linux 2 para o Amazon Linux 2023. Para obter mais informações, consulte Migração do Amazon Linux 2 para o Amazon Linux 2023.
- Remova caracteres de nova linha de seus valores secretos. Um exemplo de abordagem é codificar seus valores em Base64 antes de armazená-los no armazenamento secreto. Em seguida, seu aplicativo precisaria decodificar o valor de volta para o formato original ao referenciá-lo a partir da variável secreta do ambiente.
- Crie o código do seu aplicativo para recuperar os dados diretamente do Secrets Manager ou do Systems Manager Parameter Store. Para obter mais informações, consulte Recuperação de segredos em Usando o Secrets Manager ou Recuperação de parâmetros. Usando Systems Manager Parameter Store

Permissões do IAM necessárias para que o Elastic Beanstalk acesse segredos e parâmetros

Você deve conceder as permissões necessárias às EC2 instâncias do seu ambiente para buscar os segredos e os parâmetros do AWS Systems Manager Parameter Store. AWS Secrets Manager As permissões são fornecidas às EC2 instâncias por meio de uma função de perfil da EC2 instância.

As seções a seguir listam as permissões específicas que você precisa adicionar a um perfil de EC2 instância, dependendo do serviço que você usa. Siga as etapas fornecidas em Atualizar a política de permissões para uma função no Guia do usuário do IAM para adicionar essas permissões.

> **Permissões do IAM para a plataforma Docker gerenciada pelo ECS**
>
> A plataforma Docker gerenciada pelo ECS exige permissões adicionais do IAM além das fornecidas neste tópico. Para obter mais informações sobre todas as permissões necessárias para que seu ambiente de plataforma Docker gerenciado pelo ECS ofereça suporte à integração das variáveis de ambiente do Elastic Beanstalk com segredos, consulte. Formato ARN da função de execução

Tópicos

- Permissões do IAM necessárias para o Secrets Manager
- Permissões necessárias do IAM Systems Manager Parameter Store

Permissões do IAM necessárias para o Secrets Manager

As permissões a seguir concedem acesso para buscar segredos criptografados na AWS Secrets Manager loja:

- gerente de segredos: GetSecretValue
- kms:Decrypt

A permissão para descriptografar um só AWS KMS key é necessária se seu segredo usar uma chave gerenciada pelo cliente em vez da chave padrão. A adição do ARN da sua chave personalizada adiciona a permissão para descriptografar a chave gerenciada pelo cliente.

Example política com permissões de chave Secrets Manager e KMS

```
{
    "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"
            ]
        }
    ]
}
```

Permissões necessárias do IAM Systems Manager Parameter Store

As permissões a seguir concedem acesso para buscar parâmetros criptografados do AWS Systems Manager Parameter Store:

- sms: GetParameter
- kms:Decrypt

A permissão para descriptografar um AWS KMS key é necessária somente para tipos de `SecureString` parâmetros que usam uma chave gerenciada pelo cliente em vez de uma chave padrão. A adição do ARN da sua chave personalizada adiciona a permissão para descriptografar a chave gerenciada pelo cliente. Os tipos de parâmetros regulares que não são criptografados `String` e `StringList` não precisam de um AWS KMS key.

Example política com Systems Manager e AWS KMS principais permissões

```
{
    "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"
            ]
        }
    ]
}
```

Usando um AWS Secrets Manager repositório de AWS Systems Manager parâmetros

Este tópico fornece uma breve introdução ao AWS Systems Manager Parameter Store, informações sobre preços e referências para aprender mais sobre como criar e recuperar segredos, usando o console e as opções programáticas. AWS Secrets Manager

Sobre Secrets Manager

AWS Secrets Manager ajuda você a gerenciar, recuperar e alternar segredos ao longo de seus ciclos de vida. Exemplos de dados secretos que você pode gerenciar incluem credenciais de banco de dados, credenciais de aplicativos, OAuth tokens e chaves de API. O Secrets Manager permite que você configure um cronograma de rotação automática para seus segredos.

Sobre o Systems Manager Parameter Store

O Parameter Store é uma ferramenta em AWS Systems Manager. Oferece armazenamento hierárquico seguro para gerenciamento de dados de configuração e gerenciamento de segredos. Você pode gerenciar dados de configuração importantes como valores de parâmetros. Exemplos de dados que você pode gerenciar com o Parameter Store incluem Amazon Machine Image (AMI) IDs, códigos de licença, senhas e cadeias de banco de dados.

> **Preços**
>
> Cobranças padrão se aplicam ao uso do Secrets Manager e do Systems Manager Parameter Store. Para obter mais informações sobre preços, consulte os seguintes sites:
>
> - AWS Secrets Manager preços
> - AWS Systems Manager preços (selecione Parameter Store na lista de conteúdo)

Tópicos

- Usando o Secrets Manager para criar e recuperar segredos
- Usando o Systems Manager Parameter Store para criar e recuperar parâmetros

Usando o Secrets Manager para criar e recuperar segredos

Você pode criar e recuperar segredos do Secrets Manager usando o AWS Secrets Manager console AWS CLI, o ou o AWS SDK. Consulte os recursos a seguir para saber mais sobre os diferentes métodos para criar e recuperar segredos do Secrets Manager.

Criando segredos

- Console — Crie um AWS Secrets Manager segredo (console) no Guia do AWS Secrets Manager usuário
- AWS CLI — AWS CLI no Guia do Usuário AWS Secrets Manager
- AWS SDK — AWS SDK no Guia do Usuário AWS Secrets Manager

Recuperando segredos

- Console — Obtenha um valor secreto (console) no Guia AWS Secrets Manager do usuário
- AWS CLI — Obtenha um valor secreto (AWS CLI) no Guia do usuário AWS Secrets Manager
- AWS SDK — exemplos de código para uso do Secrets Manager AWS SDKs na biblioteca de códigos de exemplos de códigos do AWS SDK
- Outros métodos — Obtenha segredos AWS Secrets Manager no Guia do AWS Secrets Manager usuário

Para obter mais informações sobre AWS Secrets Manager, consulte O que é AWS Secrets Manager? no Guia do AWS Secrets Manager usuário.

Usando o Systems Manager Parameter Store para criar e recuperar parâmetros

Você pode criar e recuperar parâmetros do Parameter Store usando o AWS Systems Manager console AWS CLI, o ou o AWS SDK. Consulte os recursos a seguir para saber mais sobre os diferentes métodos para criar e recuperar parâmetros do Parameter Store.

Criar parâmetros

- Console — Crie um parâmetro do Systems Manager (console) no Guia AWS Systems Manager do usuário
- AWS CLI — Crie um parâmetro do Systems Manager (AWS CLI) no Guia do usuário AWS Systems Manager
- AWS SDK — Use PutParameter com um AWS SDK ou AWS CLI na biblioteca de códigos de exemplos de códigos do AWS SDK

Recuperando parâmetros

- Console — Procurando por um parâmetro (console) no Guia do AWS Systems Manager usuário
- AWS CLI — Use GetParameter com um AWS SDK ou AWS CLI na biblioteca de códigos de exemplos de códigos do AWS SDK
- AWS SDK — Use GetParameter com um AWS SDK ou AWS CLI na biblioteca de códigos de exemplos de códigos do AWS SDK

Para obter mais informações, consulte o Armazenamento de parâmetros do AWS Systems Manager, no Guia do usuário do AWS Systems Manager .

Solução de problemas: integração de segredos com variáveis de ambiente do Elastic Beanstalk

Evento: falha na implantação da instância ao obter um ou mais segredos

Essa mensagem indica que o Elastic Beanstalk não conseguiu buscar um ou mais dos segredos especificados durante a implantação do seu aplicativo.

- Verifique se os recursos especificados pelos valores de ARN na configuração da variável de ambiente existem.
- Confirme se sua função de perfil da instância do EC2 Elastic Beanstalk tem as permissões necessárias do IAM para acessar os recursos.
- Se esse evento foi acionado por meio da `RestartAppServer` operação, depois que o problema for corrigido, tente fazer a `RestartAppServer` chamada novamente para resolver o problema.
- Se o evento foi acionado por meio de uma `UpdateEnvironment` chamada, repita a `UpdateEnvironment` operação.

Para ver exemplos desses comandos, veja AWS CLI exemplos do Elastic Beanstalk. Para obter mais informações sobre as ações da API para essas operações, consulte a Referência AWS Elastic Beanstalk da API.

Evento: a implantação da instância detectou um ou mais valores de ambiente de várias linhas, que não são compatíveis com essa plataforma

As variáveis de várias linhas não são compatíveis com as plataformas Amazon Linux 2, exceto as plataformas Docker gerenciadas pelo Docker e pelo ECS. Para ver as opções disponíveis para continuar, consulteValores de várias linhas.

Evento: CreateEnvironment falha quando um segredo é especificado

Quando uma `CreateEnvironment` falha e você tem segredos como variáveis de ambiente, você precisa resolver o problema subjacente e usá-lo `UpdateEnvironment` para concluir a configuração do ambiente. Não use`RestartAppServer`, pois não será suficiente para melhorar o meio ambiente nessa situação. Para ver exemplos desses comandos, veja AWS CLI exemplos do Elastic Beanstalk. Para obter mais informações sobre as ações da API para essas operações, consulte a Referência AWS Elastic Beanstalk da API.

Usar o Elastic Beanstalk com Amazon VPC

Este tópico explica os benefícios do uso de endpoints de VPC com o Elastic Beanstalk e os diferentes tipos de configurações que você pode implementar.

Você pode usar uma Amazon Virtual Private Cloud (Amazon VPC) para criar uma rede segura para seu aplicativo Elastic Beanstalk e recursos relacionados. AWS Ao criar seu ambiente, você escolhe a VPC, as sub-redes e os security groups que serão usados nas suas instâncias de aplicativo e no load balancer. Você pode usar qualquer configuração de VPC que desejar, desde que ela atenda aos seguintes requisitos.

Requisitos da VPC

- Acesso à Internet: as instâncias podem ter acesso à Internet por meio de um dos métodos a seguir.
  - Sub-rede pública: as instâncias têm um endereço IP público e usam um gateway da Internet para acessar a Internet.
  - Sub-rede privada: as instâncias usam um dispositivo NAT para acessar a Internet.

> **Note**
>
> Se você configurar VPC endpoints em sua VPC para se conectar aos serviços `elasticbeanstalk` e `elasticbeanstalk-health`, o acesso à Internet será opcional e será necessário somente se o aplicativo precisar especificamente dele. Sem VPC endpoints, sua VPC deve ter acesso à Internet.
> A VPC padrão que o Elastic Beanstalk configura para você fornece acesso à internet.

O Elastic Beanstalk não é compatível com as configurações de proxy, como `HTTPS_PROXY` para configurar um proxy da Web.

- NTP: as instâncias no ambiente do Elastic Beanstalk usam o Network Time Protocol (NTP) para sincronizar o relógio do sistema. Caso as instâncias não consigam se comunicar na porta UDP 123, o relógio pode estar fora de sincronia, causando problemas de relatórios de integridade do Elastic Beanstalk. Certifique-se de que seus grupos de segurança e rede VPC ACLs permitam tráfego UDP de entrada e saída na porta 123 para evitar esses problemas.

O elastic-beanstalk-samplesrepositório fornece AWS CloudFormation modelos que você pode usar para criar uma VPC para uso com seus ambientes do Elastic Beanstalk.

Para criar recursos com um AWS CloudFormation modelo

1. Clone o repositório de exemplos ou faça download de um modelo usando os links no arquivo README.
2. Abra o console de AWS CloudFormation.
3. Selecione Criar pilha.
4. Escolha Upload a template to Amazon S3.
5. Escolha Upload file e faça upload do arquivo de modelo da sua máquina local.
6. Escolha Next e siga as instruções para criar uma pilha com os recursos no modelo.

Quando a criação da pilha for concluída, verifique a guia Saídas para encontrar o ID da VPC e a sub-rede. IDs Use-os para configurar a VPC no novo assistente de ambiente categoria de configuração de rede.

Tópicos

- VPC pública
- VPC pública/privada
- VPC privada
- Exemplo: Iniciar uma aplicação do Elastic Beanstalk em uma VPC com bastion hosts
- Exemplo: Iniciar um Elastic Beanstalk em uma VPC com o Amazon RDS
- Usar o Elastic Beanstalk com VPC endpoints
- Usar políticas de endpoint para controlar o acesso com VPC endpoints

VPC pública

AWS CloudFormation modelo — vpc-public.yaml

Configurações (balanceamento de carga)

- Visibilidade do balanceador de carga: público
- Sub-redes do balanceador de carga: as duas sub-redes públicas
- IP público da instância: habilitado
- Sub-redes da instância: as duas sub-redes públicas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Configurações (instância única)

- Sub-redes da instância: uma das sub-redes públicas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Um layout somente público da VPC inclui uma ou mais sub-redes públicas, um gateway da Internet e um grupo de segurança padrão que permite o tráfego entre recursos na VPC. Quando você cria um ambiente na VPC, o Elastic Beanstalk cria recursos adicionais que variam dependendo do tipo de ambiente.

Recursos da VPC

- Instância única: o Elastic Beanstalk cria um grupo de segurança para a instância da aplicação que permite o tráfego na porta 80 da Internet e atribui à instância um IP elástico para fornecer um

endereço IP público. O nome de domínio do ambiente é resolvido para o endereço IP público da instância.

- Balanceamento de carga:o Elastic Beanstalk cria um grupo de segurança para o balanceador de carga que permite o tráfego na porta 80 da Internet e um grupo de segurança para as instâncias da aplicação que permite o tráfego do grupo de segurança do balanceador de carga. O nome de domínio do ambiente é resolvido para o nome de domínio público do load balancer.

Isso é semelhante ao modo como o Elastic Beanstalk gerencia a rede quando você usa a VPC padrão. A segurança em uma sub-rede pública depende do balanceador de carga e dos grupos de segurança da instância criados pelo Elastic Beanstalk. É a configuração mais econômica, porque não requer um gateway NAT.

VPC pública/privada

AWS CloudFormation modelo — vpc-privatepublic.yaml

Configurações (balanceamento de carga)

- Visibilidade do balanceador de carga: público
- Sub-redes do balanceador de carga: as duas sub-redes públicas
- IP público da instância: desabilitado
- Sub-redes da instância: as duas sub-redes privadas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Para mais segurança, adicione sub-redes privadas à sua VPC para criar um layout público-privado. Esse layout requer um load balancer e um gateway NAT nas sub-redes públicas e permite que você execute instâncias do aplicativo, banco de dados e quaisquer outros recursos nas sub-redes privadas. Instâncias em sub-redes privadas podem se comunicar apenas com a Internet por meio do load balancer e do gateway NAT.

VPC privada

AWS CloudFormation modelo — vpc-private.yaml

Configurações (balanceamento de carga)

- Visibilidade do balanceador de carga: privado

- Sub-redes do balanceador de carga: as duas sub-redes privadas
- IP público da instância: desabilitado
- Sub-redes da instância: as duas sub-redes privadas
- Grupos de segurança de instância: adicione o grupo de segurança padrão

Para aplicativos internos que não devem ter acesso da Internet, é possível executar tudo em sub-redes privadas e configurar o load balancer para ser interno (altere Load balancer visibility (Visibilidade do load balancer) para Internal (Interno)). Esse modelo cria uma VPC sem sub-redes públicas e sem um gateway da Internet. Use esse layout para aplicativos que só devem ser acessados a partir da mesma VPC ou de uma VPN conectada.

Executar um ambiente do Elastic Beanstalk em uma VPC privada

Quando você cria o ambiente do Elastic Beanstalk em uma VPC privada, o ambiente não tem acesso à Internet. Sua aplicação pode precisar de acesso ao serviço Elastic Beanstalk ou a outros serviços. Seu ambiente pode usar relatórios de integridade avançada, e, nesse caso, as instâncias do ambiente enviam informações de integridade para o serviço de integridade avançada. E o código do Elastic Beanstalk em instâncias do ambiente envia tráfego AWS para outros serviços e outro tráfego para outros endpoints (por exemplo, para AWS baixar pacotes de dependências para seu aplicativo). Veja a seguir algumas etapas que podem ser necessárias para execução nesse caso, a fim de garantir que seu ambiente funcione corretamente.

- Configurar VPC endpoints para o Elastic Beanstalk: o Elastic Beanstalk e seu serviço de integridade aprimorada são compatíveis com VPC endpoints, que garantem que o tráfego para esses serviços permaneça dentro da rede da Amazon e não exija acesso à Internet. Para obter mais informações, consulte the section called “Endpoints da VPC”.
- Configure endpoints de VPC para serviços adicionais — As instâncias do Elastic Beanstalk enviam tráfego para vários outros serviços AWS em seu nome: Amazon Simple Storage Service (Amazon S3), Amazon Simple Queue Service (Amazon SQS) e Amazon Logs. AWS CloudFormation CloudWatch É necessário configurar VPC endpoints para esses serviços também. Para obter informações detalhadas sobre VPC endpoints, incluindo links por serviço, consulte VPC endpoints no Guia do usuário da Amazon VPC.

> **Note**
>
> Alguns AWS serviços, incluindo o Elastic Beanstalk, oferecem suporte a endpoints de VPC em um número limitado de regiões. AWS Ao projetar sua solução de VPC privada,

verifique se o Elastic Beanstalk e os outros serviços dependentes mencionados aqui oferecem suporte a endpoints de VPC na região que você escolher. AWS

- Fornecer uma imagem privada do Docker: em um ambiente Docker, o código nas instâncias do ambiente pode tentar extrair sua imagem configurada do Docker da Internet durante a criação do ambiente e falhar. Para evitar essa falha, crie uma imagem personalizada do Docker no ambiente ou use uma imagem do Docker armazenada no Amazon Elastic Container Registry (Amazon ECR) e configure um VPC endpoint para o serviço Amazon ECR.
- Habilite nomes de DNS — o código do Elastic Beanstalk em instâncias do ambiente envia tráfego AWS para todos os serviços usando seus endpoints públicos. Para garantir que esse tráfego passe, escolha a opção Enable DNS name (Habilitar nome DNS) ao configurar todos os VPC endpoints de interface. Isso adiciona uma entrada DNS em sua VPC, que mapeia o endpoint público de serviço para o VPC endpoint de interface.

  Important

  Se sua VPC não for privada e tiver acesso público à Internet e se Enable DNS name (Habilitar nome DNS) estiver desabilitado para qualquer VPC endpoint, o tráfego para o respectivo serviço será enviado pela Internet pública. Isso provavelmente não é o que você pretende. É fácil detectar esse problema com uma VPC privada, pois impede que esse tráfego passe e você receba erros. No entanto, com uma VPC voltada para o público, você não recebe nenhuma indicação.

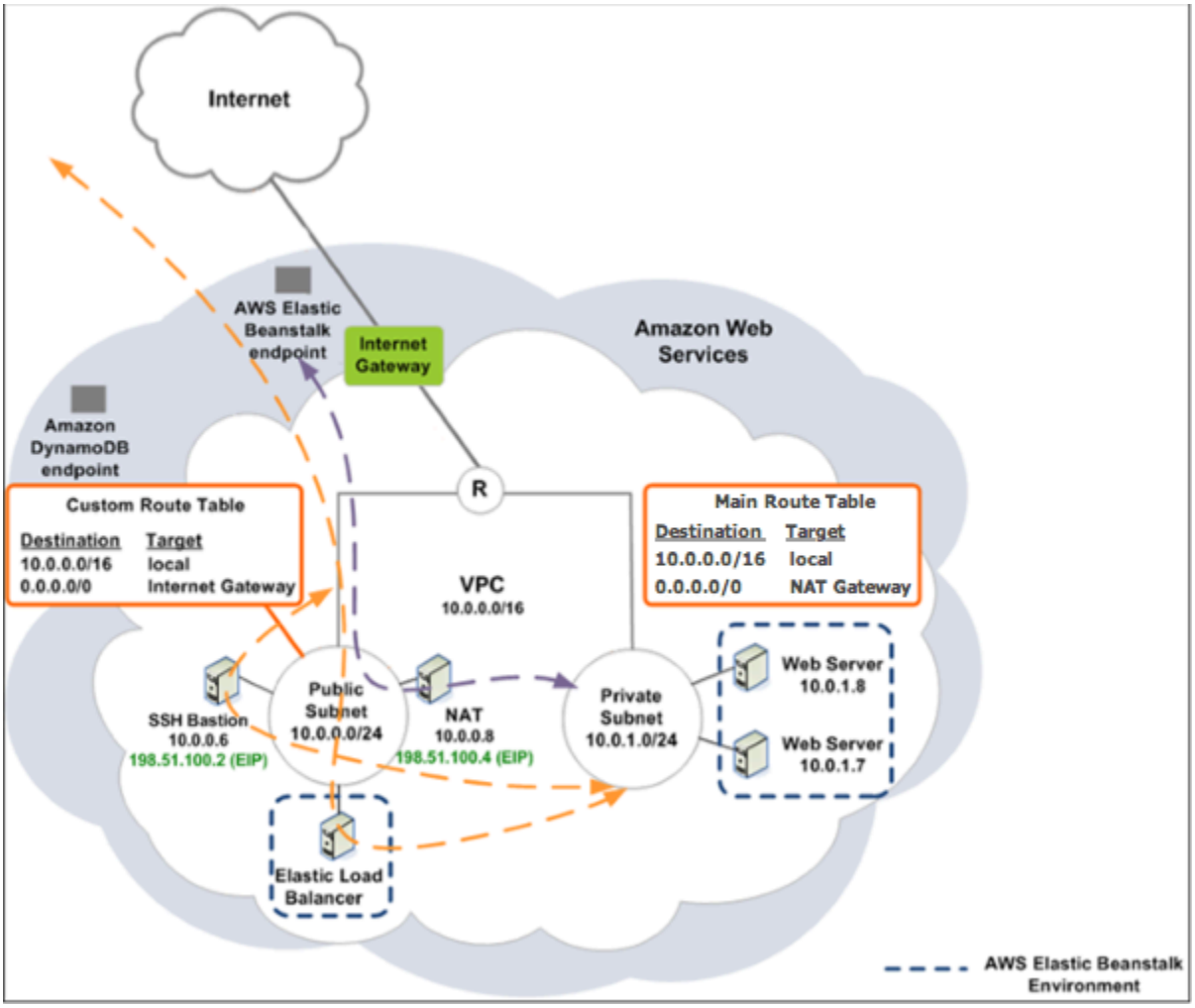
- Incluir dependências de aplicações: se sua aplicação tiver dependências como pacotes de runtime de linguagem, ele poderá tentar fazer download deles e instalá-los da Internet durante a criação do ambiente e falhar. Para evitar essa falha, inclua todos os pacotes de dependência no pacote de origem do aplicativo.
- Usar uma versão da plataforma atual: certifique-se de que seu ambiente use uma versão da plataforma lançada em 24 de fevereiro de 2020 ou posterior. Especificamente, use uma versão da plataforma que foi lançada a partir de uma destas duas atualizações: Linux Update 2020-02-28, Windows Update 2020-02-24.

> **Note**
>
> O motivo da necessidade de uma versão atualizada da plataforma é que as versões mais antigas tinham um problema que impedia que as entradas DNS criadas pela opção Enable DNS name (Habilitar nome DNS) funcionassem corretamente para o Amazon SQS.

Exemplo: Iniciar uma aplicação do Elastic Beanstalk em uma VPC com bastion hosts

Esta seção explica como implantar uma aplicação Elastic Beanstalk dentro de uma VPC usando um bastion host e por que você implementaria essa topologia.

Se suas EC2 instâncias da Amazon estiverem localizadas dentro de uma sub-rede privada, você não poderá se conectar a elas remotamente. Para se conectar às suas instâncias, configure os servidores bastion na sub-rede pública para atuarem como proxies. Por exemplo, configure encaminhadores da porta SSH ou gateways RDP na sub-rede pública para atuar como proxy do tráfego que vai para os servidores de banco de dados de sua própria rede. Esta seção fornece um exemplo de como criar uma VPC com uma sub-rede pública e privada. As instâncias estão localizadas na sub-rede privada, e o bastion host, o gateway NAT e o load balancer estão localizados na sub-rede pública. A infraestrutura terá aparência semelhante ao diagrama a seguir.



Para implantar uma aplicação do Elastic Beanstalk em uma VPC usando um bastion host, conclua as etapas descritas nas subseções a seguir.

Etapas

- Criar uma VPC com uma sub-rede privada e pública
- Criar e configurar o grupo de segurança do bastion host
- Atualizar o grupo de segurança da instância
- Criar um bastion host

Criar uma VPC com uma sub-rede privada e pública

Complete todos os procedimentos em VPC pública/privada. Ao implantar o aplicativo, você deve especificar um par de EC2 chaves da Amazon para as instâncias para que você possa se conectar

a elas remotamente. Para obter mais informações sobre como especificar o par de chaves da instância, consulte As EC2 instâncias da Amazon para seu ambiente Elastic Beanstalk.

Criar e configurar o grupo de segurança do bastion host

Crie um grupo de segurança para o Bastion Host e adicione regras que permitam tráfego SSH de entrada da Internet e tráfego SSH de saída para a sub-rede privada que contém as instâncias da Amazon. EC2

Para criar o security group do bastion host

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, escolha Grupos de segurança.
3. Escolha Criar grupo de segurança.
4. Na caixa de diálogo Create Security Group (Criar grupo de segurança), informe o seguinte e escolha Yes, Create (Sim, criar).

   Name tag (Tag do nome) (opcional)

   Informe uma tag de nome para o grupo de segurança.

   Group name

   Informe o nome do grupo de segurança.

   Descrição

   Informe uma descrição para o grupo de segurança.

   VPC

   Selecione a VPC.

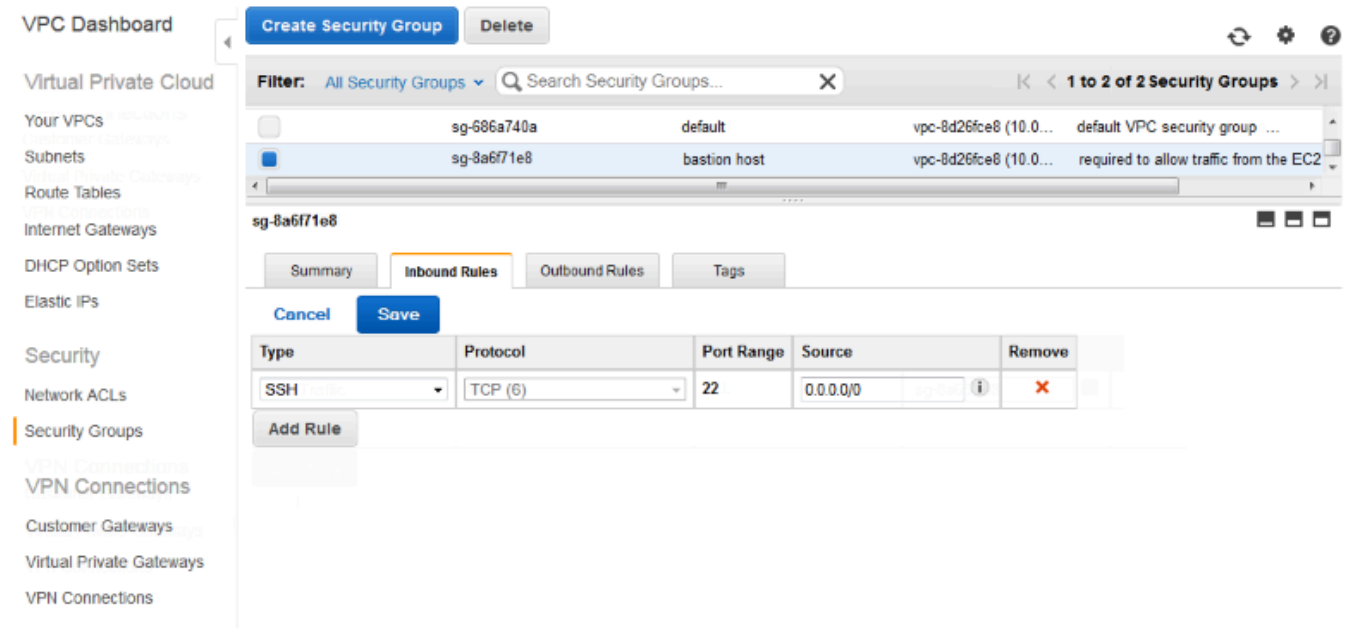
   O grupo de segurança é criado e exibido na página Security Groups (Grupos de segurança). Ele tem uma ID (por exemplo, `sg-xxxxxxxx`). Talvez você precise ativar a coluna de Group ID (ID de grupo) clicando em Show/Hide (Mostrar/ocultar) no canto superior direito da página.

Para configurar o security group do bastion host.

1. Na lista de security groups, marque a caixa de seleção do security group que acabou de criar para o bastion host.

2. Na guia Inbound Rules, escolha Edit.
3. Se necessário, escolha Add another rule.
4. Se o bastion host é uma instância do Linux, em Type, selecione SSH.

   Se o bastion host é uma instância do Windows, em Tipo, selecione RDP.
5. Insira o intervalo de CIDR de origem desejado no campo Source e escolha Save.

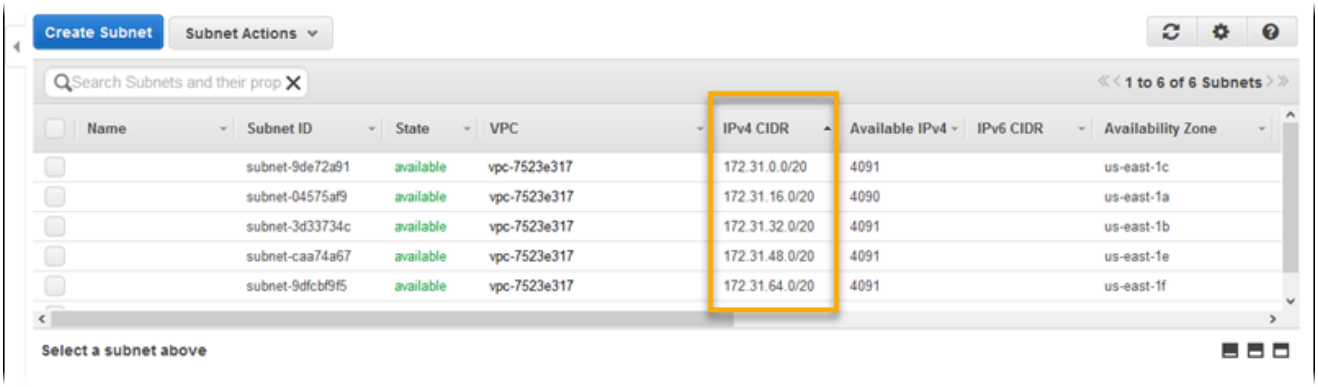


6. Na guia Inbound Rules (Regras de saída), escolha Edit (Editar).
7. Se necessário, escolha Add another rule.
8. Em Type (Tipo), selecione o tipo de dados que você especificou para a regra de entrada.
9. No campo Source (Origem), insira a variação de CIDR da sub-rede dos hosts na sub-rede privada do VPC.

   Para encontrá-lo:

   a. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
   b. No painel de navegação, escolha Sub-redes.
   c. Observe o valor em IPv4 CIDR para cada zona de disponibilidade na qual você tem hosts aos quais deseja que o bastion host faça a ponte.

> **Note**
>
> Se você tiver hosts em várias zonas de disponibilidade, crie uma regra de saída para cada uma dessas zonas de disponibilidade.



10. Escolha Salvar.

Atualizar o grupo de segurança da instância

Por padrão, o security group que você criou para suas instâncias não permite o tráfego de entrada. Embora o Elastic Beanstalk modifique o grupo padrão das instâncias para permitir o tráfego SSH, é necessário modificar o grupo de segurança da instância personalizada para permitir o tráfego RDP, caso suas instâncias sejam do Windows.

Para atualizar o security group da instância para RDP

1. Na lista de security groups, marque a caixa de seleção do security group da instância.
2. Na guia Entrada, escolha Editar.
3. Se necessário, escolha Add another rule.
4. Insira os valores a seguir e escolha Save.

   Tipo

   RDP

   Protocolo

   TCP

Intervalo de portas

`3389`

Origem

Insira o ID do security group do bastion host (por exemplo, `sg-8a6f71e8`) e escolha Save.

Criar um bastion host

Para criar um bastion host, você executa uma EC2 instância da Amazon em sua sub-rede pública que atuará como bastion host.

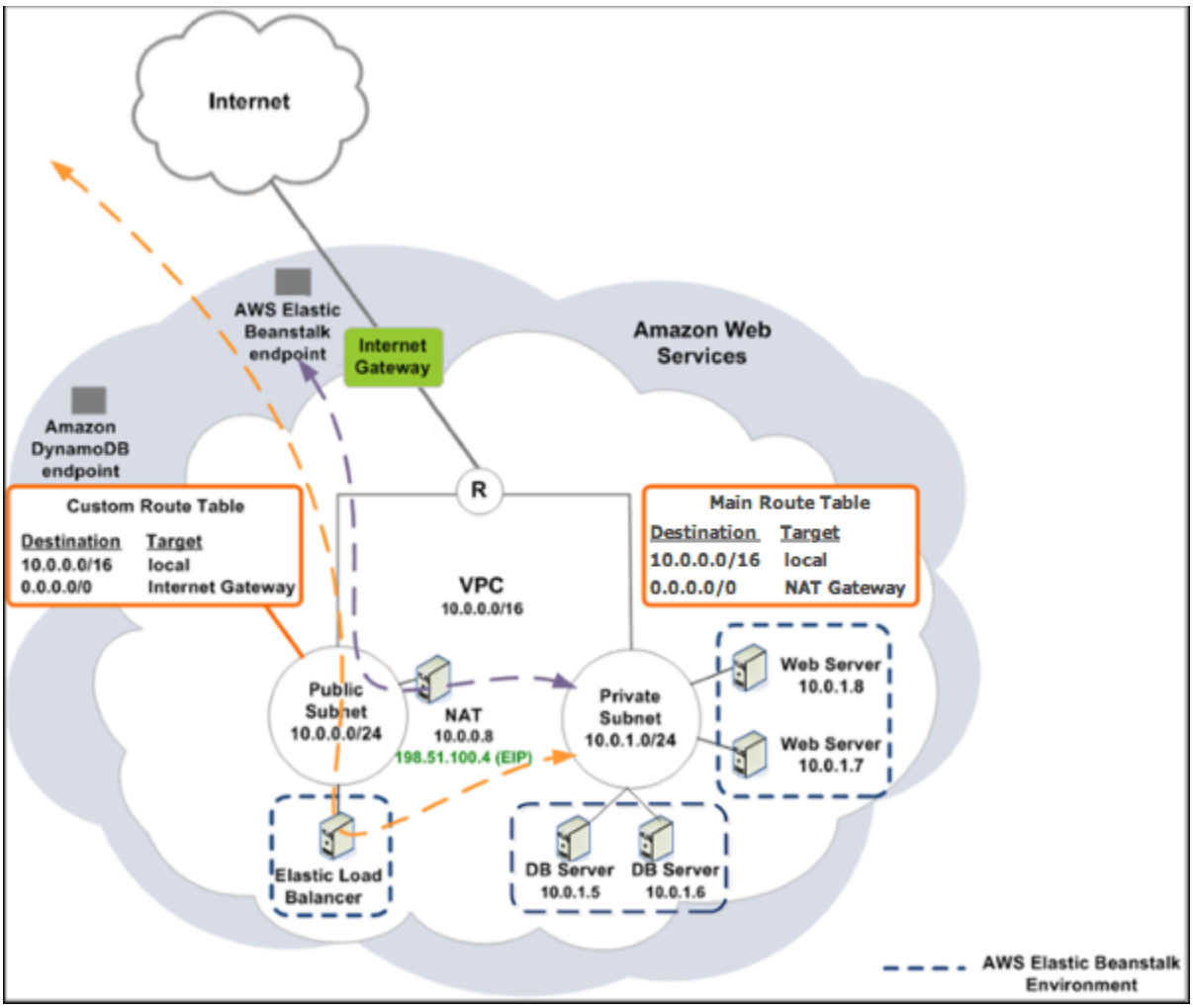
Para obter mais informações sobre como configurar um bastion host para instâncias do Windows na sub-rede privada, consulte Como controlar o acesso à rede às EC2 instâncias usando um servidor Bastion.

Para obter mais informações sobre como configurar um bastion host para instâncias Linux na sub-rede privada, consulte Conexão segura com instâncias Linux executadas em uma VPC privada da Amazon .

Exemplo: Iniciar um Elastic Beanstalk em uma VPC com o Amazon RDS

Esta seção mostra como implantar uma aplicação do Elastic Beanstalk com o Amazon RDS em uma VPC usando um gateway NAT.

A infraestrutura terá aparência semelhante ao diagrama a seguir.



> **Note**
>
> Se você não usou uma instância de banco de dados com seu aplicativo antes, tente adicionar uma a um ambiente de teste e conectar-se a uma instância de banco de dados externa antes de adicionar uma configuração de VPC à combinação.

Criar uma VPC com uma sub-rede privada e pública

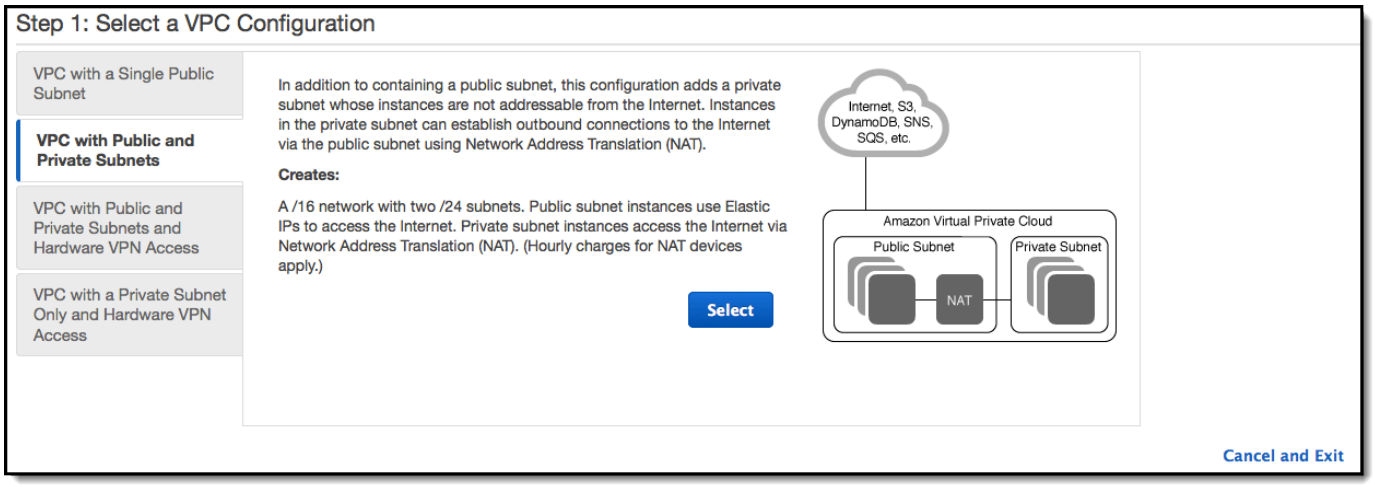
É possível usar o console da Amazon VPC para criar uma VPC.

Para criar uma VPC

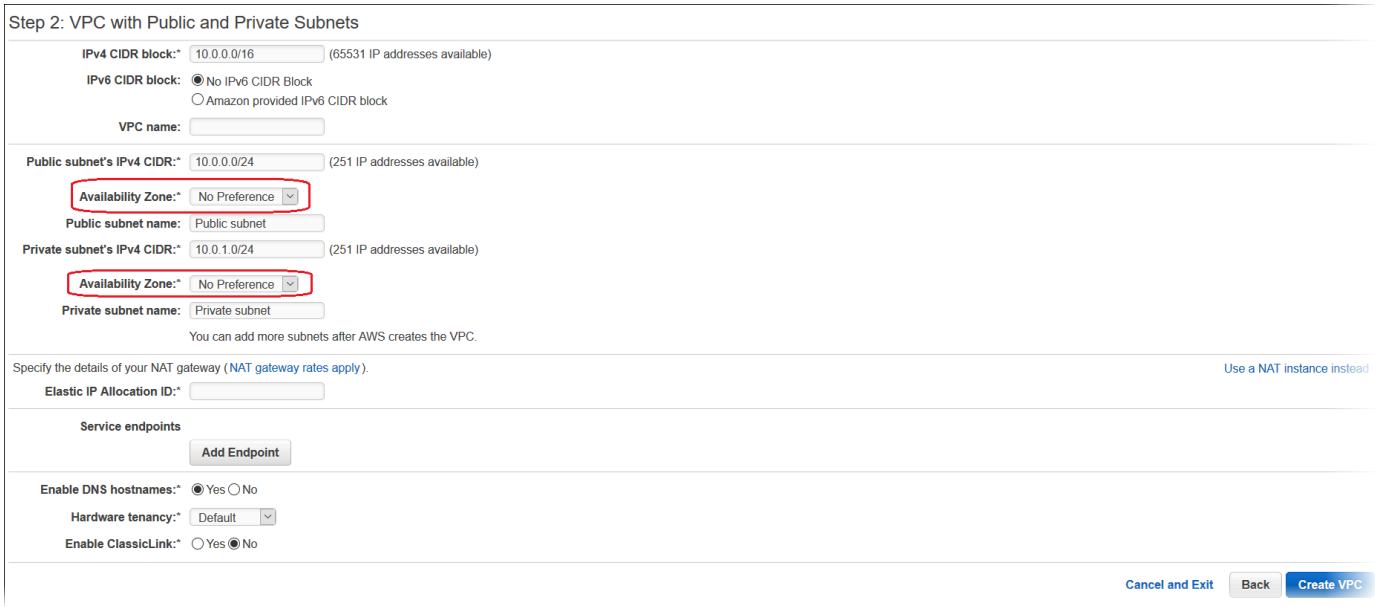
1. Faça login no console da Amazon VPC.

2. No painel de navegação, escolha VPC Dashboard (Painel da VPC). Em seguida, selecione Create VPC (Criar VPC).

3. Escolha VPC with Public and Private Subnets (VPC com sub-redes públicas e privadas) e selecione Select (Selecione).



4. Seu load balancer do Elastic Load Balancing e suas EC2 instâncias da Amazon devem estar na mesma zona de disponibilidade para que possam se comunicar entre si. Escolha mesma zona de disponibilidade em cada lista Availability Zone (Zona de disponibilidade).



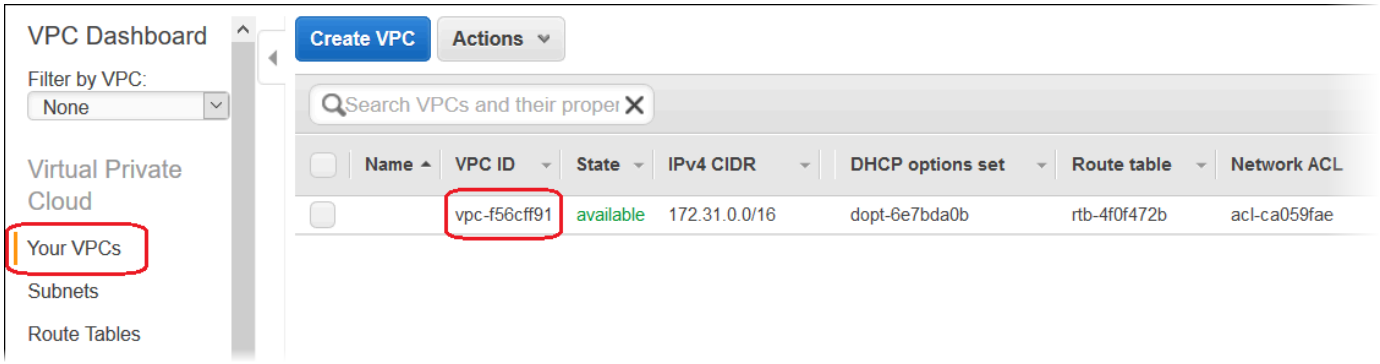
5. Escolha um endereço IP elástico no gateway NAT.

6. Escolha Criar VPC.

O assistente começa a criar a VPC, as sub-redes e o gateway da Internet. Ele também atualiza a tabela de rotas principal e cria uma tabela de rotas personalizada. Por fim, o assistente cria um gateway NAT na sub-rede pública.

> **Note**
>
> Você pode optar por ativar uma instância NAT na sub-rede pública, em vez de um gateway NAT. Para obter mais informações, consulte Cenário 2: VPC com sub-redes privadas e públicas (NAT) no Guia do usuário da Amazon VPC.

7. Depois que a VPC for criada com êxito, você receberá uma ID de VPC. Você precisa desse valor na próxima etapa. Para visualizar seu ID de VPC, escolha Seu VPCs no painel esquerdo do console da Amazon VPC.



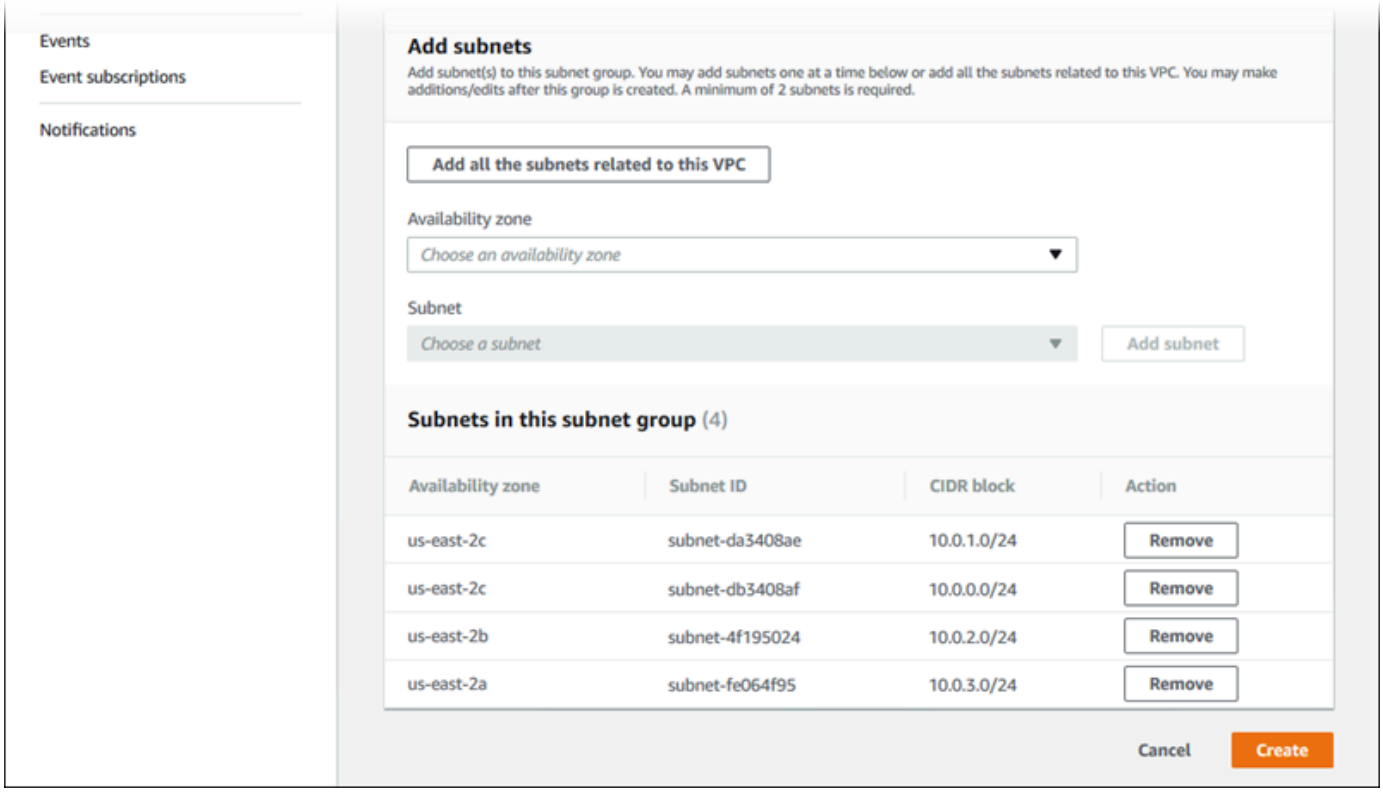
Criar um grupo de sub-redes de banco de dados

Um grupo de sub-redes de banco de dados de uma VPC é um conjunto de sub-redes (geralmente privadas) que você pode designar para suas instâncias de Banco de Dados de backend do RDS. Cada grupo de sub-redes de banco de dados deve ter pelo menos uma sub-rede para cada zona de disponibilidade em uma determinada AWS região. Para saber mais, consulte Criar uma sub-rede na VPC.

Criar um grupo de sub-redes de banco de dados

1. Abra o console do Amazon RDS.
2. No painel de navegação, escolha Subnet groups (Grupos de sub-redes).
3. Escolha Create DB Subnet Group (Criar grupo de sub-redes de banco de dados).
4. Clique em Name (Nome) e digite o nome do seu grupo de sub-redes de banco de dados.

5. Clique em Description (Descrição) e descreva o seu grupo de sub-redes de banco de dados.
6. Em VPC, selecione o ID da VPC criada por você.
7. Na seção Add subnets (Adicionar sub-redes), clique em Add all the subnets related to this VPC (Adicionar todas as sub-redes relacionadas a essa VPC).



8. Quando terminar, escolha Create (Criar imagem).

   Seu novo grupo aparece na lista de grupos de sub-redes de banco de dados do console do Amazon RDS. Clique nele para ver detalhes, por exemplo, todas as sub-redes associadas a esse grupo, no painel de detalhes, na parte inferior da página.

Implantar no Elastic Beanstalk

Depois de configurar a VPC, é possível criar o ambiente dentro dela e implantar a aplicação no Elastic Beanstalk. Você pode fazer isso usando o console do Elastic Beanstalk, ou você pode AWS usar os kits de ferramentas AWS CLI, o EB CLI ou a API do Elastic Beanstalk. Se usar o console do Elastic Beanstalk, você só precisará fazer upload do arquivo `.war` ou `.zip` e selecionar as configurações da VPC dentro do assistente. O Elastic Beanstalk acaba criando o ambiente dentro da VPC e implantando a aplicação. Como alternativa, você pode usar os AWS kits de ferramentas AWS CLI, o EB CLI ou a API do Elastic Beanstalk para implantar seu aplicativo. Para isso, você precisa

definir as configurações de opção da VPC em um arquivo de configuração e implantar esse arquivo com o pacote de origem. Este tópico apresenta instruções para ambos os métodos.

Implantar com o console do Elastic Beanstalk

O console do Elastic Beanstalk orienta durante a criação do novo ambiente dentro da VPC. Você precisa fornecer um arquivo `.war` (para aplicações Java) ou um arquivo `.zip` (para todas as outras aplicações). Na página VPC Configuration (Configuração de VPC) do assistente de ambiente do Elastic Beanstalk, você deve selecionar o seguinte:

VPC

Selecione a VPC.

Grupo de segurança de VPC

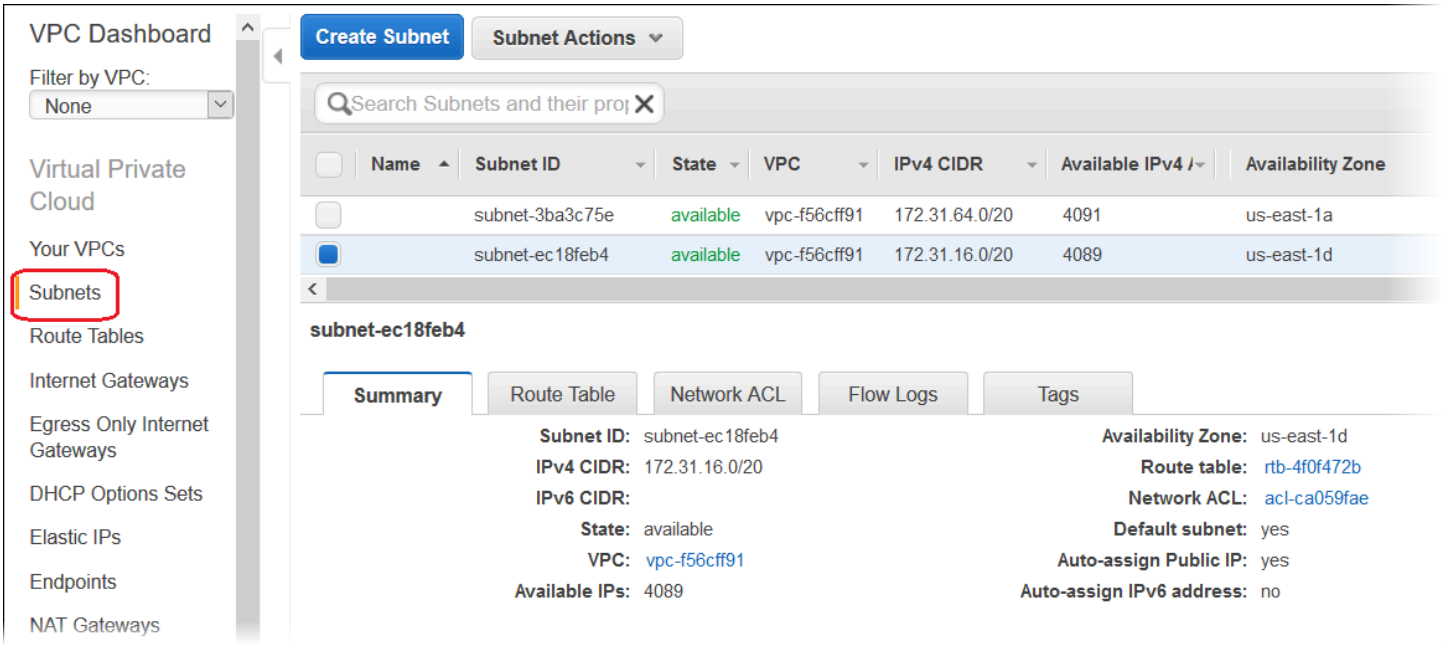
Selecione o security group da instância que você criou acima.

ELB visibility (Visibilidade do ELB)

Selecione `External` se o load balancer precisar estar disponível publicamente, ou selecione `Internal` se o load balancer precisar estar disponível apenas dentro da VPC.

Selecione as sub-redes para seu balanceador de carga e instâncias. EC2 Certifique-se de selecionar a sub-rede pública para o balanceador de carga e a sub-rede privada para suas instâncias da Amazon. EC2 Por padrão, o assistente de criação da VPC criar a sub-rede pública na sub-rede privada `10.0.0.0/24` em `10.0.1.0/24`.

Você pode visualizar sua sub-rede IDs escolhendo Sub-redes no console da Amazon VPC.



Implantação com AWS kits de ferramentas, EB CLI ou API AWS CLI

Ao implantar seu aplicativo no Elastic Beanstalk usando AWS kits de ferramentas, EB AWS CLI CLI ou API, você pode especificar suas configurações de opções de VPC em um arquivo e implantá-las com seu pacote de origem. Consulte Personalização avançada de ambiente com arquivos de configuração (`.ebextensions`) para obter mais informações.

Quando você atualiza as configurações de opção, é necessário especificar pelo menos:

- VPCId—Contém o ID da VPC.
- Subnets: contém o ID da sub-rede de grupos de Auto Scaling. Neste exemplo, é o ID da sub-rede privada.
- ELBSubnets—Contém o ID da sub-rede do balanceador de carga. Neste exemplo, é o ID da sub-rede pública.
- SecurityGroups—Contém a ID dos grupos de segurança.
- DBSubnets—Contém o ID das sub-redes do banco de dados.

> **Note**
>
> Ao usar sub-redes de banco de dados, você precisa criar sub-redes adicionais em sua VPC para cobrir todas as zonas de disponibilidade na região. AWS

Como opção, especifique também as seguintes informações:

- ELBScheme— Especifique `internal` a criação de um balanceador de carga interno dentro de sua VPC para que seu aplicativo Elastic Beanstalk não possa ser acessado de fora de sua VPC.

Veja a seguir um exemplo das configurações de opção que podem ser usadas ao implantar a aplicação do Elastic Beanstalk em uma VPC. Para obter mais informações sobre configurações de opção de VPC (inclusive exemplos de como especificá-las, valores padrão e valores válidos), consulte a tabela de namespace aws:ec2:vpc em Opções de configuração.

```
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**
>
> Ao usar sub-redes de banco de dados, certifique-se de ter sub-redes em sua VPC para cobrir todas as zonas de disponibilidade na região. AWS

Usar o Elastic Beanstalk com VPC endpoints

Este tópico explica os benefícios que um endpoint de VPC pode oferecer à sua aplicação Elastic Beanstalk. Ele também fornece instruções para criar um endpoint de VPC de interface para um serviço do Elastic Beanstalk.

Um VPC endpoint permite que você conecte sua VPC de forma privada a serviços compatíveis AWS e serviços de endpoint de VPC fornecidos por AWS PrivateLink, sem exigir um gateway de internet, dispositivo NAT, conexão VPN ou conexão. AWS Direct Connect

As instâncias na sua VPC não exigem que endereços IP públicos se comuniquem com recursos no serviço. O tráfego entre a sua VPC e os outros serviços não sai da rede da Amazon. Para obter mais informações sobre VPC endpoints, consulte VPC endpoints no Guia do usuário da Amazon VPC.

AWS Elastic Beanstalk suporta AWS PrivateLink, que fornece conectividade privada ao serviço Elastic Beanstalk e elimina a exposição do tráfego à Internet pública. Para permitir que seu aplicativo envie solicitações para o Elastic AWS PrivateLink Beanstalk usando, você configura um tipo de VPC endpoint conhecido como VPC endpoint de interface (endpoint de interface). Para obter mais informações, consulte Endpoints da VPC da interface (AWS PrivateLink) no Guia do usuário da Amazon VPC.

> **Note**
>
> O Elastic AWS PrivateLink Beanstalk oferece suporte e interface com VPC endpoints em um número limitado de. Regiões da AWS Estamos trabalhando para estender o suporte a mais Regiões da AWS em um futuro próximo.

IPv6 apoio

O Elastic Beanstalk suporta tráfego de entrada por e. IPv4 IPv6 Esta seção descreve os endpoints públicos que oferecem suporte IPV6 e também explica como configurar seus endpoints VPC do Elastic Beanstalk para suportar tráfego de pilha dupla.

Para obter mais informações gerais sobre IPv6, consulte AWS os serviços que oferecem suporte IPv6 no Guia do usuário da Amazon VPC e no AWS whitepaper IPv6 sobre. AWS

Endpoints públicos

O serviço Elastic Beanstalk tem dois conjuntos de endpoints que consistem nos endpoints mais antigos e nos endpoints mais recentes com capacidade de IPv4 pilha dupla. Ambos os conjuntos de endpoints seguem os padrões de AWS nomenclatura:

- IPv4 os endpoints usam o formato de domínio `amazonaws.com` para endpoint de serviço geral: `elasticbeanstalk.region.amazonaws.com`
- Os endpoints de pilha dupla usam o domínio `api.aws` - formato para endpoint de serviço geral: `elasticbeanstalk.region.api.aws`

Os endpoints para integridade do serviço e FIPS têm nomes de host diferentes, mas seguem o mesmo padrão de nome de domínio. Para obter uma lista de endpoints, consulte Endpoints de serviço do Elastic Beanstalk no arquivo Referência geral da Amazon Web Services.

Solicitações ao Elastic Beanstalk

Ao enviar solicitações para o serviço do Elastic Beanstalk AWS CLIcom o ou o SDK, você pode IPv4 especificar um AWS endpoint ou um endpoint de pilha dupla. O AWS SDK AWS CLI e usa os endpoints IPv4 -only por padrão se o URL do endpoint não for especificado.

O exemplo a seguir demonstra o AWS CLI envio de uma solicitação para um endpoint de pilha dupla:

Example

```
aws elasticbeanstalk list-available-solution-stacks \
    --endpoint-url "https://elasticbeanstalk.us-east-1.api.aws"
```

O exemplo a seguir demonstra o SDK do AWS Python enviando uma solicitação para um endpoint de pilha dupla:

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())
```

Endpoints VPC para pilha dupla IPs

Para configurar seus endpoints VPC do Elastic Beanstalk para suportar o tráfego de pilha dupla, especifique dualstack para o parâmetro de tipo de endereço IP do endpoint VPC. É possível especificar esse campo através da AWS CLI, do AWS SDK ou do console do AWS PrivateLink . Para obter instruções sobre como fazer isso no AWS PrivateLink console, consulte Criar um VPC endpoint no Guia.AWS PrivateLink

> **Note**
>
> Você deve especificar o tipo de endereço IP do VPC endpoint como pilha dupla ou IPv4pilha dupla. No momento, os endpoints VPC do Elastic Beanstalk não oferecem suporte a um tipo de endereço IPv6IP, o que indicaria suporte somente. IPv6 A opção dualstack permite tanto os protocolos quanto os da Internet IPv4 . IPv6

O exemplo a seguir demonstra como criar um endpoint VPC de pilha dupla com o AWS CLI:

Example

```
aws ec2 create-vpc-endpoint \
    --vpc-id "vpc-example"
    --service-name "com.amazonaws.us-east-1.elasticbeanstalk"
    --ip-address-type "dualstack"
```

Configurar um VPC endpoint para o Elastic Beanstalk

Para criar o VPC endpoint de interface para o serviço do Elastic Beanstalk na VPC, siga o procedimento Criar um endpoint de interface.

- Em Nome do serviço, escolha com.amazonaws. ***region***. pé de feijão elástico.
- Para o tipo de endereço IP, escolha um IPv4ou o Dualstack. No momento, os endpoints VPC do Elastic Beanstalk não oferecem suporte a um tipo de endereço IPv6IP, o que indicaria suporte somente. IPv6 A opção Dualstack permite tanto os protocolos quanto os da Internet IPv4 . IPv6

Se a VPC estiver configurada com acesso público à Internet, a aplicação ainda poderá acessar o Elastic Beanstalk pela Internet usando o endpoint público `elasticbeanstalk.region.amazonaws.com` ou `elasticbeanstalk.region.api.aws`. É possível impedir isso habilitando Enable DNS name (Habilitar nome DNS) durante a criação do endpoint (true por padrão). Isso adiciona uma entrada DNS em sua VPC, que mapeia o endpoint público de serviço para o VPC endpoint de interface.

Configurar um VPC endpoint para integridade avançada

Se você habilitou os relatórios de integridade aprimorados para o seu ambiente, você pode configurar as informações de saúde aprimoradas para serem enviadas AWS PrivateLink também. As informações de integridade aprimorada são enviadas pelo daemon `healthd`, um componente do Elastic Beanstalk nas instâncias do seu ambiente, para um serviço de integridade aprimorada separado do Elastic Beanstalk. Para criar um VPC endpoint de interface para esse serviço na VPC, siga o procedimento Criar um endpoint de interface.

- Em Nome do serviço, escolha com.amazonaws. ***region***. pedúnculo de feijão elástico - saúde.
- Para o tipo de endereço IP, escolha um IPv4ou o Dualstack. No momento, os endpoints VPC do Elastic Beanstalk não oferecem suporte a um tipo de endereço IPv6IP, o que indicaria suporte somente. IPv6 A opção Dualstack permite tanto os protocolos quanto os da Internet IPv4 . IPv6

> **Important**
>
> O daemon `healthd` envia informações de integridade aprimoradas ao endpoint público `elasticbeanstalk-health.region.amazonaws.com` ou `elasticbeanstalk-health.region.api.aws`. Se a VPC estiver configurada com acesso público à Internet e Enable DNS name (Habilitar nome DNS) estiver desabilitado para o VPC endpoint, as informações de integridade avançada serão enviadas pela Internet pública. Isso provavelmente não é sua intenção ao configurar um VPC endpoint de integridade avançada. Certifique-se de que Enable DNS name (Habilitar nome DNS) esteja habilitado (true por padrão).

Usar VPC endpoints em uma VPC privada

Uma VPC privada ou uma sub-rede privada em uma VPC não tem acesso público à Internet. Convém executar o ambiente do Elastic Beanstalk em uma VPC privada e configurar VPC endpoints

de interface para segurança aprimorada. Nesse caso, lembre-se de que o ambiente pode tentar se conectar à Internet por outros motivos, além de entrar em contato com o serviço do Elastic Beanstalk. Para saber mais sobre como executar um ambiente em uma VPC privada, consulte the section called “Executar um ambiente do Elastic Beanstalk em uma VPC privada”.

Usar políticas de endpoint para controlar o acesso com VPC endpoints

Este tópico explica como anexar uma política a endpoints de VPC para controlar o acesso à sua aplicação (serviço) e ao ambiente Elastic Beanstalk.

Uma política de endpoint é uma política de recursos AWS Identity and Access Management (IAM) que controla o acesso do endpoint ao serviço especificado. A política de endpoint é específica ao endpoint. Ela é separada de outras políticas do IAM de usuário ou instância que seu ambiente possa ter e não as substitui.

Por padrão, um VPC endpoint permite acesso total ao serviço ao qual está associado. Ao criar ou modificar um endpoint, você pode anexar uma política de endpoint para controlar o acesso a recursos específicos associados ao serviço. Para obter detalhes sobre a criação e o uso de políticas de endpoint de VPC, consulte Controle o acesso aos endpoints da VPC usando políticas de endpoint no Guia do AWS PrivateLink .

> **Note**
>
> Ao criar políticas de endpoint restritivas, talvez você precise adicionar permissões específicas aos recursos necessários para que o acesso a esses recursos não seja bloqueado pela política de endpoint. Isso garante que seu ambiente continue a ser implementado e funcione adequadamente.

O exemplo a seguir nega a todos os usuários a permissão para encerrar um ambiente por meio do VPC endpoint e permite acesso total a todas as outras ações.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
```

```
        {
            "Action": "elasticbeanstalk:TerminateEnvironment",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Permissões de bucket do Amazon S3 necessárias para políticas restritivas de endpoint de VPC

Se você adicionar restrições às suas políticas de endpoint de VPC, deverá incluir permissões específicas de bucket do Amazon S3 para garantir que o ambiente continue a ser implantado e funcione adequadamente. Esta seção explica os buckets S3 necessários e inclui exemplos de políticas.

Tópicos

- Buckets do S3 que armazenam ativos para gerenciar plataformas de ambiente
- S3 Buckets de propriedade da AWS CloudFormation
- Buckets do S3 pertencentes a contas de clientes para armazenar código-fonte e outros itens
- Buckets do S3 pertencentes a contas de clientes para oferecer suporte à autenticação de registro do Docker
- Atualizar sua política de endpoint de VPC

Buckets do S3 que armazenam ativos para gerenciar plataformas de ambiente

O serviço Elastic Beanstalk possui buckets do S3 que armazenam os ativos associados a uma pilha de soluções (versão da plataforma). Esses ativos incluem arquivos de configuração, a aplicação de exemplo e os tipos de instância disponíveis. Quando o Elastic Beanstalk cria e gerencia seu ambiente, ele recupera as informações necessárias para a versão específica da plataforma do bucket de ativos para cada correspondente. Região da AWS

ARN do bucket do S3

`arn:aws:s3:::elasticbeanstalk-samples-`*`region`*

Amazon Linux 2 e posterior

- `arn:aws:s3:::elasticbeanstalk-platform-assets-region`

> **Note**
>
> O nome do bucket segue uma convenção diferente para a região BJS. A string public-beta-cn-north-1 é usada no lugar de*region*. Por exemplo, .`arn:aws:s3:::elasticbeanstalk-platform-assets-public-beta-cn-north-1`

Windows Server, Amazon Linux (AMI), Amazon Linux 2 e versões posteriores

- `arn:aws:s3:::elasticbeanstalk-env-resources-region`
- `arn:aws:s3:::elasticbeanstalk-region`

> **Note**
>
> Os nomes de buckets para buckets platform-assets e env-resources seguem convenções diferentes em algumas regiões. Consulte a seção de padrões de nomenclatura de bucket específicos da região abaixo para obter detalhes.

Padrões de ARN de bucket específicos da região

Região	Tipo de balde	Padrão ARN do bucket
me-central-1	ativos da plataforma	`arn:aws:s3:::elasticbeanstalk-platform-assets-me-central-1-f08b818c`
	recursos ambientais	`arn:aws:s3:::elasticbeanstalk-env-resources-me-central-1-f08b818c`

Operações

GetObject

Exemplo de política de endpoint de VPC

O exemplo a seguir ilustra como fornecer acesso aos buckets do S3 necessários para operações do Elastic Beanstalk na região Leste dos EUA (Ohio) (us-east-2). O exemplo lista todos os buckets das plataformas Amazon Linux e Windows Server. Atualize sua política para incluir apenas os buckets que se aplicam ao sistema operacional do ambiente.

Exemplo de política

> **Important**
>
> Recomendamos que você evite usar caracteres curinga (*) no lugar das regiões específicas nessa política. Por exemplo, use `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` e não use `arn:aws:s3:::cloudformation-waitcondition-*/*`. O uso de curingas pode fornecer acesso a buckets do S3 aos quais você não pretende conceder acesso. Se você quiser usar a política para mais de uma região, recomendamos repetir o primeiro bloco `Statement` para cada região.

```
{
    "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 de propriedade da AWS CloudFormation

O Elastic AWS CloudFormation Beanstalk usa para criar recursos para seu ambiente. CloudFormation possui buckets S3 em cada um Região da AWS para monitorar as respostas às condições de espera.

Serviços como o Elastic Beanstalk CloudFormation se comunicam enviando solicitações para uma URL pré-assinada do Amazon S3 para o bucket do S3 que possui. CloudFormation CloudFormation cria a URL pré-assinada do Amazon S3 usando `cloudformation.amazonaws.com` o principal do serviço.

Para obter informações mais detalhadas, consulte Considerações sobre CloudFormation VPC endpoints no AWS CloudFormation Guia do usuário. Para saber mais sobre pré-assinados URLs, consulte Como trabalhar com pré-assinados URLs no Guia do usuário do Amazon S3.

ARN do bucket do S3

- `arn:aws:s3:::cloudformation-waitcondition-`*`region`*

  Ao usar condições de espera, os nomes de região contêm traços. Por exemplo, Oeste-EUA-2.
- `arn:aws:s3:::cloudformation-custom-resource-response-`*`region`*

  Ao usar recursos personalizados, os nomes de região não contêm traços. Por exemplo: uswest2.

Operações

GetObject

Exemplo de política de endpoint de VPC

O exemplo a seguir ilustra como fornecer acesso aos buckets do S3 necessários para operações do Elastic Beanstalk na região Leste dos EUA (Ohio) (us-east-2).

Exemplo de política

> **Important**
>
> Recomendamos que você evite usar caracteres curinga (*) no lugar das regiões específicas nessa política. Por exemplo, use `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` e não use `arn:aws:s3:::cloudformation-waitcondition-*/*`.

O uso de curingas pode fornecer acesso a buckets do S3 aos quais você não pretende conceder acesso. Se você quiser usar a política para mais de uma região, recomendamos repetir o primeiro bloco `Statement` para cada região.

```
{
    "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/*"
            ]
         }
    ]
}
```

Buckets do S3 pertencentes a contas de clientes para armazenar código-fonte e outros itens

Esse bucket é de propriedade da conta do AWS cliente que é proprietária do ambiente. Ele armazena recursos específicos do seu ambiente, como código-fonte e logs solicitados.

ARN do bucket do S3

`arn:aws:s3:::elasticbeanstalk-region-account-id`

Operações

- GetObject
- GetObjectAcl
- PutObject
- PutObjectAcl
- ListBucket

Exemplo de política de endpoint de VPC

O exemplo a seguir ilustra como fornecer acesso aos buckets do S3 necessários para as operações do Elastic Beanstalk na região Leste dos EUA (Ohio) (us-east-2) e para o exemplo de ID de conta 123456789012. AWS

Exemplo de política

> **Important**
>
> Recomendamos que você evite usar caracteres curinga (*) no lugar das regiões específicas nessa política. Por exemplo, use `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` e não use `arn:aws:s3:::cloudformation-waitcondition-*/*`. O uso de curingas pode fornecer acesso a buckets do S3 aos quais você não pretende conceder acesso. Se você quiser usar a política para mais de uma região, recomendamos repetir o primeiro bloco `Statement` para cada região.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToCustomerItems",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["GetObject",
                       "GetObjectAcl",
                       "PutObject",
                       "PutObjectAcl",
                       "ListBucket"
                       ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/*"
            ]
         }
    ]
}
```

Buckets do S3 pertencentes a contas de clientes para oferecer suporte à autenticação de registro do Docker

Esse bucket só se aplica a ambientes baseados na plataforma Docker. O bucket armazena um arquivo usado para autenticação em um registro privado do Docker que reside em um bucket do S3 provisionado pelo cliente. Para obter mais informações, consulte Usando o arquivo `Dockerrun.aws.json` v3 no capítulo sobre a plataforma Docker deste guia.

ARN do bucket do S3

O ARN varia de acordo com a conta do cliente.

O ARN do bucket do S3 tem o seguinte formato: `arn:aws:s3:::bucket-name`

Operações

GetObject

Exemplo de política de endpoint de VPC

O exemplo a seguir ilustra como fornecer acesso a um bucket do S3 com o nome amzn-s3-demo-bucket1.

Exemplo de política

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRequestsToDockerRegistryAuth",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["GetObject"],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
         }
    ]
}
```

Atualizar sua política de endpoint de VPC

Como um endpoint de VPC tem apenas uma política anexada, você deve combinar todas as permissões em uma única política. O exemplo a seguir fornece todos os exemplos anteriores combinados em um só.

Para obter detalhes sobre a criação e o uso de políticas de endpoint de VPC, consulte Controle o acesso aos endpoints da VPC usando políticas de endpoint no Guia do AWS PrivateLink .

Assim como os exemplos anteriores, o exemplo a seguir ilustra como fornecer acesso aos buckets do S3 necessários para as operações do Elastic Beanstalk na região leste dos EUA (Ohio) (us-east-2). Também inclui buckets com exemplo de ID de AWS conta 123456789012 e exemplo de nome de bucket amzn-s3-demo-bucket1.

> **Important**
>
> Recomendamos que você evite usar caracteres curinga (*) no lugar das regiões específicas nessa política. Por exemplo, use `arn:aws:s3:::cloudformation-waitcondition-us-east-2/*` e não use `arn:aws:s3:::cloudformation-waitcondition-*/*`. O uso de curingas pode fornecer acesso a buckets do S3 aos quais você não pretende conceder acesso. Se você quiser usar a política para mais de uma região, recomendamos repetir o primeiro bloco `Statement` para cada região.

```
{
    "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/*"
            ]
         },
        {
            "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/*"
            ]
         },
         {
            "Sid": "AllowRequestsToCustomerItems",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["GetObject",
                       "GetObjectAcl",
                       "PutObject",
                       "PutObjectAcl",
                       "ListBucket"
                       ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-us-east-2-123456789012/*"
            ]
         },
        {
            "Sid": "AllowRequestsToDockerRegistryAuth",
            "Effect": "Allow",
            "Principal": {"AWS": "*"},
            "Action": ["GetObject"],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1""
            ]
         }
    ]
}
```

AWS Elastic Beanstalk segurança

Use este capítulo para saber mais sobre as tarefas de segurança pelas quais o Elastic Beanstalk é responsável, juntamente com as configurações de segurança que você deve considerar ao usar o Elastic Beanstalk para atender aos seus objetivos de segurança e conformidade.

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem.

Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa todos os serviços oferecidos na AWS nuvem e fornecer serviços que você possa usar com segurança. Nossa responsabilidade de segurança é a maior prioridade em AWS, e a eficácia de nossa segurança é regularmente testada e verificada por auditores terceirizados como parte dos Programas de AWS Conformidade. Analise os AWS serviços no escopo dos programas de AWS garantia para obter informações relacionadas ao Elastic Beanstalk.

Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você está usando e por outros fatores, incluindo a sensibilidade de seus dados, os requisitos da sua organização e as leis e regulamentos aplicáveis. Esta documentação se destina a ajudar você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Elastic Beanstalk.

Tópicos

- Proteção de dados no Elastic Beanstalk
- Gerenciamento de identidade e acesso para o Elastic Beanstalk
- Registrar em log e monitorar no Elastic Beanstalk
- Validação de conformidade para o Elastic Beanstalk
- Resiliência no Elastic Beanstalk
- Segurança da infraestrutura no Elastic Beanstalk
- Análise de configuração e vulnerabilidade no Elastic Beanstalk
- Melhores práticas de segurança para Elastic Beanstalk

Proteção de dados no Elastic Beanstalk

O modelo de responsabilidade AWS compartilhada modelo se aplica à proteção de dados em AWS Elastic Beanstalk. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Data Privacy FAQ. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais da e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use as soluções de AWS criptografia, juntamente com todos os controles de segurança padrão em Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sigilosas, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Elastic Beanstalk Serviços da AWS ou outro usando o console, a API ou. AWS CLI AWS SDKs Quaisquer dados inseridos em tags ou em campos de texto de formato

livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Tópicos

- Proteção de dados usando criptografia
- Privacidade do tráfego entre redes

Proteção de dados usando criptografia

É possível usar diferentes formas de criptografia de dados para proteger os dados do Elastic Beanstalk. Proteção de dados se refere aos dados em trânsito (à medida que são transferidos para e do Elastic Beanstalk) e em repouso (enquanto estão armazenados em datacenters da). AWS

Criptografia em trânsito

É possível obter proteção de dados em trânsito de duas maneiras: criptografar a conexão usando Secure Sockets Layer (SSL) ou usar a criptografia do lado do cliente (onde o objeto é criptografado antes de ser enviado). Ambos os métodos são válidos para proteger os dados do aplicativo. Para proteger a conexão, criptografe-a usando SSL sempre que o aplicativo e seus desenvolvedores, administradores e usuários finais enviarem ou receberem objetos. Para obter detalhes sobre como criptografar o tráfego da web de e para o aplicativo, consulte the section called “HTTPS”.

A criptografia do lado do cliente não é um método válido para proteger seu código-fonte em versões de aplicações e pacotes de origem carregados. O Elastic Beanstalk precisa de acesso a esses objetos, para que eles não possam ser criptografados. Assim, certifique-se de proteger a conexão entre o ambiente de desenvolvimento ou de implantação e o Elastic Beanstalk.

Criptografia em repouso

Para proteger os dados em repouso do aplicativo, saiba mais sobre a proteção de dados no serviço de armazenamento usado pelo aplicativo. Por exemplo, consulte Data Protection in Amazon RDS (Proteção de dados no Amazon RDS) no Amazon RDS User Guide (Guia do usuário do Amazon RDS), Proteção de dados no Amazon S3 no Guia do usuário do Amazon Simple Storage Service, ou Criptografia de dados e metadados no EFS no Guia do usuário do Amazon Elastic File System.

O Elastic Beanstalk armazena vários objetos em um bucket criptografado do Amazon Simple Storage Service (Amazon S3) criado por ele para cada região da em que AWS você cria ambientes. Como o Elastic Beanstalk retém a criptografia padrão fornecida pelo Amazon S3, ele cria buckets criptografados do Amazon S3. Para obter detalhes, consulte the section called “Amazon S3”. Forneça alguns dos objetos armazenados e envie-os para o Elastic Beanstalk, por exemplo, versões de aplicações e pacotes de origem. O Elastic Beanstalk gera outros objetos, por exemplo, arquivos de log. Além dos dados que o Elastic Beanstalk armazena, a aplicação pode transferir e/ou armazenar dados como parte de sua operação.

Para proteger os dados armazenados em volumes do Amazon Elastic Block Store (Amazon EBS) anexados a instâncias do seu ambiente, habilite a criptografia do Amazon EBS por padrão em sua AWS conta e região. Quando ativados, todos os novos volumes do Amazon EBS e seus snapshots são automaticamente criptografados usando AWS Key Management Service chaves. Para obter mais informações, consulte Criptografia por padrão no Guia do usuário do Amazon EBS.

Consulte mais informações sobre proteção de dados na publicação do blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS .

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Privacidade do tráfego entre redes

É possível usar o Amazon Virtual Private Cloud (Amazon VPC) para criar limites entre recursos em sua aplicação do Elastic Beanstalk e controlar o tráfego entre eles, sua rede on-premises e a internet. Para obter mais detalhes, consulte the section called “Amazon VPC”.

Para obter mais informações sobre segurança da Amazon VPC, consulte Segurança no Guia do usuário da Amazon VPC.

Consulte mais informações sobre proteção de dados na publicação do blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS .

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Gerenciamento de identidade e acesso para o Elastic Beanstalk

AWS Identity and Access Management (IAM) é um AWS serviço que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem

pode ser autenticado (fazer login) e autorizado (ter permissões) a usar os recursos do AWS Elastic Beanstalk . O IAM é um AWS serviço que você pode usar sem custo adicional.

Para obter detalhes sobre como trabalhar com o IAM, consulte Uso do Elastic Beanstalk com o AWS Identity and Access Management.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

AWS políticas gerenciadas para AWS Elastic Beanstalk

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso de todos os AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas operações de API são disponibilizadas para serviços existentes.

Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

Atualizações do Elastic AWS Beanstalk nas políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Elastic Beanstalk desde 1º de março de 2021.

Para ver a origem JSON de uma política gerenciada específica, consulte o Guia de referência de políticas gerenciadas do AWS .

Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk realize atualizações gerenciadas quando estiver habilitado. Propagação de tags para modelos de execução Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	27 de fevereiro de 2025
AdministratorAccess-AWSElastic Beanstalk — Política existente atualizada	Essa política foi atualizada para substituir o StringLikeoperador pelo ArnLikeoperador para avaliar as chaves do tipo ARN no bloco de condições. `iam:PolicyArn` Isso proporciona uma fiscalização mais segura. Para obter mais informações, consulte Gerenciar políticas de usuário do Elastic Beanstalk.	11 de dezembro de 2024
As seguintes políticas receberam atualizações: • AWSElasticBeanstalkInternalMaintenanceRolePolicy • AWSElasticBeanstalkMaintenance	Essas políticas foram atualizadas para permitir que o Elastic Beanstalk adicione ou remova tags ao criar ou atualizar uma pilha ou um conjunto de AWS CloudFormation alterações.	30 de abril de 2024

Alteração	Descrição	Data
• AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy • AWSElasticBeanstalkManagedUpdatesServiceRolePolicy • AWSElasticBeanstalkRoleCore	Para obter mais informações sobre o `AWSElasticBeanstalkManagedUpdatesServiceRolePolicy`, consulte Permissões de função vinculada ao serviço para o Elastic Beanstalk. Para obter mais informações sobre o `AWSElasticBeanstalkRoleCore`, consulte Políticas de integração com outros serviços.	

Alteração	Descrição	Data
AWSElasticBeanstalkService— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk marque recursos para o Elastic Load Balancing, os grupos do Auto Scaling (ASG) e o Amazon ECS ao criá-los. **Note** Essa política foi substituída anteriormente pela `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`. Embora essa política não esteja mais disponível para anexação a novos usuários, grupos ou perfis do IAM, ela ainda pode estar anexada aos que existiam antes. Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	10 de maio de 2023

Alteração	Descrição	Data
AWSElasticBeanstalkMulticontainerDocker— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk marque os recursos para o Amazon ECS ao criá-los. Para obter mais informações, consulte Gerenciar perfis de instância do Elastic Beanstalk.	23 de março de 2023
AWSElasticBeanstalkRoleECS — Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk marque os recursos para o Amazon ECS ao criá-los. Para obter mais informações, consulte Políticas de integração com outros serviços.	23 de março de 2023
AdministratorAccess-AWSElastic Beanstalk — Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk marque os recursos para o Amazon ECS ao criá-los. Para obter mais informações, consulte Gerenciar políticas de usuário do Elastic Beanstalk.	23 de março de 2023

Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy — Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags aos recursos do Amazon ECS ao criá-los. Para obter mais informações, consulte Permissões de função vinculada ao serviço para o Elastic Beanstalk.	23 de março de 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags aos recursos do Amazon ECS ao criá-los. Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	23 de março de 2023
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Política existente atualizada	Esta política foi atualizada para permitir que o Elastic Beanstalk adicione tags a grupos do Auto Scaling ao criá-los. Para obter mais informações, consulte A função vinculada ao serviço de atualizações gerenciadas.	27 de janeiro de 2023

Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags na criação de um grupo do Auto Scaling (ASG). Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	23 de janeiro de 2023
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk adicione tags na criação de um Elastic Load Balancing (ELB). Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	21 de dezembro de 2022

Alteração	Descrição	Data
AWSElasticBeanstalkManagedUpdatesServiceRolePolicy— Política existente atualizada	As permissões foram adicionadas a esta política para permitir que o Elastic Beanstalk faça o seguinte durante as atualizações gerenciadas: • Crie e exclua modelos de lançamento e versões de modelos. • Inicie EC2 instâncias da Amazon com modelos de lançamento. • Se um Amazon RDS estiver presente, recupere uma lista dos mecanismos de banco de dados disponíveis e informações sobre instâncias do RDS provisionadas. Para obter mais informações, consulte A função vinculada ao serviço de atualizações gerenciadas.	23 de agosto de 2022

Alteração	Descrição	Data
AWSElasticBeanstalkReadOnlyAccess— Obsoleto GovCloud (EUA) Região da AWS	Esta política foi substituída por `AWSElasticBeanstalkReadOnly` . Essa política será eliminada gradualmente nos GovCloud (EUA) Região da AWS. Quando essa política for eliminada, ela não estará mais disponível para anexar a novos usuários, grupos ou funções do IAM após 17 de junho de 2021. Para obter mais informações, consulte as Políticas do usuário.	17 de junho de 2021
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy— Política existente atualizada	Essa política foi atualizada para permitir que o Elastic Beanstalk leia atributos para zonas de disponibilidade. EC2 Ele habilita que o Elastic Beanstalk forneça uma validação mais eficaz da seleção do tipo de instância em Zonas de disponibilidade. Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	16 de junho de 2021

Alteração	Descrição	Data
AWSElasticBeanstalkFullAccess— Obsoleto GovCloud (EUA) Região da AWS	Esta política foi substituída por `AdministratorAccess-AWSElasticBeanstalk` . Essa política será eliminada gradualmente nos GovCloud (EUA) Região da AWS. Quando essa política for eliminada, ela não estará mais disponível para anexação a novos usuários, grupos ou funções do IAM após 10 de junho de 2021. Para obter mais informaçõ es, consulte as Políticas do usuário.	10 de junho de 2021

Alteração	Descrição	Data
As seguintes políticas gerenciadas foram preteridas em todos os países da China: Região da AWS • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess	A política `AWSElasticBeanstalkFullAccess` foi substituída por `AdministratorAccess-AWSElasticBeanstalk` . A política `AWSElasticBeanstalkReadOnlyAccess` foi substituída por `AWSElasticBeanstalkReadOnly` . Essas políticas foram eliminadas gradualmente em todos os países da China Região da AWS. Essas políticas não estarão mais disponíveis para anexar a novos usuários, grupos ou funções do IAM após 3 de junho de 2021. Para obter mais informações, consulte as Políticas do usuário.	3 de junho de 2021

Alteração	Descrição	Data
AWSElasticBeanstalkService— Obsoleto	Essa política foi substituída pela `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` . Essa política está sendo descontinuada e não está mais disponível a novos usuários, grupos ou funções do IAM para anexação. Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	Junho de 2021 a janeiro de 2022

Alteração	Descrição	Data
As seguintes políticas gerenciadas foram preteridas em todos os países, Região da AWS exceto na China e GovCloud (EUA): • AWSElasticBeanstalkFullAccess • AWSElasticBeanstalkReadOnlyAccess	A política `AWSElasticBeanstalkFullAccess` foi substituída por `AdministratorAccess-AWSElasticBeanstalk`. A política `AWSElasticBeanstalkReadOnlyAccess` foi substituída por `AWSElasticBeanstalkReadOnly`. Essas políticas foram eliminadas gradualmente em todos os Região da AWS anos, exceto na China e GovCloud (EUA). Essas políticas não estarão mais disponíveis para anexar a novos usuários, grupos ou funções do IAM após 16 de abril de 2021. Para obter mais informações, consulte as Políticas do usuário.	16 de abril de 2021

Alteração	Descrição	Data
As seguintes políticas gerenciadas foram atualizadas: • AdministratorAccess- Pé AWSElastic de feijão • AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	Ambas as políticas agora oferecem suporte a PassRole permissões na China Regiões da AWS. Para obter mais informações sobre o `AdministratorAccess-AWSElasticBeanstalk` , consulte as Políticas do usuário. Para obter mais informações sobre `AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` , consulte as Políticas gerenciadas de função de serviços.	9 de março de 2021
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy – Nova política	O Elastic Beanstalk adicionou uma nova política para substituir a política gerenciada `AWSElasticBeanstalkService` . Essa nova política gerenciada melhora a segurança de seus recursos aplicando um conjunto mais restritivo de permissões. Para obter mais informações, consulte as Políticas gerenciadas de função de serviços.	3 de março de 2021

Alteração	Descrição	Data
O Elastic Beanstalk começou a monitorar as alterações	O Elastic Beanstalk começou a monitorar as mudanças nas políticas gerenciadas. AWS	1º de março de 2021

Registrar em log e monitorar no Elastic Beanstalk

AWS fornece várias ferramentas para monitorar seus recursos do Elastic Beanstalk e responder a possíveis incidentes. O monitoramento é importante para manter a confiabilidade, a disponibilidade e o desempenho de AWS Elastic Beanstalk suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra.

Para obter mais informações sobre o monitoramento, consulte Ambientes de monitoramento no Elastic Beanstalk.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Relatórios de integridade aprimorada

Os relatórios de integridade aprimorada são um recurso que você pode habilitar no ambiente para permitir que o Elastic Beanstalk obtenha informações adicionais sobre os recursos no ambiente. O Elastic Beanstalk analisa as informações reunidas para fornecer uma visão melhor da integridade geral do ambiente e ajudar na identificação de problemas que podem fazer com que sua aplicação se torne indisponível. Para obter mais informações, consulte Relatórios e monitoramento de saúde aprimorados no Elastic Beanstalk.

Registros de EC2 instâncias da Amazon

As EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk geram registros que você pode visualizar para solucionar problemas com seu aplicativo ou arquivos de configuração. Os registros são criados pelo servidor web, servidor de aplicativos e scripts da plataforma Elastic Beanstalk e são armazenados localmente em AWS CloudFormation instâncias individuais. Você pode recuperá-los facilmente usando o console de gerenciamento do ambiente ou a CLI do EB. Você também pode configurar seu ambiente para transmitir registros para o Amazon CloudWatch Logs em tempo real. Para obter mais informações, consulte Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

Notificações do ambiente

É possível configurar o ambiente do Elastic Beanstalk para usar o Amazon Simple Notification Service (Amazon SNS) para notificá-lo sobre eventos importantes que afetam a aplicação. Especifique um endereço de e-mail durante ou após a criação do ambiente para receber e-mails de AWS quando ocorre um erro ou quando a integridade do seu ambiente muda. Para obter mais informações, consulte Notificações do ambiente do Elastic Beanstalk com o Amazon SNS.

CloudWatch Alarmes da Amazon

Usando CloudWatch alarmes, você observa uma única métrica durante um período de tempo especificado por você. Se a métrica exceder um determinado limite, uma notificação será enviada para um tópico AWS Auto Scaling ou política do Amazon SNS. CloudWatch os alarmes não invocam ações porque estão em um estado específico. Em vez disso, os alarmes do invocam ações quando o estado mudou e foi mantido por um número específico de períodos. Para obter mais informações, consulte Usando o Elastic Beanstalk com a Amazon CloudWatch.

AWS CloudTrail troncos

CloudTrail fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Elastic Beanstalk. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Elastic Beanstalk, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais. Para obter mais informações, consulte Registro de chamadas de API do Elastic Beanstalk com AWS CloudTrail.

AWS X-Ray depuração

X-Ray é um AWS serviço que reúne dados sobre as solicitações que seu aplicativo atende e os usa para criar um mapa de serviços que você pode usar para identificar problemas com seu aplicativo e oportunidades de otimização. Você pode usar o AWS Elastic Beanstalk console ou um arquivo de configuração para executar o daemon X-Ray nas instâncias do seu ambiente. Para obter mais informações, consulte Configurando a depuração AWS X-Ray.

Validação de conformidade para o Elastic Beanstalk

A segurança e a conformidade do AWS Elastic Beanstalk são avaliadas por auditores terceirizados como parte de vários programas de AWS conformidade. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros. AWS fornece uma lista frequentemente atualizada de AWS serviços no escopo de programas de conformidade específicos em AWS Serviços no escopo por programa de conformidade.

Relatórios de auditoria de terceiros estão disponíveis para você baixar usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact.

Para obter mais informações sobre programas de AWS conformidade, consulte Programas de AWS conformidade.

Sua responsabilidade de conformidade ao usar o Elastic Beanstalk é determinada pela confidencialidade de seus dados, pelas metas de conformidade da sua empresa e pelas regulamentações e leis aplicáveis. Se seu uso do Elastic Beanstalk estiver sujeito à conformidade com padrões como HIPAA, PCI ou FedRAMP, fornece recursos para ajudar a: AWS

- Guias de início rápido sobre segurança e conformidade — guias de implantação que discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados na segurança e na conformidade em. AWS
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Um whitepaper que descreve como as empresas podem usar AWS para criar aplicativos compatíveis com a HIPAA.
- AWS Recursos de conformidade — Uma coleção de pastas de trabalho e guias de conformidade que podem ser aplicados ao seu setor e local.
- AWS Config: um serviço que avalia até que ponto suas configurações de recursos estão compatíveis com práticas internas, diretrizes do setor e regulamentações.
- AWS Security Hub— Uma visão abrangente do seu estado de segurança interno AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Resiliência no Elastic Beanstalk

AWS Elastic Beanstalk gerencia e automatiza o uso da infraestrutura AWS global em seu nome. Ao usar o Elastic Beanstalk, você se beneficia dos mecanismos de disponibilidade e tolerância a falhas que ele oferece. AWS

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade.

AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância.

Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que executam o failover automaticamente entre as zonas de disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Segurança da infraestrutura no Elastic Beanstalk

Como serviço gerenciado, AWS Elastic Beanstalk é protegido pelos procedimentos AWS globais de segurança de rede descritos em nosso site de Melhores Práticas de Segurança, Identidade e Conformidade.

Você usa chamadas de API AWS publicadas para acessar o Elastic Beanstalk pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com Perfect Forward Secrecy (PFS – Sigilo de encaminhamento perfeito), como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria das plataformas modernas oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Análise de configuração e vulnerabilidade no Elastic Beanstalk

AWS e nossos clientes compartilham a responsabilidade de alcançar um alto nível de segurança e conformidade de componentes de software. AWS Elastic Beanstalk ajuda você a cumprir sua parte do modelo de responsabilidade compartilhada, fornecendo um recurso de atualizações gerenciadas. Esse recurso aplica automaticamente atualizações secundárias e patches para uma versão da plataforma compatível do Elastic Beanstalk.

Para obter mais informações, consulte Modelo de responsabilidade compartilhada para manutenção da plataforma do Elastic Beanstalk.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Melhores práticas de segurança para Elastic Beanstalk

AWS Elastic Beanstalk fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As práticas recomendadas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes no seu ambiente, trate-as como considerações úteis, não como requisitos.

Para outros tópicos de segurança do Elastic Beanstalk, consulte AWS Elastic Beanstalk segurança.

Práticas recomendadas de segurança preventiva

Os controles de segurança preventiva tentam evitar incidentes antes que ocorram.

Implemente o acesso de privilégio mínimo

O Elastic AWS Identity and Access Management Beanstalk fornece políticas gerenciadas (IAM) paraperfis de instância, funções de serviço e usuários do IAM. Essas políticas gerenciadas especificam todas as permissões que podem ser necessárias para a operação correta do ambiente e do aplicativo.

O aplicativo pode não exigir todas as permissões em nossas políticas gerenciadas. É possível personalizá-los e conceder apenas as permissões necessárias para que as instâncias do ambiente, o serviço do Elastic Beanstalk e os usuários executem suas tarefas. Isso é especialmente relevante para políticas de usuário, em que diferentes funções de usuário podem ter diferentes necessidades de permissão. A implementação do acesso de privilégio mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

Proteja dados confidenciais de aplicativos

Quando seu aplicativo precisar acessar informações confidenciais, como credenciais, chaves de API ou dados de configuração, siga estas práticas para manter a segurança:

- Recupere dados confidenciais diretamente do nosso AWS Secrets Manager AWS Systems Manager Parameter Store usando seus respectivos códigos SDKs ou APIs em seu aplicativo. Isso fornece a maneira mais segura e flexível de acessar informações confidenciais.
- Se você passar dados confidenciais do AWS Secrets Manager nosso AWS Systems Manager Parameter Store como variáveis de ambiente (consulteBusque segredos para variáveis de

ambiente), restrinja cuidadosamente o acesso aos pares de EC2 chaves e configure as funções apropriadas do IAM com permissões de privilégio mínimo para suas instâncias.

- Nunca imprima, registre ou exponha dados confidenciais no código do aplicativo, pois esses valores podem acabar em arquivos de log ou mensagens de erro que podem estar visíveis para usuários não autorizados.

Atualizar as plataformas regularmente

O Elastic Beanstalk lança regularmente novas versões de plataforma para atualizar todas as suas plataformas. As novas versões de plataforma fornecem atualizações de sistema operacional, runtime, servidor de aplicações, servidor Web e atualizações de componentes do Elastic Beanstalk. Muitas dessas atualizações de plataforma incluem correções de segurança importantes. Certifique-se de que os ambientes do Elastic Beanstalk estejam em execução em uma versão compatível da plataforma (normalmente a versão mais recente da plataforma). Para obter detalhes, consulte Atualizar a versão de plataforma do ambiente Elastic Beanstalk.

A maneira mais fácil de manter a plataforma do ambiente atualizada é configurar o ambiente para usar atualizações gerenciadas de plataforma.

Aplique IMDSv2 em instâncias do ambiente

As instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em seus ambientes do Elastic Beanstalk usam o serviço de metadados da instância (IMDS), um componente na instância, para acessar com segurança os metadados da instância. O IMDS suporta dois métodos para acessar dados: IMDSv1 e. IMDSv2 IMDSv2 usa solicitações orientadas à sessão e mitiga vários tipos de vulnerabilidades que poderiam ser usadas para tentar acessar o IMDS. Para obter detalhes sobre as vantagens de IMDSv2, consulte aprimoramentos para adicionar defesa em profundidade ao Serviço de Metadados de EC2 Instância.

IMDSv2 é mais seguro, então é uma boa ideia impor o uso de IMDSv2 em suas instâncias. Para impor IMDSv2, certifique-se de que todos os componentes do seu aplicativo sejam compatíveis e IMDSv2, em seguida, IMDSv1 desative-o. Para obter mais informações, consulte the section called “IMDS”.

Práticas recomendadas de segurança de detecção

Os controles de segurança de detecção identificam violações de segurança depois de ocorrerem. Eles podem ajudar a detectar uma possível ameaça ou incidente de segurança.

Implementar o monitoramento

O monitoramento é uma parte importante da manutenção da confiabilidade, segurança, disponibilidade e desempenho de suas soluções do Elastic Beanstalk. AWS fornece várias ferramentas e serviços para ajudá-lo a monitorar seus AWS serviços.

Veja a seguir alguns exemplos de itens a serem monitorados:

- CloudWatch Métricas da Amazon para o Elastic Beanstalk — Defina alarmes para as principais métricas do Elastic Beanstalk e para as métricas personalizadas do seu aplicativo. Para obter detalhes, consulte Usando o Elastic Beanstalk com a Amazon CloudWatch.
- AWS CloudTrail entradas — acompanhe ações que possam afetar a disponibilidade, como `UpdateEnvironment` ou`TerminateEnvironment`. Para obter detalhes, consulte Registro de chamadas de API do Elastic Beanstalk com AWS CloudTrail.

Habilitar AWS Config

AWS Config fornece uma visão detalhada da configuração dos AWS recursos em sua conta. Você pode ver como os recursos estão relacionados, obter um histórico de alterações de configuração e ver como os relacionamentos e as configurações foram alterados ao longo do tempo.

Você pode usar AWS Config para definir regras que avaliam as configurações de recursos para fins de conformidade de dados. AWS Config as regras representam as configurações ideais para seus recursos do Elastic Beanstalk. Se um recurso violar uma regra e for sinalizado como não compatível, AWS Config poderá alertá-lo usando um tópico do Amazon Simple Notification Service (Amazon SNS). Para obter mais detalhes, consulte Encontrando e monitorando recursos do Elastic Beanstalk com AWS Config.

Perfis do Elastic Beanstalk Service, perfis de instância e políticas de usuário

Os papéis são entidades com as quais você cria AWS Identity and Access Management (IAM) para aplicar permissões. Há perfis necessárias para que seu ambiente do Elastic Beanstalk funcione corretamente. Você também tem a opção de criar suas próprias políticas e perfis personalizados que podem ser atribuídos a usuários ou grupos.

Perfis necessários para seu ambiente do Elastic Beanstalk

Ao criar um ambiente, AWS Elastic Beanstalk solicita que você forneça as seguintes funções AWS Identity and Access Management (IAM):

- Função de serviço: o Elastic Beanstalk assume uma função de serviço para usar outras pessoas em seu nome. Serviços da AWS
- Perfil de instância O Elastic Beanstalk aplica um perfil de instância às instâncias da EC2 Amazon em seu ambiente. Essa ação permite que eles executem as tarefas necessárias, como a recuperação de informações do Amazon Simple Storage Service (Amazon S3) e o upload de logs no S3.

Crie a função de serviço e a função de perfil da EC2 instância

Se sua AWS conta não tiver um perfil de EC2 instância ou uma função de serviço, você deverá criar uma de cada uma usando o serviço IAM. Em seguida, você pode atribuir o perfil da EC2 instância e a função de serviço aos novos ambientes que você criar. O assistente de criação de ambiente guia você até o serviço IAM, para que você possa criar essas funções com as permissões necessárias.

Políticas e perfis opcionais para gerenciar seu ambiente do Elastic Beanstalk

Como opção, é possível criar políticas de usuário e aplicá-las aos usuários e grupos do IAM na sua conta. Com isso, os usuários podem criar e gerenciar aplicações e ambientes Elastic Beanstalk. Você também pode atribuir políticas gerenciadas do Elastic Beanstalk para acesso total e somente leitura a usuários ou grupos. Para obter mais informações sobre essas políticas, consulte the section called “Políticas de usuário”.

Você pode criar seus próprios perfis de instância e políticas de usuário para cenários avançados. Se suas instâncias precisarem acessar serviços que não estejam incluídos nas políticas padrão, você poderá criar uma nova política ou acrescentar políticas adicionais à política padrão. Se a política gerenciada for muito permissiva para o que você precisa, também é possível criar políticas de usuário mais restritivas. Para obter mais informações sobre AWS permissões, consulte o Guia do usuário do IAM.

Função de serviço do Elastic Beanstalk

Uma função de serviço é uma função do IAM que o console do Elastic Beanstalk assume ao chamar outros serviços em seu nome. Por exemplo, o Elastic Beanstalk usa uma função de serviço quando chama o Amazon Elastic Compute Cloud (Amazon) EC2, o Elastic Load Balancing e o EC2 Amazon Auto Scaling para coletar informações. APIs A função de serviço que o Elastic Beanstalk usa é a função que você especificou ao criar o ambiente do Elastic Beanstalk.

Existem duas políticas gerenciadas que são anexadas ao perfil de serviço. Essas políticas fornecem as permissões que permitem que o Elastic Beanstalk acesse AWS os recursos necessários para criar e gerenciar seus ambientes. Uma das políticas gerenciadas fornece permissões para monitoramento avançado de integridade e compatibilidade com o Amazon SQS no nível do operador e a outra fornece as permissões adicionais necessárias para atualizações gerenciadas da plataforma.

`AWSElasticBeanstalkEnhancedHealth`

Esta política concede permissões ao Elastic Beanstalk para monitorar a integridade do ambiente e da instância. Ela também inclui ações do Amazon SQS para permitir que o Elastic Beanstalk monitore a atividade das filas nos ambientes de operador. Para ver o conteúdo dessa política gerenciada, consulte a AWSElasticBeanstalkEnhancedHealthpágina no Guia de referência da política AWS gerenciada.

`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy`

Esta política concede permissões ao Elastic Beanstalk para atualizar ambientes em seu nome a fim de realizar atualizações gerenciadas de plataforma. Para ver o conteúdo dessa política gerenciada, consulte a AWSElasticBeanstalkManagedUpdatesCustomerRolePolicypágina no Guia de referência da política AWS gerenciada.

Agrupamentos de permissão no nível de serviço

Esta política é agrupada em declarações com base no conjunto de permissões fornecidas.

- *ElasticBeanstalkPermissions*— Esse grupo de permissões serve para chamar as ações de serviço do Elastic Beanstalk (Elastic Beanstalk). APIs
- *AllowPassRoleToElasticBeanstalkAndDownstreamServices* - Esse grupo de permissões permite que qualquer função seja transferida para o Elastic Beanstalk e para outros serviços downstream, como o AWS CloudFormation.
- *ReadOnlyPermissions*: esse grupo de permissões destina-se a coletar informações sobre o ambiente em execução.
- **OperationPermissions*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma.
- **BroadOperationPermissions*: os grupos com este padrão de nomenclatura destinam-se a acionar as operações necessárias para executar atualizações na plataforma. Eles também incluem permissões amplas para oferecer suporte a ambientes herdados.
- **TagResource*— Os grupos com esse padrão de nomenclatura são para chamadas que usam o tag-on-create APIs para anexar tags em recursos que estão sendo criados em um ambiente do Elastic Beanstalk.

Você pode criar um ambiente Elastic Beanstalk com qualquer das abordagens a seguir. Cada seção descreve como a abordagem lida com o perfil de serviço.

Console do Elastic Beanstalk

Se você criar um ambiente usando o console do Elastic Beanstalk, o Elastic Beanstalk solicita a criação de uma função de serviço chamada `aws-elasticbeanstalk-service-role`. Quando criado por meio do Elastic Beanstalk, esse perfil inclui uma política de confiança que permite ao Elastic Beanstalk assumir o perfil de serviço. As duas políticas gerenciadas descritas anteriormente neste tópico também estão vinculadas ao perfil.

Interface de linha de comando do Elastic Beanstalk (CLI do EB)

Você pode criar um ambiente usando o comando the section called “eb create” da interface da linha de comando do Elastic Beanstalk (CLI do EB). Se você não especificar um perfilde serviço por meio da opção `--service-role`. O Elastic Beanstalk cria o mesmo perfil de serviço padrão `aws-elasticbeanstalk-service-role`. Se a função de serviço padrão já existir, o Elastic Beanstalk

a usará para o novo ambiente. Quando criado por meio do Elastic Beanstalk, esse perfil inclui uma política de confiança que permite ao Elastic Beanstalk assumir o perfil de serviço. As duas políticas gerenciadas descritas anteriormente neste tópico também estão vinculadas ao perfil.

API do Elastic Beanstalk

Você pode criar um ambiente usando a ação `CreateEnvironment` da API do Elastic Beanstalk. Se você não especificar um perfil de serviço, o Elastic Beanstalk criará um perfil vinculado ao serviço de monitoramento. Esse é um tipo exclusivo de função de serviço predefinido pelo Elastic Beanstalk para incluir todas as permissões que o serviço exige para ligar para outras pessoas em seu nome. Serviços da AWS A função vinculada a serviço é associada à sua conta. O Elastic Beanstalk cria essa função uma só vez e a reutiliza ao criar ambientes adicionais. Você também pode usar o IAM para criar antecipadamente a função vinculada ao serviço de monitoramento para sua conta. Quando sua conta tem um perfil vinculado ao serviço de monitoramento, é possível usá-lo para criar um ambiente usando a API, o console ou a CLI do Elastic Beanstalk. Para obter instruções sobre como usar funções vinculadas a serviços com ambientes do Elastic Beanstalk, consulte Usar funções vinculadas ao serviço para o Elastic Beanstalk.

Para obter mais informações sobre perfis de serviço, consulte Gerenciar funções de serviço do Elastic Beanstalk.

Perfil de instância do Elastic Beanstalk

Um perfil de instância é uma função do IAM aplicada às EC2 instâncias da Amazon que são lançadas em seu ambiente do Elastic Beanstalk. Ao criar um ambiente do Elastic Beanstalk, você especifica o perfil da instância que é usado EC2 quando suas instâncias realizam as seguintes ações:

- Recuperam versões de aplicações do Amazon Simple Storage Service (Amazon S3)
- Gravam logs no Amazon S3
- Em ambientes integrados do AWS X-Ray, fazem upload dos dados de depuração para o X-Ray
- Em ambientes do Docker gerenciados pelo ESC, coordenam implantações de contêiner com o Amazon Elastic Container Service (Amazon ECS)
- Em ambientes de operador, leem a partir de uma fila do Amazon Simple Queue Service (Amazon SQS)
- Em ambientes de operador, escolhem o líder com o Amazon DynamoDB
- Em ambientes de trabalho, publique métricas de integridade da instância na Amazon CloudWatch

Políticas gerenciadas

O Elastic Beanstalk fornece um conjunto de políticas gerenciadas que EC2 permitem que as instâncias em seu ambiente realizem as operações necessárias. As políticas gerenciadas que se seguem são as necessárias para os casos de uso básicos.

- `AWSElasticBeanstalkWebTier`
- `AWSElasticBeanstalkWorkerTier`
- `AWSElasticBeanstalkMulticontainerDocker`

Se seu aplicativo web exigir acesso a outros Serviços da AWS, adicione declarações ou políticas gerenciadas ao perfil da instância que permita o acesso a esses serviços. Para obter mais informações, consulte Adicionar permissões ao perfil da instância padrão.

Criação de um perfil de EC2 instância

Se sua AWS conta não tiver um perfil de EC2 instância, você deverá criar um usando o serviço IAM. Em seguida, você pode atribuir o perfil da EC2 instância aos novos ambientes criados por você. As etapas de criação do ambiente no console do Elastic Beanstalk fornecem acesso ao console do IAM, para que você possa EC2 criar um perfil de instância com as permissões necessárias.

Você também pode criar um perfil de EC2 instância acessando diretamente o console do IAM, sem precisar passar pelo console do Elastic Beanstalk. Para obter etapas detalhadas para criar um perfil de instância do EC2 Elastic Beanstalk no console do IAM, consulte. Criar um perfil da instância

Política de usuário do Elastic Beanstalk

A fim de evitar o uso de sua conta raiz ou o compartilhamento de credenciais, crie usuários do IAM para cada pessoa que use o Elastic Beanstalk. Como uma prática recomendada de segurança, conceda apenas as permissões para que esses usuários acessem os serviços e os recursos de que precisam.

O Elastic Beanstalk requer permissões não apenas para suas próprias ações de API, mas também para vários outros serviços da AWS . O Elastic Beanstalk usa permissões de usuário para iniciar recursos em um ambiente. Esses recursos incluem EC2 instâncias, um balanceador de carga do Elastic Load Balancing e um grupo de Auto Scaling. O Elastic Beanstalk também usa permissões de usuário para salvar registros e modelos no Amazon Simple Storage Service (Amazon S3), enviar

notificações para o Amazon SNS, atribuir perfis de instância e publicar métricas no. CloudWatch O Elastic AWS CloudFormation Beanstalk exige permissões para orquestrar implantações e atualizações de recursos. Ele também exige permissões do Amazon RDS para criar bancos de dados quando necessário e permissões do Amazon SQS para criar filas para ambientes de operador.

Para obter mais informações sobre políticas de usuário, consulte Gerenciar políticas de usuário do Elastic Beanstalk.

Tutoriais e amostras

Tutoriais específicos de linguagem e estrutura estão espalhados por todo o Guia do AWS Elastic Beanstalk Desenvolvedor. Tutoriais novos e atualizados são adicionados a essa lista à medida que são publicados. As atualizações mais recentes são mostradas primeiro.

Esses tutoriais destinam-se a usuários intermediários e podem não conter instruções de etapas básicas, como cadastrar-se na AWS. Se esta é sua primeira vez usando o AWS Elastic Beanstalk, confira o passo a passo de introdução para colocar seu primeiro ambiente do Elastic Beanstalk em funcionamento.

- Ruby on Rails: Implantar uma aplicação rails no Elastic Beanstalk
- Ruby e Sinatra: Implantar uma aplicação sinatra no Elastic Beanstalk
- Configuração de HA do PHP e MySQL: Implantar uma aplicação PHP de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk
- PHP e Laravel: Implantar uma aplicação Laravel no Elastic Beanstalk
- PHP e CakePHP: Implantar uma aplicação CakePHP no Elastic Beanstalk
- Configuração de HA do PHP e Drupal: Implantar um site Drupal de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk
- Configuração de PHP e WordPress HA - Implantação de um WordPress site de alta disponibilidade com um banco de dados externo do Amazon RDS no Elastic Beanstalk
- Node.js com configuração de HA do DynamoDB: Implantar uma aplicação Node.js com o DynamoDB no Elastic Beanstalk
- ASP.NET Core: QuickStart: Implante um aplicativo ASP.NET no Elastic Beanstalk
- Python e Flask: Implantar uma aplicação Flask no Elastic Beanstalk
- Python e Django: Implantar uma aplicação Django no Elastic Beanstalk
- Node.js e Express: Implantar uma aplicação Node.js Express no Elastic Beanstalk
- Docker, PHP e nginx: Criação de um ambiente Docker gerenciado pelo ECS com o console do Elastic Beanstalk

É possível fazer download das aplicações demonstrativas usadas pelo Elastic Beanstalk ao criar um ambiente, sem fornecer um pacote de origem com os seguintes links:

- Docker: docker.zip

- Docker com vários contêineres — 2.zip docker-multicontainer-v
- Docker pré-configurado (Glassfish) — 1.zip docker-glassfish-v
- Go: go.zip
- Corretto: corretto.zip
- Tomcat: tomcat.zip
- .NET Core no Linux — dotnet-core-linux.zip
- Núcleo.NET — dotnet-asp-windows.zip
- Node.js: nodejs.zip
- PHP: php.zip
- Python: python.zip
- Ruby: ruby.zip

Aplicativos de amostra mais envolvidos que mostram o uso de estruturas, bibliotecas e ferramentas adicionais da Web estão disponíveis como projetos de código aberto em GitHub:

- Balanceamento de carga WordPress (tutorial) — Arquivos de configuração para instalação WordPress segura e execução em um ambiente Elastic Beanstalk com balanceamento de carga.
- Drupal com balanceamento de carga (tutorial): arquivos de configuração e instruções para instalar com segurança o Drupal e executá-lo em um ambiente do Elastic Beanstalk com balanceamento de carga.
- Scorekeep - API RESTful da web que usa a estrutura Spring e a AWS SDK para Java para fornecer uma interface para criar e gerenciar usuários, sessões e jogos. A API faz parte do pacote de um aplicativo Web Angular 1.5 que a consome via HTTP. Inclui filiais que mostram a integração com o Amazon Cognito e o Amazon Relational Database Service. AWS X-Ray

  O aplicativo usa recursos da plataforma Java SE para fazer download de dependências e criar na instância, minimizando o tamanho do pacote de origem. O aplicativo também inclui arquivos de configuração nginx que substituem a configuração padrão para servir o aplicativo web de front-end estaticamente na porta 80 através do proxy e encaminhar as solicitações para atalhos em `/api` para a API em execução em `localhost:5000`.
- Does it Have Snakes? - Aplicação Tomcat que mostra o uso do RDS em uma aplicação da Web Java EE no Elastic Beanstalk. O projeto mostra o uso dos arquivos de configuração Servlets JSPs, Simple Tag Support, Tag Files, JDBC, SQL, Log4J, Bootstrap, Jackson e Elastic Beanstalk.

- Locust Load Generator – esse projeto mostra o uso de recursos da Plataforma Java SE para instalar e executar Locust, uma ferramenta de geração de carga escrita em Python. O projeto inclui arquivos de configuração que instalam e configuram o Locust, um script de compilação que configura uma tabela do DynamoDB, e um Procfile que executa o Locust.
- Share Your Thoughts (tutorial): aplicação em PHP que mostra o uso de MySQL no Amazon RDS, Composer e arquivos de configuração.
- Uma nova inicialização (tutorial) - exemplo de aplicativo Node.js que mostra o uso do DynamoDB, o SDK do Node.js, AWS do gerenciamento JavaScript de pacotes npm e dos arquivos de configuração.

Migração de aplicativos IIS para o Elastic Beanstalk

AWS Elastic Beanstalk fornece um caminho de migração simplificado para seus aplicativos Windows executados nos Serviços de Informações da Internet (IIS). A capacidade de migração descrita neste capítulo reduz significativamente o tempo e a complexidade normalmente associados às migrações para a nuvem, ajudando você a manter a funcionalidade do aplicativo e a integridade da configuração durante a transição para o. AWS

A eb migrate operação

Use o eb migrate comando na interface de linha de comando do Elastic Beanstalk (EB CLI) para descobrir, empacotar e implantar automaticamente seus aplicativos IIS no. Nuvem AWS O processo mantém a funcionalidade do aplicativo e preserva suas configurações, incluindo associações, pools de aplicativos e configurações de autenticação.

As etapas a seguir resumem o processo que a `eb migrate` operação executa para fazer a transição do aplicativo para o Nuvem AWS:

1. Descubra os sites do IIS e suas configurações.
2. Conteúdo e configuração do aplicativo Package.
3. Crie o ambiente e o aplicativo do Elastic Beanstalk.
4. Implante o aplicativo com configurações preservadas.

Opções de execução de fluxo de trabalho e localização

O eb migrate comando fornece opções para fluxos de trabalho de migração flexíveis e locais de execução. Por padrão, execute o comando no servidor de destino que contém o aplicativo que você deseja migrar para o Elastic Beanstalk. Se você não conseguir executar comandos diretamente no servidor do aplicativo, use a `remote` opção de executar o comando a partir de um host bastion que se conecta ao servidor de destino que contém seu aplicativo e suas configurações. Para concluir a migração em duas etapas, você também pode gerar o pacote de migração sem implantá-lo usando a `archive-only` opção e, em seguida, implantá-lo posteriormente, conforme sua conveniência, usando a `archive` opção.

Para obter informações de referência sobre o eb migrate comando, consultethe section called “eb migrate”.

Tópicos

Os tópicos a seguir fornecem informações detalhadas sobre a migração de aplicativos do IIS para o Elastic Beanstalk:

- the section called “Pré-requisitos”- Entenda o software, o acesso e as permissões necessários para migrar seus aplicativos do Windows para AWS Elastic Beanstalk ambientes.
- the section called “Glossário de migração”- Entenda como os componentes do IIS são mapeados para os recursos do Elastic Beanstalk
- the section called “Mapeamento de migração do IIS para o Elastic Beanstalk”- Entenda como os componentes do IIS são mapeados para os recursos do Elastic Beanstalk
- the section called “Migrações básicas do IIS”- Aprenda a realizar migrações básicas
- the section called “Cenários avançados de migração”- Lidar com cenários complexos de migração
- the section called “Configuração de segurança”- Defina as configurações de segurança durante a migração
- the section called “Configuração de rede”- Gerenciar configurações de rede e portas
- the section called “Solução de problemas e diagnósticos”- Solucionar problemas comuns de migração
- the section called “Opções de migração: EB CLI versus MGN”- Compare duas opções principais de migração.

Pré-requisitos

Antes de usar o eb migrate comando, certifique-se de que seu ambiente atenda aos seguintes requisitos:

Instalação e versão do IIS

O servidor do qual você está migrando deve executar os Serviços de Informações da Internet (IIS) versão 7.0 ou posterior. O IIS 10.0 no Windows Server 2016 ou posterior fornece o ambiente mais compatível para migração.

Para verificar sua versão do IIS, execute o seguinte comando:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\InetStp\"
...
SetupString             : IIS 10.0
VersionString           : Version 10.0
...
```

Requisitos do Windows Server

Seu ambiente de origem deve executar o Windows Server 2016 ou posterior para otimizar a compatibilidade. O Elastic Beanstalk é compatível com essas versões do Windows Server como plataformas de destino:

- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

Instalação do EB CLI

- Fluxo de trabalho padrão (sem a `--remote` opção):
  - O Python e a interface de linha de comando (EB CLI) do Elastic Beanstalk devem estar instalados no servidor que contém o aplicativo que você deseja migrar para o Elastic Beanstalk. Embora não seja necessário, recomendamos instalar a CLI do EB dentro de uma `virtualenv` sandbox, conforme descrito em. the section called “Instale no virtualenv”
- Usando a `--remote` opção:
  - O Python e a interface de linha de comando (EB CLI) do Elastic Beanstalk devem estar instalados no seu host bastion. Embora não seja necessário, recomendamos instalar a CLI do EB dentro de uma `virtualenv` sandbox, conforme descrito em. the section called “Instale no virtualenv”

Permissões obrigatórias do

Você precisa das seguintes credenciais e permissões:

- Privilégios de administrador no servidor IIS de origem ou no host bastion (se estiver usando a `--remote` opção).
- AWS credenciais com permissões para criar e gerenciar recursos do Elastic Beanstalk

Web Deploy 3.6

A ferramenta Microsoft Web Deploy (versão 3.6 ou posterior) deve ser instalada no servidor de origem ou no host bastion (se estiver usando a `--remote` opção). Essa ferramenta é usada eb migrate para empacotar seus aplicativos.

Para verificar a instalação, execute o seguinte 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\
...
```

Para obter instruções de instalação, consulte Instalando e configurando o Web Deploy no IIS 8.0 ou posterior no site de documentação do produto Microsoft Windows.

Requisitos de rede

- Fluxo de trabalho padrão (sem a *--remote* opção):
  - Seu servidor de origem deve ter acesso de saída à Internet aos AWS serviços.
- Usando a *--remote* opção:
  - Seu servidor de origem deve ter acesso de saída à Internet aos AWS serviços.
  - Configure as regras de entrada de grupos de segurança adequadas que permitam uma conexão de rede de saída do seu host bastion e uma conexão de entrada na máquina remota. Certifique-se de que o IP do bastion host esteja na lista de permissões via TCP na porta 22 para acessar a máquina remota.
  - Certifique-se de que seu cliente SSH esteja instalado e em execução em sua máquina remota, bem como em seu host bastion.
  - Verifique se a configuração do firewall contém as regras apropriadas que abrem a porta 22 ou permitem a conexão com o cliente.
  - Teste sua conexão inserindo manualmente o SSH no host remoto a partir do bastion host antes de tentar a migração.

Glossário de migração

Este glossário fornece definições dos principais termos e conceitos relacionados ao IIS, ao Elastic Beanstalk e à migração de aplicativos do IIS para o Elastic Beanstalk.

Termos do Windows, IIS e.NET

IIS

Internet Information Services, um software de servidor web desenvolvido pela Microsoft para uso com o Windows Server. O IIS hospeda sites, aplicativos da Web e serviços da Web, fornecendo

uma plataforma para executar o ASP.NET e outras tecnologias da Web. Durante a migração para o Elastic Beanstalk, os sites do IIS e suas configurações são empacotados e implantados em instâncias do Windows Server na nuvem. AWS

As versões 7.0 e posteriores do IIS têm suporte para migração, com o IIS 10.0 no Windows Server 2016 ou posterior fornecendo o ambiente mais compatível.

.NET Estrutura

Uma plataforma de desenvolvimento de software desenvolvida pela Microsoft para criar e executar aplicativos do Windows. Ele fornece uma grande biblioteca de classes chamada Framework Class Library (FCL) e oferece suporte à interoperabilidade de linguagens em várias linguagens de programação.

Ao migrar para o Elastic Beanstalk, os aplicativos criados no.NET Framework continuam sendo executados na mesma versão da estrutura no ambiente de nuvem. O Elastic Beanstalk oferece suporte a várias versões do.NET Framework (4.x) em suas plataformas Windows Server.

.NET Core

Um sucessor multiplataforma e de código aberto do .NET Framework, projetado para ser mais modular e leve. O.NET Core (agora chamado simplesmente de .NET 5 e versões posteriores) permite que os desenvolvedores criem aplicativos que são executados no Windows, Linux e macOS.

Ao migrar aplicativos criados no.NET Core para o Elastic Beanstalk, você pode escolher entre plataformas Windows Server ou plataformas baseadas em Linux, dependendo dos requisitos e dependências do seu aplicativo.

Common Language Runtime (CLR)

O componente de máquina virtual do.NET Framework que gerencia a execução de programas.NET. O CLR fornece serviços como gerenciamento de memória, segurança de tipos, tratamento de exceções, coleta de lixo e gerenciamento de threads.

Ao migrar para o Elastic Beanstalk, a versão CLR apropriada está automaticamente disponível na plataforma Windows Server que você selecionar, garantindo a compatibilidade com os requisitos do seu aplicativo.

Site

Um contêiner lógico no IIS que representa um aplicativo ou serviço Web, identificado por uma associação exclusiva de endereço IP, porta e cabeçalho do host. Cada site do IIS tem seu

próprio pool de aplicativos, associações e definições de configuração e pode conter um ou mais aplicativos.

Aplicativo

Um agrupamento de arquivos de conteúdo e código em um site do IIS que processa solicitações de um espaço de URL específico. Os aplicativos no IIS podem ter suas próprias configurações, que normalmente são armazenadas em `web.config` arquivos.

Ao migrar para o Elastic Beanstalk, os aplicativos são preservados com sua estrutura de caminhos e suas definições de configuração. O processo de migração garante que os aplicativos aninhados mantenham sua hierarquia e caminhos de URL no ambiente de nuvem.

ApplicationPool

Um recurso do IIS que isola aplicativos web para melhorar a segurança, a confiabilidade e o gerenciamento do desempenho. Os pools de aplicativos definem o ambiente de execução dos aplicativos, incluindo a versão do.NET Framework, o modo de pipeline e as configurações de identidade.

VirtualDirectory

Um mapeamento de diretórios no IIS que permite que o conteúdo seja servido de um local fora do diretório raiz do site. Os diretórios virtuais permitem que você organize o conteúdo em diferentes locais físicos e, ao mesmo tempo, apresente uma estrutura de URL unificada aos usuários.

Ao migrar para o Elastic Beanstalk, os diretórios virtuais são preservados com seus mapeamentos de caminhos. O eb migrate comando cria a estrutura de diretórios e as configurações necessárias no ambiente de nuvem para manter os mesmos caminhos de URL.

ARR

Application Request Routing, uma extensão do IIS que fornece recursos de balanceamento de carga e proxy para servidores web. O ARR permite roteamento baseado em URL, encaminhamento de solicitações HTTP e distribuição de carga em vários servidores.

Durante a migração para o Elastic Beanstalk, as configurações de ARR são preservadas por meio da instalação de EC2 recursos de ARR nas instâncias e da configuração de regras de roteamento apropriadas. Para cenários complexos de roteamento, o processo de migração também pode aproveitar as regras do Application Load Balancer para implementar funcionalidades semelhantes.

Reescrita de URL

Um módulo do IIS que modifica as solicitações URLs com base em regras definidas antes que elas cheguem ao aplicativo web. A reescrita de URL permite a manipulação, o redirecionamento e a entrega de conteúdo de URL com base em padrões e condições.

Ao migrar para o Elastic Beanstalk, as regras de regravação de URL `web.config` dos seus arquivos são traduzidas em regras de roteamento do ALB sempre que possível ou preservadas na configuração do IIS nas instâncias. EC2 Isso garante que os padrões e redirecionamentos de URL continuem funcionando conforme o esperado no ambiente de nuvem.

msdeploy.exe

Uma ferramenta de linha de comando usada para implantar aplicativos e sites da Web em servidores IIS. Também conhecido como Web Deploy, ele fornece uma maneira de empacotar, sincronizar e implantar aplicativos web, sites e configurações de servidores.

O eb migrate comando usa o Web Deploy (versão 3.6 ou posterior) para empacotar seus aplicativos durante a migração para o Elastic Beanstalk. Essa ferramenta deve estar instalada no servidor de origem para que o processo de migração funcione corretamente.

Caminho físico

A localização real do sistema de arquivos em que os arquivos de conteúdo de um site ou aplicativo do IIS são armazenados. Os caminhos físicos podem apontar para diretórios locais, compartilhamentos de rede ou outros locais de armazenamento acessíveis ao servidor IIS.

Durante a migração para o Elastic Beanstalk, os caminhos físicos são mapeados para os locais EC2 apropriados nas instâncias do seu ambiente. O processo de migração preserva a estrutura do conteúdo e garante que todos os arquivos sejam implantados adequadamente no ambiente de nuvem.

ApplicationHost.config

O arquivo de configuração raiz do IIS que define as configurações de todo o servidor e contém configurações para todos os sites, aplicativos e diretórios virtuais. Esse arquivo está localizado no `%windir%\System32\inetsrv\config` diretório e controla o comportamento geral do servidor IIS.

Ao migrar para o Elastic Beanstalk, `applicationHost.config` as configurações relevantes do são extraídas e aplicadas à configuração do IIS nas instâncias do seu ambiente. EC2 Isso garante que as configurações de todo o servidor sejam preservadas durante a migração.

web.config

Um arquivo de configuração baseado em XML usado em aplicativos ASP.NET para controlar as configurações, a segurança e o comportamento do aplicativo no nível do aplicativo ou do diretório. `web.config`os arquivos podem conter configurações para autenticação, autorização, estado da sessão, compilação e parâmetros personalizados do aplicativo.

Durante a migração para o Elastic `web.config` Beanstalk, os arquivos são preservados e implantados com seu aplicativo. O processo de migração garante que as configurações específicas do aplicativo continuem funcionando conforme o esperado no ambiente de nuvem.

DefaultDocument

Um recurso do IIS que especifica o arquivo padrão a ser exibido quando um usuário solicita um diretório sem especificar um nome de arquivo. Os documentos padrão são habilitados por padrão, e o IIS 7 define os seguintes arquivos de documentos padrão no `applicationHost.config` arquivo como padrões para todo o servidor: Default.htm, Default.asp, Index.htm, Index.html, Iisstart.htm.

Ao migrar para o Elastic Beanstalk, as configurações padrão do documento são preservadas na configuração do IIS nas instâncias, garantindo que as solicitações de diretório sejam tratadas de forma consistente EC2 no ambiente de nuvem.

System.WebServer

Uma seção de configuração em `web.config` ou `applicationHost.config` que contém configurações específicas do IIS para módulos, manipuladores e outros comportamentos do servidor. Esta seção controla como o IIS processa solicitações, gerencia módulos e configura os recursos do servidor.

Durante a migração para o Elastic Beanstalk, as configurações do System.WebServer são preservadas nos arquivos do seu `web.config` aplicativo e aplicadas à instalação do IIS nas instâncias do seu ambiente. EC2 Isso garante que os comportamentos específicos do IIS sejam mantidos no ambiente de nuvem.

Termos do Elastic Beanstalk

Plataforma

Uma combinação de sistema operacional, tempo de execução da linguagem de programação, servidor web, servidor de aplicativos e componentes do Elastic Beanstalk que definem a pilha de software para execução de aplicativos.

Para migrações do Windows, o Elastic Beanstalk fornece plataformas baseadas no Windows Server 2016, 2019 e 2022 com IIS e várias versões do.NET Framework para garantir a compatibilidade com seu ambiente de origem.

SolutionStack

Uma configuração de plataforma predefinida no Elastic Beanstalk que especifica o sistema operacional, o tempo de execução e outros componentes necessários para executar um aplicativo. Conceitualmente idêntico a uma plataforma e usado de forma intercambiável para operar ambientes.

Durante a migração, o eb migrate comando seleciona uma pilha de soluções apropriada com base na configuração do seu ambiente de origem, garantindo a compatibilidade com seus aplicativos do IIS.

CreateEnvironment

Uma ação da API do Elastic Beanstalk que cria um novo ambiente para hospedar uma versão do aplicativo. Essa API é usada pelo eb migrate comando para provisionar os AWS recursos necessários para seu aplicativo migrado.

O processo de migração configura os parâmetros de ambiente apropriados com base no seu ambiente IIS de origem, incluindo tipo de instância, variáveis de ambiente e configurações de opções.

CreateApplicationVersion

Uma ação da API do Elastic Beanstalk que cria uma nova versão do aplicativo a partir de um pacote de origem armazenado no Amazon S3. O eb migrate comando usa essa API para registrar seu aplicativo IIS empacotado como uma versão no Elastic Beanstalk.

Durante a migração, os arquivos e a configuração do seu aplicativo são empacotados, enviados para o Amazon S3 e registrados como uma versão do aplicativo antes da implantação.

DescribeEvents

Uma ação da API do Elastic Beanstalk que recupera uma lista de eventos para um ambiente, incluindo implantações, alterações de configuração e problemas operacionais. O eb migrate comando usa essa API para monitorar o progresso da sua migração.

Você também pode usar o eb events comando após a migração para visualizar o histórico de eventos do seu ambiente.

DescribeEnvironmentHealth

Uma ação da API do Elastic Beanstalk que fornece informações detalhadas de saúde sobre as instâncias e outros componentes de um ambiente. Essa API é usada para verificar a integridade do seu aplicativo migrado após a implantação.

Após a migração, você pode usar o eb health comando para verificar o status do seu ambiente e identificar quaisquer problemas que precisem de atenção.

Saúde D

Um agente de monitoramento no Elastic Beanstalk que coleta métricas, monitora registros e relata o status de saúde EC2 de instâncias em um ambiente. O HealthD fornece relatórios de integridade aprimorados para seus aplicativos migrados.

Após a migração, o HealthD monitora o desempenho do aplicativo, a utilização dos recursos e as taxas de sucesso das solicitações, fornecendo uma visão abrangente da integridade do seu ambiente.

Registros do pacote

Um recurso do Elastic Beanstalk que comprime e carrega registros EC2 de instâncias para o Amazon S3 para armazenamento e análise centralizados. Esse recurso ajuda você a solucionar problemas com seus aplicativos migrados.

Após a migração, você pode usar o eb logs comando para recuperar e visualizar os registros do seu ambiente.

aws-windows-deployment-manifest.json

Um arquivo que descreve o conteúdo, as dependências e a configuração de um pacote de software ou aplicativo. Esse manifesto é gerado durante o processo de migração para definir como seus aplicativos IIS devem ser implantados no Elastic Beanstalk.

seção de manifesto personalizada

Uma seção interna `aws-windows-deployment-manifest.json` que fornece controle personalizado sobre a implantação do aplicativo. Esta seção contém PowerShell scripts e comandos que são executados durante o processo de implantação.

Durante a migração, seções de manifesto personalizadas são geradas para lidar com aspectos específicos da configuração do IIS, como configuração do diretório virtual, gerenciamento de permissões e configuração do pool de aplicativos.

EB CLI

Uma ferramenta de linha de comando que fornece comandos para criar, configurar e gerenciar aplicativos e ambientes do Elastic Beanstalk. O EB CLI inclui o eb migrate comando específico para migrar aplicativos do IIS para o Elastic Beanstalk.

Após a migração, você pode continuar usando a CLI do EB para gerenciar seu ambiente, implantar atualizações, monitorar a integridade e realizar outras tarefas administrativas.

Configurações da opção

Valores de configuração que definem como o Elastic Beanstalk provisiona AWS e configura recursos em seu ambiente. As configurações das opções são organizadas em namespaces que representam diferentes componentes do seu ambiente, como balanceadores de carga, instâncias e processos do ambiente.

Durante a migração, o eb migrate comando gera as configurações de opções apropriadas com base na configuração do IIS para garantir que seu ambiente de nuvem corresponda aos recursos do seu ambiente de origem. Para obter mais informações, consulte Opções de configuração no Guia do desenvolvedor do Elastic Beanstalk.

aws:elbv2:listener:default

Um namespace de configuração do Elastic Beanstalk para o ouvinte padrão em um Application Load Balancer. Durante a migração, esse namespace é configurado com base nas ligações do site do IIS para garantir o roteamento de tráfego adequado.

O ouvinte padrão normalmente manipula o tráfego HTTP na porta 80, que é então encaminhado para as instâncias do seu aplicativo de acordo com as regras de roteamento.

aws:elbv2: listener: listener_port

Um namespace de configuração do Elastic Beanstalk para uma porta de ouvinte específica em um Application Load Balancer. Esse namespace é usado para configurar ouvintes adicionais para seus aplicativos migrados, como HTTPS na porta 443.

Durante a migração, os ouvintes são criados com base nas ligações de portas dos sites do IIS, garantindo que seus aplicativos permaneçam acessíveis nas mesmas portas do ambiente de origem.

aws:elbv2: listenerrule:rule_name

Um namespace de configuração do Elastic Beanstalk para definir regras de roteamento para um ouvinte do Application Load Balancer. Essas regras determinam como as solicitações recebidas são roteadas para diferentes grupos-alvo com base nos padrões de caminho ou nos cabeçalhos do host.

Durante a migração, as regras de ouvinte são criadas para corresponder à estrutura de URL dos aplicativos do IIS, garantindo que as solicitações sejam roteadas para os caminhos corretos do aplicativo.

aws: elasticbeanstalk: ambiente: processo: padrão

Um namespace de configuração do Elastic Beanstalk para o processo padrão em um ambiente. Esse namespace define como o processo padrão do aplicativo Web é configurado, incluindo configurações de verificação de integridade, mapeamentos de portas e configurações de proxy.

Durante a migração, o processo padrão é configurado com base nas configurações do site principal do IIS, garantindo o monitoramento adequado da integridade e o tratamento de solicitações.

aws: elasticbeanstalk: ambiente: processo: nome_de_processo

Um namespace de configuração do Elastic Beanstalk para um processo nomeado específico em um ambiente. Esse namespace permite definir vários processos com configurações diferentes, semelhante a ter vários pools de aplicativos no IIS.

Durante a migração, processos adicionais podem ser criados para representar diferentes vinculações de sites do seu ambiente de origem.

> **Note**
>
> Para obter mais informações sobre alguns dos termos descritos neste tópico, consulte os seguintes recursos:
>
> - Ações da API do Elastic Beanstalk — Referência da API AWS Elastic Beanstalk
> - Plataformas do Elastic Beanstalk, incluindo versões de plataforma compatíveis - Plataformas suportadas no guia de plataformas AWS Elastic Beanstalk
> - Namespaces de configuração do Elastic Beanstalk - neste guia Opções gerais para todos os ambientes
> - O EB CLI ou comandos específicos do EB CLI - neste guia Configuração da interface de linha de comando do EB (EB CLI) para gerenciar o Elastic Beanstalk

Termos em Python

pip

O instalador de pacotes para Python, usado para instalar e gerenciar pacotes de software escritos em Python. O EB CLI é instalado e atualizado usando pip.

Durante o processo de migração, o pip é usado para instalar o pacote EB CLI e suas dependências no servidor de origem, fornecendo as ferramentas necessárias para a migração.

PyPi

Python Package Index, o repositório oficial para pacotes de software Python de terceiros, do qual o pip recupera e instala pacotes. O EB CLI e suas dependências são hospedados no PyPI.

Ao instalar o EB CLI para migração, o pip se conecta ao PyPI para baixar e instalar os pacotes necessários.

ambiente virtual

Uma ferramenta para criar ambientes Python isolados, permitindo que diferentes projetos tenham suas próprias dependências e pacotes sem conflitos. O uso do virtualenv é recomendado ao instalar a CLI do EB para evitar conflitos com outros aplicativos Python.

Criar um ambiente virtual antes de instalar o EB CLI garante que as ferramentas de migração tenham um ambiente limpo e isolado com as dependências corretas.

pywin32

Um conjunto de extensões do Python que fornecem acesso a muitos dos Windows APIs, permitindo a interação com o sistema operacional Windows e seus componentes. O EB CLI usa pywin32 para interagir com recursos específicos do Windows durante a migração.

Durante o processo de migração, o pywin32 é usado para acessar a configuração do IIS, as configurações do registro do Windows e outras informações do sistema necessárias para empacotar e migrar adequadamente seus aplicativos.

pythonnet

Um pacote que permite que o código Python interaja com aplicativos.NET Framework e o.NET Core. Essa integração permite que o EB CLI trabalhe com componentes do.NET durante o processo de migração.

O processo de migração pode usar pythonnet para interagir com assemblies e componentes do.NET ao analisar e empacotar seus aplicativos para implantação no Elastic Beanstalk.

Executando migrações básicas do IIS

Esta seção orienta você pelo processo de migração de seus aplicativos do IIS para o Elastic eb migrate Beanstalk usando o comando.

Explorando seu ambiente IIS

Antes de fazer qualquer alteração, você deve entender quais recursos existem no seu servidor. Comece explorando seus sites do IIS executandoeb migrate explore, conforme mostrado no exemplo a seguir:

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

Esse comando revela seus sites do IIS. Consulte a lista a seguir:

```
Default Web Site
Intranet
API.Internal
Reports
```

Para obter uma visão detalhada da configuração de cada site, incluindo vinculações, aplicativos e diretórios virtuais, adicione a `--verbose` opção, conforme mostrado neste exemplo:

```
PS C:\migrations_workspace> eb migrate explore --verbose
```

A lista a seguir mostra as informações abrangentes sobre seu ambiente que o comando fornece:

```
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:
...
```

Entendendo o resultado da descoberta

A saída detalhada fornece as seguintes informações críticas para o planejamento da migração:

Sites

A saída da descoberta lista todos os sites do IIS em seu servidor. Cada site é identificado por seu nome (por exemplo, “Site padrão”, “Intranet”, “API.internal”) e numerado sequencialmente. Quando existem vários sites em um servidor, o `eb migrate` comando pode empacotar e implantar cada um separadamente ou em conjunto, dependendo da sua estratégia de migração.

Ligações

As vinculações de protocolo revelam quais protocolos (HTTP/HTTPS) seus sites usam e em quais portas eles operam. As informações de vinculação incluem os requisitos do cabeçalho do host que definem as configurações de roteamento com base no domínio.

Aplicações

Os caminhos do aplicativo mostram as estruturas do aplicativo raiz e aninhado na configuração do IIS. As atribuições do pool de aplicativos indicam como seus aplicativos são isolados uns dos outros para fins de segurança e gerenciamento de recursos.

Diretórios virtuais

Os mapeamentos de caminhos físicos indicam onde seu conteúdo reside no sistema de arquivos. As configurações de autenticação mostram requisitos de acesso especiais que precisam ser mantidos após a migração.

Preparar-se para migração

Com uma compreensão do seu ambiente, certifique-se de que seu servidor atenda aos pré-requisitos. Primeiro, verifique sua versão do IIS com o seguinte PowerShell comando:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\InetStp\" -Name
 MajorVersion
```

Você precisa do IIS 7.0 ou posterior. A ferramenta de migração usa o Web Deploy 3.6 para empacotar seus aplicativos. Verifique sua instalação com o seguinte comando:

```
PS C:\migrations_workspace> Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\IIS Extensions
\MSDeploy\3" -Name InstallPath
```

Se o Web Deploy não estiver instalado em seu servidor, você poderá baixá-lo na página de download do Microsoft Web Platform Installer.

Sua primeira migração

Vamos começar com uma migração básica do site padrão. O exemplo a seguir mostra o comando mais simples,eb migrate.

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

Esse comando inicia uma série de etapas automatizadas, mostradas no seguinte exemplo de saída:

```
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
```

A ferramenta de migração cria um diretório estruturado contendo seus artefatos de implantação. A lista a seguir mostra a estrutura do diretório:

```
C:\migration_workspace\
### .\migrations\latest\
    ### upload_target\
        ### DefaultWebSite.zip
        ### aws-windows-deployment-manifest.json
        ### ebmigrateScripts\
            ### site_installer.ps1
            ### permission_handler.ps1
            ### >other helper scripts<
```

Controle da migração

Para ter mais controle sobre o processo de migração, você pode especificar exatamente quais sites migrar com o seguinte comando:

```
PS C:\migrations_workspace> eb migrate --sites "Default Web Site,Intranet"
```

Você também pode personalizar o nome do ambiente e o nome do aplicativo, conforme mostrado no seguinte exemplo de comando:

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

```
    --sites "Default Web Site" `
    --application-name "CorporateApp" `
    --environment-name "Production"
```

Para obter uma lista completa das opções, consultethe section called “eb migrate”.

Monitorar o progresso

Durante a migração, eb migrate fornece atualizações de status em tempo real. Consulte o seguinte exemplo de saída:

```
...
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
```

Verificando a migração

Quando o ambiente estiver pronto, o Elastic Beanstalk fornece várias maneiras de verificar sua implantação.

Acesse seu aplicativo

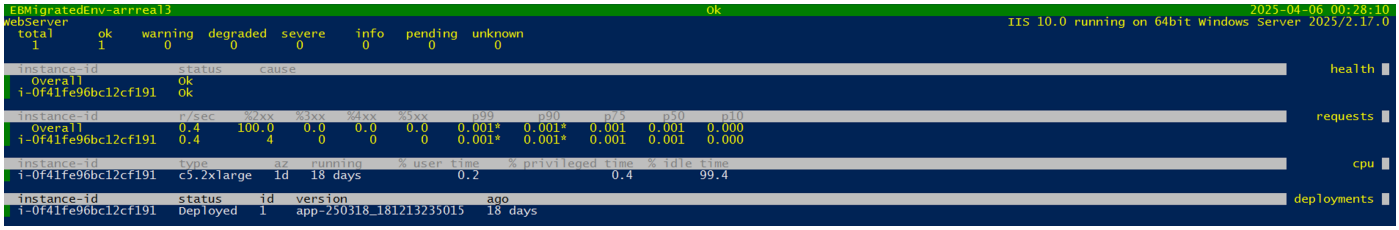
Abra o URL do aplicativo (CNAME) em um navegador da Web para verificar se ele está funcionando corretamente.

Verifique a integridade do ambiente

Use o eb health comando para visualizar a integridade do seu ambiente.

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

A imagem da tela a seguir mostra a integridade da instância, as métricas de resposta do aplicativo e a utilização dos recursos do sistema.



Use o eb logs comando para acessar os registros e solucionar qualquer problema:

```
PS C:\migrations_workspace> eb logs --zip
```

O eb logs comando baixa os registros para o `.elasticbeanstalk/logs` diretório. Para obter mais informações, consulte the section called “CloudWatch Registros”.

Conecte-se às instâncias

Se você especificou um key pair durante a migração, você pode se conectar às suas instâncias usando o RDP para solucionar problemas diretamente.

Acesse o console do Elastic Beanstalk

Você pode visualizar a integridade, os registros e as propriedades de configuração do ambiente por meio do console de gerenciamento do ambiente desse ambiente.

Gerenciando artefatos de migração

O eb migrate comando cria artefatos locais durante o processo de migração. Esses artefatos contêm informações confidenciais e podem consumir um espaço significativo em disco ao longo do tempo. Use o cleanup subcomando para gerenciar esses artefatos, conforme mostrado no exemplo a seguir:

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

Para forçar a limpeza sem confirmação, use a --force opção:

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

O processo de limpeza preserva a migração bem-sucedida mais recente no `./migrations/latest` diretório e remove os diretórios de migração mais antigos.

Configuração de rede e configurações de porta

Esta seção aborda as opções de configuração de rede para migrações do IIS, incluindo configurações de VPC, configurações de portas e implantações em vários sites.

Configuração de VPC

O eb migrate comando fornece opções flexíveis de configuração de VPC para seu ambiente do Elastic Beanstalk. A ferramenta pode detectar configurações de VPC de uma EC2 instância de origem ou aceitar configurações personalizadas de VPC por meio de parâmetros de linha de comando. Analise Usar o Elastic Beanstalk com Amazon VPC para entender como configurar o Elastic Beanstalk com a VPC.

Detecção automática de VPC

Quando eb migrate executado em uma EC2 instância, ele descobre e usa automaticamente a configuração de VPC das instâncias do ambiente EC2 de origem. O exemplo de saída a seguir ilustra as informações de configuração que ele detecta:

```
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
...
```

A configuração detectada inclui:

- Identificador VPC
- Configurações de atribuição de IP público
- Esquema de balanceamento de carga (público/privado)
- EC2 atribuições de sub-rede de instância
- Associações de grupos de segurança
- Atribuições de sub-rede do balanceador de carga

Hosts locais ou fora AWS da nuvem

Quando eb migrate executado a partir de um servidor local ou de um host fora da AWS nuvem, o serviço Elastic Beanstalk usará a VPC padrão em sua conta. AWS A lista a seguir mostra um exemplo de comando e saída:

```
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.
...
```

Analise Usar o Elastic Beanstalk com Amazon VPC para entender como o Elastic Beanstalk configura a VPC padrão para seu ambiente.

Configuração de VPC personalizada

Para qualquer ambiente de origem (EC2local ou não na AWS nuvem) em que você precise de configurações específicas de VPC, forneça um arquivo de configuração de VPC como o mostrado no exemplo a seguir:

```
{
    "id": "vpc-12345678",
    "publicip": "true",
    "elbscheme": "public",
    "ec2subnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"],
```

```
    "securitygroups": "sg-123456,sg-789012",
    "elbsubnets": ["subnet-a1b2c3d4", "subnet-e5f6g7h8"]
}
```

Aplique essa configuração usando o seguinte comando:

```
PS C:\migrations_workspace> eb migrate --vpc-config vpc-config.json
```

> **Note**
>
> O arquivo de configuração da VPC exige o `id` campo que especifica a ID da VPC. Todos os outros campos são opcionais, e o Elastic Beanstalk usará valores padrão para qualquer campo que você não especificar.

> **Important**
>
> A migração ignorará todas as configurações de VPC existentes do ambiente de origem quando você especificar o *`--vpc-config`* parâmetro. Quando você usa esse parâmetro, a migração usará somente as configurações de VPC especificadas no arquivo de configuração que você está transmitindo. O uso desse parâmetro substitui o comportamento padrão de descobrir a configuração da VPC da instância de origem ou usar a VPC padrão.

Use o `--vpc-config` parâmetro nos seguintes cenários:

- Quando você migra EC2 ambientes que não têm configurações de VPC detectáveis
- Quando você migra para uma VPC diferente daquela usada pelo ambiente de origem
- Quando você precisa personalizar seleções de sub-rede ou configurações de grupos de segurança
- Quando a descoberta automática não identifica corretamente as configurações de VPC desejadas
- Quando você migra do local e não quer usar a VPC padrão

Configuração de segurança de rede

Por padrão, eb migrate abre a porta 80 nas instâncias de destino, mas não copia outras regras do Firewall do Windows da máquina de origem. Para incluir todas as configurações de firewall, use o seguinte comando:

```
PS C:\migrations_workspace> eb migrate --copy-firewall-config
```

Esse comando executa as seguintes ações:

- Identifica as portas usadas pelas vinculações de sites do IIS
- Recupera as regras de firewall correspondentes
- Gera PowerShell scripts para recriar regras nas instâncias de destino
- Preserva todas as regras DENY para a porta 80 da máquina de origem (caso contrário, a porta 80 é permitida por padrão)

Considere um caso de uso em que sua máquina de origem tenha as regras de firewall especificadas no exemplo a seguir:

```
# 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
```

A migração cria um script (`modify_firewall_config.ps1`) que contém a seguinte configuração:

```
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
```

A ferramenta de migração executa automaticamente as seguintes ações:

- Extrai portas HTTP/HTTPS de todas as ligações de sites do IIS
- Usa a interface do Firewall do Windows INetFwPolicy2 para enumerar as regras de firewall
- Filtra as regras para incluir somente aquelas que fazem referência explícita às portas especificadas
- Processa somente associações de sites HTTP e HTTPS e suas regras de firewall associadas
- Preserva as propriedades da regra, incluindo nome de exibição, ação, protocolo e estado ativado
- Lida com portas individuais e intervalos de portas nas regras de firewall
- Adiciona o script de configuração do firewall ao manifesto de implantação

Configuração do balanceador de carga

Você pode especificar a configuração do Load Balancer por meio do `--vpc-config` argumento. Os exemplos a seguir demonstram os parâmetros.

Seleção do esquema

Escolha entre esquemas de balanceador de carga públicos e privados:

```
{
    "id": "vpc-12345678",
    "elbscheme": "private",
    "elbsubnets": ["subnet-private1", "subnet-private2"]
}
```

Distribuição de sub-rede

Para obter alta disponibilidade, distribua sub-redes do balanceador de carga entre as zonas de disponibilidade:

```
{
    "elbsubnets": [
        "subnet-az1", // Availability Zone 1
        "subnet-az2", // Availability Zone 2
        "subnet-az3"  // Availability Zone 3
    ]
}
```

> **Note**
>
> Embora o Elastic Beanstalk ofereça suporte à criação de ambientes com Application Load Balancers, Network Load Balancers e Classic Load Balancers, o comando só oferece suporte a Application Load Balancerseb migrate. Para obter mais informações sobre os tipos de balanceador de carga, consulte Balanceador de carga do ambiente do Elastic Beanstalk.

Implantações em vários locais com configurações de portas

O eb migrate comando manipula implantações complexas do IIS em vários sites, nas quais os aplicativos podem compartilhar dependências ou usar portas não padrão. Considere o exemplo a seguir de uma configuração corporativa típica com vários sites:

```
<!-- 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>
```

Para migrar essa configuração, use o comando e os parâmetros de exemplo a seguir:

```
PS C:\migrations_workspace> eb migrate `
    --sites "Default Web Site,InternalAPI,ReportingPortal" `
    --copy-firewall-config `
    --instance-type "c5.large"
```

O eb migrate comando cria um pacote de implantação que preserva a identidade e a configuração de cada site. O comando gera um `aws-windows-deployment-manifest.json` que define como esses sites devem ser implantados. O exemplo a seguir demonstra um arquivo json gerado:

```
{
    "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"
                    }
                }
            }
        ]
    }
}
```

O processo de migração cria as seguintes regras de ouvinte do Application Load Balancer que mantêm sua lógica de roteamento original:

- Rotas de tráfego da porta 80 para o site padrão
- Rotas de tráfego da porta 8081 para a API interna
- Rotas de tráfego da porta 8082 para ReportingPortal

Configuração e dependências compartilhadas

Quando os sites compartilham configurações ou dependências, eb migrate gerencia esses relacionamentos adequadamente. Consulte o exemplo a seguir em que vários sites compartilham uma configuração comum:

```
<!-- Shared configuration in applicationHost.config -->
<location path="Default Web Site">
    <system.webServer>
        <asp enableSessionState="true" />
        <caching enabled="true" enableKernelCache="true" />
    </system.webServer>
</location>
```

O processo de migração conclui as seguintes tarefas:

1. Identifica configurações compartilhadas entre sites
2. Gera PowerShell scripts apropriados para aplicar essas configurações
3. Mantém a hierarquia e a herança da configuração

Práticas recomendadas

Recomendamos que você siga as melhores práticas para a configuração de rede do seu aplicativo migrado. Os agrupamentos a seguir fornecem diretrizes resumidas.

Design em VPC

- Siga as AWS melhores práticas de design de VPC
- Use sub-redes separadas para balanceadores de carga e instâncias EC2
- Implemente tabelas de rotas adequadas e NACLs
- Considere os endpoints de VPC para serviços AWS

Alta disponibilidade

- Implantar em diversas zonas de disponibilidade

- Use pelo menos duas sub-redes para balanceadores de carga
- Configure o auto-scaling em AZs
- Implemente verificações de saúde adequadas

Segurança

- Siga as melhores práticas de segurança
- Use grupos de segurança como controle de acesso primário
- Implemente listas de controle de acesso à rede (ACLs) para segurança adicional
- Monitore os registros de fluxo da VPC

Solução de problemas

Os problemas comuns de configuração de rede incluem as seguintes áreas. Seguindo cada assunto, há exemplos de comandos para obter mais informações sobre a configuração da rede e a integridade do seu ambiente.

Configuração de sub-rede

```
# 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'
```

Acesso ao grupo de segurança

```
# 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
```

Integridade do balanceador de carga

```
# 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
```

Configurações de segurança e funções do IAM

O eb migrate comando gerencia as configurações AWS de segurança por meio de funções do IAM, perfis de instância e funções de serviço. A compreensão desses componentes garante o controle de acesso adequado e a conformidade de segurança durante a migração.

Configuração do perfil da instância

Um perfil de instância serve como um contêiner para uma função do IAM que o Elastic Beanstalk EC2 anexa às instâncias em seu ambiente. Ao executareb migrate, você pode especificar um perfil de instância personalizado:

```
PS C:\migrations_workspace> eb migrate --instance-profile "CustomInstanceProfile"
```

Se você não especificar um perfil de instância, eb migrate crie um perfil padrão com as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::elasticbeanstalk-*",
                "arn:aws:s3:::elasticbeanstalk-*/*"
            ]
        }
    ]
}
```

Gerenciamento de funções de serviço

Uma função de serviço permite que o Elastic Beanstalk gerencie recursos em AWS seu nome. Especifique uma função de serviço personalizada durante a migração com o seguinte comando:

```
PS C:\migrations_workspace> eb migrate --service-role "CustomServiceRole"
```

Se não for especificado, eb migrate cria uma função de serviço padrão nomeada `aws-elasticbeanstalk-service-role` com uma política de confiança que permite que o Elastic Beanstalk assuma a função. Essa função de serviço é essencial para que o Elastic Beanstalk monitore a integridade do seu ambiente e realize atualizações gerenciadas da plataforma. A função de serviço exige duas políticas gerenciadas:

- `AWSElasticBeanstalkEnhancedHealth`- Permite que o Elastic Beanstalk monitore a integridade da instância e do ambiente usando o sistema aprimorado de relatórios de integridade
- `AWSElasticBeanstalkManagedUpdates`- Permite que o Elastic Beanstalk realize atualizações gerenciadas da plataforma, incluindo a atualização dos recursos do ambiente quando uma nova versão da plataforma estiver disponível

Com essas políticas, a função de serviço tem permissões para:

- Crie e gerencie grupos de Auto Scaling
- Crie e gerencie balanceadores de carga de aplicativos
- Carregar registros para a Amazon CloudWatch
- Gerenciar EC2 instâncias

Para obter mais informações sobre funções de serviço, consulte o Guia do Função de serviço do Elastic Beanstalk Desenvolvedor do Elastic Beanstalk.

Configuração do security group

O eb migrate comando configura automaticamente os grupos de segurança com base nas ligações do site do IIS. Por exemplo, se seu ambiente de origem tiver sites usando as portas 80, 443 e 8081, os seguintes resultados de configuração:

```
<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>
```

O processo de migração conclui as seguintes ações:

- Cria um grupo de segurança de balanceador de carga que permite tráfego de entrada nas portas 80 e 443 da Internet (0.0.0.0/0)
- Cria um grupo EC2 de segurança que permite o tráfego do balanceador de carga
- Configura portas adicionais (como 8081) se for especificado `--copy-firewall-config`

Por padrão, o Application Load Balancer é configurado com acesso público pela Internet. Se precisar personalizar esse comportamento, como restringir o acesso a intervalos de IP específicos ou usar um balanceador de carga privado, você pode substituir a configuração padrão da VPC e do grupo de segurança usando o parâmetro: `--vpc-config`

```
PS C:\migrations_workspace> eb migrate --vpc-config vpc-config.json
```

Por exemplo, a `vpc-config.json` configuração a seguir cria um balanceador de carga privado em uma sub-rede privada:

```
{
    "id": "vpc-12345678",
    "publicip": "false",
    "elbscheme": "internal",
    "ec2subnets": ["subnet-private1", "subnet-private2"],
    "elbsubnets": ["subnet-private1", "subnet-private2"]
}
```

Para obter mais informações sobre as opções de configuração de VPC, consulte. Configuração de VPC

Integração de certificados SSL

Ao migrar sites com ligações HTTPS, integre certificados SSL por meio AWS Certificate Manager do (ACM):

```
PS C:\migrations_workspace> eb migrate --ssl-certificates
 "arn:aws:acm:region:account:certificate/certificate-id"
```

Essa configuração conclui as seguintes ações:

- Associa o certificado ao Application Load Balancer
- Mantém a terminação HTTPS no balanceador de carga
- Preserva a comunicação HTTP interna entre o balanceador de carga e as instâncias EC2

Autenticação do Windows

Para aplicativos que usam a Autenticação do Windows, eb migrate preserva as configurações de autenticação no aplicativo da `web.config` seguinte forma:

```
<configuration>
    <system.webServer>
        <security>
            <authentication>
                <windowsAuthentication enabled="true">
                    <providers>
                        <add value="Negotiate" />
                        <add value="NTLM" />
                    </providers>
                </windowsAuthentication>
            </authentication>
        </security>
    </system.webServer>
</configuration>
```

> **Important**
>
> O eb migrate comando não copia perfis ou contas de usuário do seu ambiente de origem para as instâncias de destino do Elastic Beanstalk. Todas as contas de usuário ou grupos personalizados que você criou no servidor de origem precisarão ser recriados no ambiente de destino após a migração.

Contas `IUSR` e grupos internos do Windows`IIS_IUSRS`, assim como todas as outras contas e grupos internos, são incluídos por padrão nas instâncias de destino do Windows Server. Para obter

mais informações sobre contas e grupos internos do IIS, consulte Compreendendo contas de usuário e grupo incorporadas no IIS na documentação da Microsoft.

Se seu aplicativo depende de contas de usuário personalizadas do Windows ou da integração com o Active Directory, você precisará configurar esses aspectos separadamente após a conclusão da migração.

Melhores práticas e solução de problemas

Gerenciamento de perfis

Implemente as melhores práticas AWS do IAM ao gerenciar funções em seus ambientes do Elastic Beanstalk:

Criação e gerenciamento de funções

- Crie funções usando políticas AWS gerenciadas sempre que possível
- Siga as melhores práticas de segurança do IAM
- Use o AWS Policy Generator para políticas personalizadas
- Implemente limites de permissão para segurança adicional

Monitoramento e auditoria

Habilite AWS CloudTrail para monitorar o uso da função:

- Siga o Guia AWS CloudTrail do usuário
- Configure a integração de CloudWatch registros para monitoramento em tempo real
- Configure alertas para chamadas de API não autorizadas

Processo de revisão regular

Estabeleça um ciclo de revisão trimestral para realizar as seguintes tarefas:

- Audite permissões não utilizadas usando o IAM Access Analyzer
- Remover permissões desatualizadas
- Atualize funções com base nos princípios de privilégios mínimos

Gerenciamento de certificados

Implemente essas práticas para certificados SSL/TLS em seus ambientes do Elastic Beanstalk:

Ciclo de vida do certificado

- Use AWS Certificate Managerpara gerenciamento de certificados
- Ativar a renovação automática para certificados emitidos pelo ACM
- Configurar notificações de expiração

Padrões de segurança

- Use o TLS 1.2 ou posterior
- Siga as políticas AWS de segurança para ouvintes HTTPS
- Implemente o HTTP Strict Transport Security (HSTS), se necessário

Gerenciamento de grupos de segurança

Implemente essas melhores práticas do grupo de segurança:

Gerenciamento de regras

- Documente todos os requisitos de porta personalizados
- Use registros de fluxo de VPC para monitorar o tráfego
- Use regras de referência de grupos de segurança em vez de intervalos de IP sempre que possível

Auditoria regular

Estabeleça avaliações mensais para realizar as seguintes tarefas:

- Identifique e remova regras não utilizadas
- Valide os requisitos de origem/destino
- Verifique se há regras sobrepostas

Registro em log e monitoramento

Para um monitoramento de segurança eficaz, configure os seguintes registros:

Registros de eventos do Windows em EC2 instâncias

```
# 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 Integração de registros

Configure o agente de CloudWatch registros para transmitir registros de eventos do Windows CloudWatch para monitoramento e alertas centralizados.

Para problemas persistentes, reúna esses registros e entre em contato AWS Support com as seguintes informações:

- ID do ambiente
- ID de implantação (se aplicável)
- Mensagens de erro relevantes
- Cronograma das mudanças de segurança

Entendendo o mapeamento de migração do IIS para o Elastic Beanstalk

A migração do IIS para o Elastic Beanstalk envolve o mapeamento da configuração do servidor Windows local para os recursos da nuvem. AWS Compreender esse mapeamento é crucial para migrações bem-sucedidas e gerenciamento pós-migração.

Sites e aplicativos do IIS no Elastic Beanstalk

No IIS, um site representa uma coleção de aplicativos da Web e diretórios virtuais, cada um com sua própria configuração e conteúdo. Ao migrar para o Elastic Beanstalk, esses componentes são transformados da seguinte forma:

Sites do IIS

Seus sites do IIS se tornam aplicativos dentro do Elastic Beanstalk. A configuração de cada site, incluindo suas vinculações, pools de aplicativos e configurações de autenticação, é preservada por meio do manifesto de implantação () do Elastic Beanstalk. `aws-windows-deployment-manifest.json`

Por exemplo, se você tiver vários sites, como o site padrão IntranetSite, eb migrate empacota o conteúdo e a configuração de cada site, mantendo seu isolamento.

O comando cria regras de ouvinte apropriadas do Application Load Balancer (ALB) para lidar com solicitações de roteamento para seus aplicativos. Ele também configura grupos de segurança para garantir o acesso adequado às portas com base nas ligações originais do IIS.

Pools de aplicativos

Os pools de aplicativos do IIS fornecem isolamento de processos de trabalho, gerenciamento de tempo de execução e recursos de reciclagem para seus aplicativos. No Elastic Beanstalk, eles são mapeados para processos de ambiente definidos `aws:elasticbeanstalk:environment:process` por meio do namespace e configurados via IIS nas instâncias. EC2

A migração preserva as configurações críticas do pool de aplicativos, incluindo as seguintes:

- Configurações do modelo de processo - Identidade (ApplicationPoolIdentity, NetworkService, ou contas personalizadas), configurações de tempo limite de inatividade e intervalos de reciclagem do processo
- Configurações da versão do.NET CLR - mantém sua versão especificada do.NET Framework (v2.0, v4.0 ou Sem código gerenciado) para garantir a compatibilidade do aplicativo
- Modo de pipeline gerenciado - preserva as configurações do modo de pipeline integrado ou clássico para manter sua arquitetura de processamento de solicitações HTTP
- Configurações avançadas - comprimento da fila, limites de CPU, limites de proteção contra falhas rápidas e limites de tempo de inicialização

O eb migrate comando preserva os mapeamentos entre sites e pools de aplicativos durante a migração para o ambiente do Elastic Beanstalk.

Se seus pools de aplicativos usarem cronogramas de reciclagem personalizados (horários específicos ou limites de memória), eles serão implementados por meio de PowerShell scripts no pacote de implantação que definem as configurações apropriadas do IIS nas EC2 instâncias.

Vinculações de sites

As vinculações de sites do IIS, que definem como os clientes acessam seus aplicativos, são transformadas nas seguintes configurações do Application Load Balancer (ALB):

- As ligações de porta são mapeadas de acordo com as regras de ouvinte ALB correspondentes
- As configurações do cabeçalho do host são traduzidas em regras de roteamento ALB
- Sites habilitados para SSL usam o AWS Certificate Manager (ACM) para gerenciamento de certificados

Gerenciamento virtual de diretórios e caminhos de aplicativos

Os diretórios e aplicativos virtuais do IIS fornecem mapeamento de caminhos de URL para diretórios físicos. O Elastic Beanstalk mantém esses relacionamentos por meio das seguintes construções:

Diretórios virtuais

O processo de migração preserva os caminhos físicos de seus diretórios virtuais no pacote de implantação.

Os mapeamentos de caminho são configurados na configuração do IIS nas EC2 instâncias, garantindo que sua estrutura de URL permaneça intacta após a migração.

Caminhos físicos que não são do sistema

> **Important**
>
> Por padrão, os ambientes Windows do Elastic Beanstalk provisionam somente a unidade C:\ (volume raiz). Na versão atual, aplicativos com conteúdo em unidades que não são do sistema (D:\, E:\ etc.) não têm suporte para migração.

O eb migrate comando detecta automaticamente caminhos físicos localizados em unidades que não são do sistema e avisa sobre possíveis problemas, como no exemplo a seguir:

```
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 seu aplicativo tiver dependências em unidades que não sejam do sistema, você precisará modificá-lo para armazenar todo o conteúdo na unidade C:\ antes da migração.

Aplicativos aninhados

Os aplicativos aninhados em sites são implantados com as configurações de caminho corretas e as atribuições apropriadas do pool de aplicativos. O processo de migração preserva todas

as `web.config` configurações, garantindo que as configurações específicas do aplicativo continuem funcionando conforme o esperado no ambiente de nuvem.

Reescrita de URL e roteamento de solicitação de aplicativo (ARR)

Se sua implantação do IIS usa Regravação de URL ou Roteamento de Solicitação de Aplicativo (ARR), eb migrate manipule essas configurações por meio das seguintes regras e configurações:

Regras de reescrita de URL

As regras de regravação de URL de seus `web.config` arquivos são traduzidas em regras de roteamento ALB sempre que possível. Por exemplo, a entrada a seguir se torna uma regra de ouvinte do ALB que direciona o tráfego com base nos cabeçalhos do host e nos padrões de caminho. :

```
<!-- 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>
```

Roteamento de solicitações de aplicativos

As configurações de ARR são preservadas por meio da instalação de recursos de ARR nas instâncias. EC2 O processo de migração conclui as seguintes tarefas:

- Define as configurações de proxy para corresponder ao seu ambiente de origem
- Mantém as regras de regravação de URL associadas ao ARR

Estrutura do artefato de migração

Quando você executaeb migrate, ele cria um diretório estruturado contendo todos os componentes de implantação necessários. A lista a seguir descreve a estrutura de diretórios:

```
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
```

O `aws-windows-deployment-manifest.json` arquivo é o arquivo de configuração principal que instrui o Elastic Beanstalk a implantar seus aplicativos. Consulte o exemplo de estrutura a seguir:

```
{
    "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"
                    }
                }
            }
        ]
    }
}
```

Esse manifesto garante os seguintes resultados para sua migração:

- Os aplicativos são implantados para corrigir os caminhos do IIS
- Configurações personalizadas são aplicadas
- As configurações específicas do site são preservadas
- A ordem de implantação é mantida

Cenários avançados de migração

Esta seção aborda cenários avançados de migração para implantações complexas do IIS.

Migrações de vários sites com Roteamento de Solicitação de Aplicativos (ARR)

O eb migrate comando detecta e preserva automaticamente as configurações de ARR durante a migração. Ao identificar as configurações de ARR em seu IIS`applicationHost.config`, ele gera os PowerShell scripts necessários para reinstalar e configurar o ARR nas instâncias de destino. EC2

Detecção de configuração ARR

O processo de migração examina três seções principais de configuração no IIS:

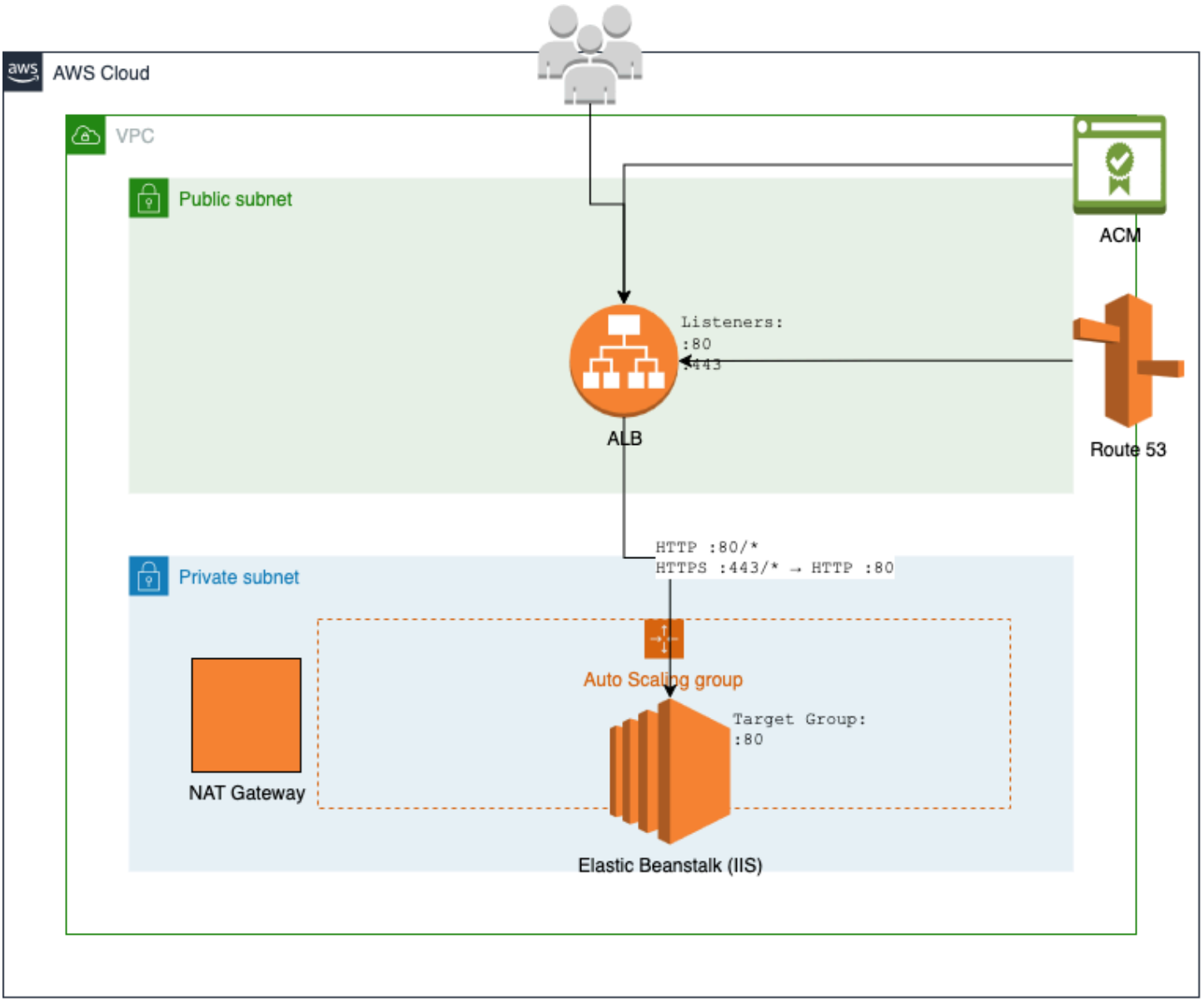
- `system.webServer/proxy`: configurações principais do proxy ARR
- `system.webServer/rewrite`: regras de reescrita de URL
- `system.webServer/caching`: Configuração de cache

Por exemplo, considere uma configuração ARR comum em que um proxy `RouterSite` executado na porta 80 solicita e é `AdminPortal` executado nas portas 8081 `APIService` e 8082, respectivamente:

```
<!-- 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>
```

O diagrama a seguir mostra como essas regras estão ocultas atrás da porta 80 no servidor IIS e não são expostas por meio dos grupos EC2 de segurança. Somente a porta 80 pode ser acessada pelo Application Load Balancer e todo o tráfego dela é roteado para o grupo-alvo na porta 80.



O comando a seguir pode migrar essa configuração:

```
PS C:\migrations_workspace> eb migrate --sites "RouterSite,APIService,AdminPortal" `
    --copy-firewall-config
```

Processo de migração do ARR

O processo de migração preserva sua configuração de ARR por meio de várias etapas.

Exportação de configuração

A ferramenta exporta suas configurações ARR existentes das três seções principais de configuração para arquivos XML separados armazenados no `ebmigrateScripts` diretório:

```
ebmigrateScripts\
### arr_config_proxy.xml
### arr_config_rewrite.xml
### arr_config_caching.xml
```

Scripts de instalação

Dois PowerShell scripts são gerados para lidar com a configuração do ARR:

1. `arr_msi_installer.ps1`: baixa e instala o módulo ARR
2. `arr_configuration_importer_script.ps1`: Importa sua configuração ARR exportada

Integração do manifesto de implantação

Os scripts são integrados ao processo de implantação por meio de entradas em`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"
                    }
                }
            }
        ]
    }
}
```

Integração com balanceador de carga

O processo de migração traduz suas regras de ARR em regras de ouvinte do Application Load Balancer (ALB) sempre que possível. Por exemplo, a configuração de ARR acima resulta em regras de ALB que roteiam o tráfego com base nos padrões de caminho de URL, mantendo o roteamento interno nas instâncias. EC2

O ambiente resultante mantém sua lógica de roteamento ARR enquanto aproveita a infraestrutura elástica AWS da empresa. Seus aplicativos continuam funcionando como antes, com o ARR gerenciando o roteamento interno enquanto o Application Load Balancer gerencia a distribuição do tráfego externo.

Migrações de vários sites sem ARR usando roteamento baseado em host

Embora o Application Request Routing (ARR) seja uma abordagem comum para gerenciar vários sites no IIS, você também pode migrar implantações de vários sites diretamente para o Elastic Beanstalk sem ARR, aproveitando os recursos de roteamento baseado em host do Application Load Balancer. Essa abordagem pode reduzir a complexidade e melhorar o desempenho ao eliminar uma camada adicional de roteamento.

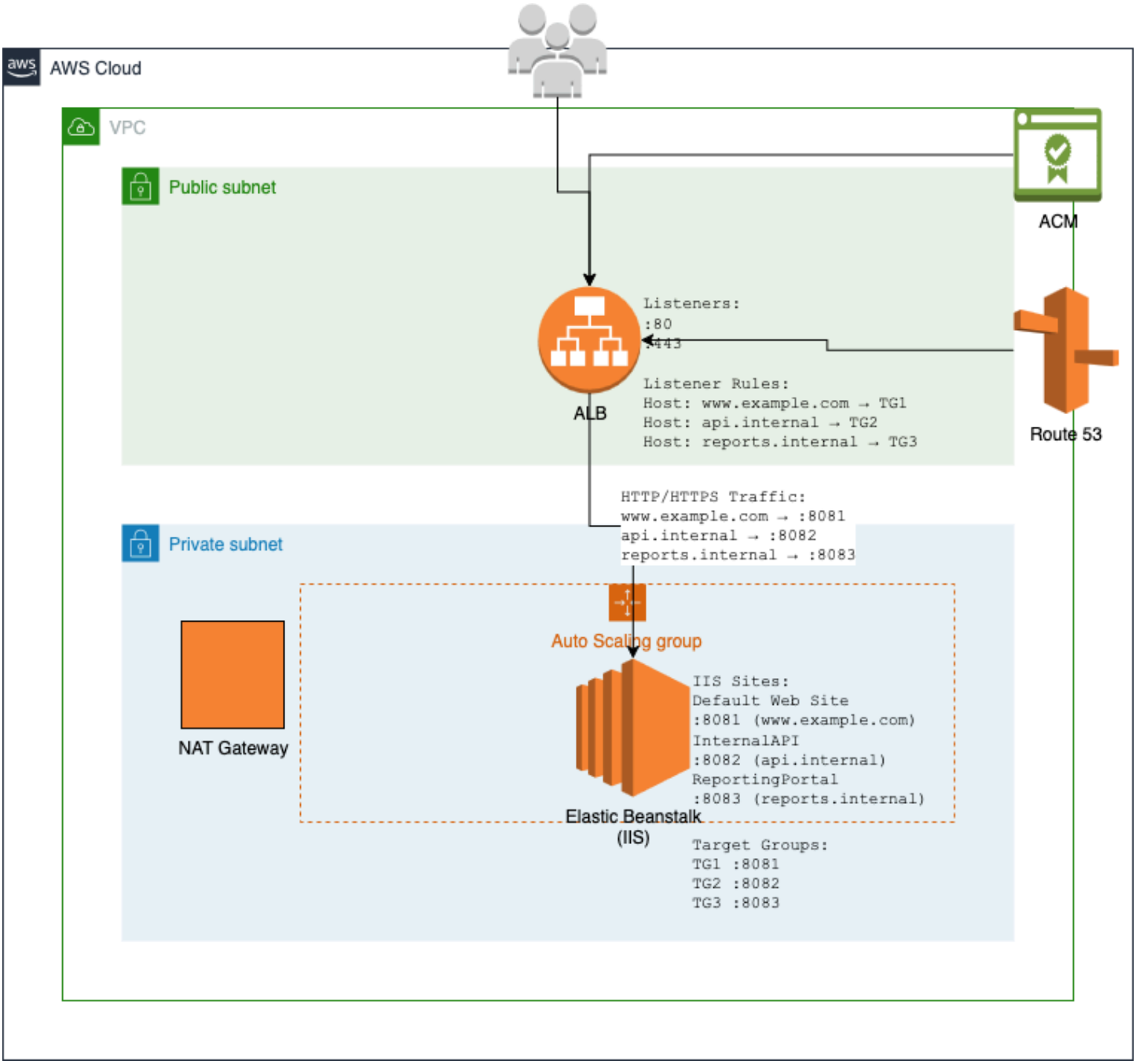
Visão geral do roteamento baseado em host

Nessa abordagem, cada site do IIS é exposto fora da EC2 instância, e o Application Load Balancer encaminha o tráfego diretamente para a porta apropriada com base no cabeçalho do host. Isso elimina a necessidade de ARR e, ao mesmo tempo, mantém a separação entre seus aplicativos.

Considere uma configuração do IIS de vários sites com três sites, cada um com sua própria associação de nome de host:

```
<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>
```

Esses sites são expostos nas portas 8081, 8082 e 8083 por meio dos EC2 grupos de segurança. O Application Load Balancer encaminha até eles com base na configuração da regra do ouvinte do Load Balancer.



Processo de migração

Para migrar essa configuração para o Elastic Beanstalk sem usar o ARReb migrate, use o comando no exemplo a seguir:

```
PS C:\migrations_workspace> eb migrate --sites "Default Web
 Site,InternalAPI,ReportingPortal"
```

O processo de migração configura automaticamente o Application Load Balancer com regras de roteamento baseadas em host que direcionam o tráfego para o grupo-alvo apropriado com base no cabeçalho do host. Cada grupo-alvo encaminha o tráfego para a porta correspondente em suas EC2 instâncias:

1. Cabeçalho do host www.example.com → Grupo de destino na porta 8081
2. Cabeçalho do host api.internal → Grupo de destino na porta 8082
3. Relatórios do cabeçalho do host. Internal → Grupo de destino na porta 8083

Configuração de SSL/TLS

Para proteger seus aplicativos com SSL/TLS, siga as seguintes etapas:

1. Solicite certificados para seus domínios por meio do AWS Certificate Manager(ACM).
2. Configure ouvintes HTTPS em seu Application Load Balancer usando esses certificados.
3. Atualize a configuração do seu ambiente para incluir ouvintes HTTPS com as seguintes opções de configuração.

```
option_settings:
  aws:elb:listener:443:
    ListenerProtocol: HTTPS
    SSLCertificateId: arn:aws:acm:region:account-id:certificate/certificate-id
    InstancePort: 80
    InstanceProtocol: HTTP
```

Com essa configuração, o encerramento do SSL ocorre no balanceador de carga e o tráfego é encaminhado para suas instâncias por HTTP. Isso simplifica o gerenciamento de certificados e, ao mesmo tempo, mantém conexões seguras com os clientes.

Práticas recomendadas

Grupos de segurança

> Configure grupos de segurança para permitir tráfego de entrada somente nas portas usadas pelos sites do IIS (8081, 8082, 8083 neste exemplo) do grupo de segurança Application Load Balancer.

Verificações de integridade

Configure verificações de saúde para cada grupo-alvo para garantir que o tráfego seja roteado somente para instâncias íntegras. Crie endpoints de verificação de integridade para cada aplicativo, caso eles ainda não existam.

Monitoramento

Configure CloudWatch alarmes para monitorar a saúde e o desempenho de cada grupo-alvo separadamente. Isso permite identificar problemas específicos de aplicativos individuais.

Escalabilidade

Considere os requisitos de recursos de todos os aplicativos ao configurar políticas de escalonamento automático. Se um aplicativo tiver necessidades de recursos significativamente diferentes, considere migrá-lo para um ambiente separado.

Gerenciamento de diretórios virtuais

O eb migrate comando preserva as estruturas de diretórios virtuais ao migrar seus aplicativos IIS para o Elastic Beanstalk.

Configuração de permissão padrão

Ao migrar diretórios virtuais, eb migrate estabelece um conjunto básico de permissões concedendo acesso a: ReadAndExecute

- IIS_IUSRS
- IUSR
- Usuários autenticados

Por exemplo, considere uma estrutura típica de diretórios virtuais:

```
<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>
```

Diretórios virtuais protegidos por senha

Quando eb migrate encontra diretórios virtuais protegidos por senha, ele emite avisos e requer intervenção manual.

O exemplo de configuração a seguir causará a resposta de aviso que segue o exemplo.

```
<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.
```

Para manter a proteção por senha, crie um script de implantação personalizado como o seguinte:

```
# 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
```

Adicione esse script à sua implantação incluindo-o no manifesto:

```
{
    "manifestVersion": 1,
    "deployments": {
        "custom": [
            {
                "name": "SecureVirtualDirectory",
                "scripts": {
                    "install": {
                        "file": "secure_vdir_config.ps1"
                    }
                }
            }
        ]
    }
}
```

Gerenciamento de permissões personalizado

O eb migrate comando fornece uma estrutura para scripts de permissão personalizados para acomodar aplicativos que exigem permissões diferentes das padrões.

```
$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"
}
```

Práticas recomendadas

Siga essas melhores práticas para planejar, executar, monitorar e verificar sua migração.

Planejamento de pré-migração

Documente as permissões e os requisitos de autenticação existentes antes da migração. Teste scripts de permissão personalizados em um ambiente de desenvolvimento antes de implantá-los na produção.

Gerenciamento de conteúdo compartilhado

Para diretórios de conteúdo compartilhado, certifique-se de que todas as permissões necessárias do sistema de arquivos estejam configuradas adequadamente por meio de scripts personalizados. Considere usar o Amazon FSx para Windows File Server para atender aos requisitos de armazenamento compartilhado.

Monitoramento e verificação

Monitore os registros do aplicativo após a migração para verificar o acesso adequado aos diretórios virtuais. Preste atenção especial às seguintes áreas:

- Acesso à identidade do pool de aplicativos
- Permissões personalizadas da conta de serviço
- Conectividade de compartilhamento de rede
- Falhas de autenticação

Configurações personalizadas do pool de aplicativos

Por padrão, o eb migrate comando não copia as configurações personalizadas do pool de aplicativos. Para preservar as configurações personalizadas do pool de aplicativos, siga este procedimento para criar e aplicar uma seção de manifesto personalizada.

1. Crie um arquivo de seus artefatos de migração.

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

2. Crie um PowerShell script personalizado para configurar grupos de aplicativos.

```
# 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. Atualize o `aws-windows-deployment-manifest.json` arquivo para incluir seu script personalizado.

```
{
    "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. Crie um ambiente com o diretório de arquivamento atualizado.

```
PS C:\migrations_workspace> eb migrate `
    --archive-dir '.\migrations\latest\upload_target\'
```

O `--archive-dir` argumento diz eb migrate para usar o código-fonte que ele criou anteriormente, evitando a criação de novos arquivos.

Implantação de versões anteriores

O eb migrate mantém um histórico de suas migrações por meio de diretórios com data e hora e versões de aplicativos no Elastic Beanstalk. Cada migração cria um arquivo zip exclusivo que pode ser implantado se necessário.

```
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
```

O link `latest` simbólico sempre aponta para o diretório de artefatos de migração criado mais recentemente. Além dos registros relevantes de aplicativos e erros, cada diretório de artefatos de migração também contém um `upload_target.zip` arquivo que você pode implantar no 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
```

Você pode implantar o `upload_target.zip` arquivo usandoeb migrate:

```
PS C:\migrations_workspace> eb migrate --zip .\migrations\latest\upload_target.zip
```

Solução de problemas e diagnósticos

Esta seção fornece orientação para solucionar problemas comuns que podem surgir durante a migração de aplicativos IIS para o Elastic Beanstalk.

Associando um EC2 par de chaves ao seu ambiente

Você pode fazer login com segurança nas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) provisionadas para seu aplicativo Elastic Beanstalk com um par de chaves da Amazon. EC2 Para obter instruções sobre como criar um par de chaves, consulte Como criar um par de chaves usando a Amazon EC2 no Guia EC2 do usuário da Amazon.

Especificar um nome de chave eb migrate tem o efeito de associar seu ambiente do Elastic Beanstalk ao par de chaves. Por motivos de segurança, isso não abrirá a porta 3389 no grupo de segurança de suas EC2 instâncias. Você pode associar grupos EC2 de segurança adicionais que permitam o tráfego na porta 3389 até eb config depois da migração inicial.

```
PS C:\migrations_workspace> eb migrate  `
    --keyname "my-keypair"  `
    --verbose
```

Quando você cria um par de chaves, a Amazon EC2 armazena uma cópia da sua chave pública. Se você não precisar mais usá-lo para se conectar a nenhuma instância do ambiente, poderá excluí-lo da Amazon EC2. Para obter detalhes, consulte Excluindo seu par de chaves no Guia do EC2 usuário da Amazon.

Para obter mais informações sobre como se conectar às EC2 instâncias do Windows Amazon, consulte Conectando-se à instância do Windows.

Acessando registros

O EB CLI fornece um eb logs recurso que você pode usar para recuperar registros de um ambiente do Elastic Beanstalk sem fazer login em suas instâncias. EC2 Após a execução doeb migrate, você pode emitir o eb logs --zip comando que baixará e salvará os registros no `.elasticbeanstalk\logs` diretório.

Como alternativa, você pode visualizar os registros por meio do console do AWS Elastic Beanstalk. Para obter mais informações, consulte Visualização de registros de EC2 instâncias da Amazon em seu ambiente do Elastic Beanstalk.

Acessando artefatos do lado do cliente

O eb migrate comando armazena registros de aplicativos e erros gerados por diretórios de artefatos de migração msdeploy internos.

```
./migrations/
### latest -> migration_20240308_123456/
### migration_20240308_123456/
    ### application.log
    ### error.log
    ### upload_target\
```

Monitorando a saúde do ambiente

O Elastic Beanstalk ajuda você a monitorar a saúde usando os recursos aprimorados de monitoramento da saúde. É um sistema automatizado de monitoramento de integridade que rastreia continuamente o status operacional das instâncias do aplicativo, aproveitando métricas integradas, como utilização da CPU, latência, contagem de solicitações e códigos de resposta.

O sistema de monitoramento de integridade utiliza uma abordagem baseada em agentes para coletar dados em nível de instância e se integra com registros e alertas em tempo real. O Elastic Load Balancing (ELB) e o Auto Scaling respondem dinamicamente às mudanças no status de saúde, garantindo alta disponibilidade e tolerância a falhas. Os modos avançados de monitoramento, incluindo relatórios de integridade aprimorados, fornecem visibilidade granular do comportamento do aplicativo, permitindo a solução proativa de problemas e mecanismos de recuperação automática.

Execute o eb health comando EB CLI para exibir a integridade do ambiente. A informação a seguir será exibida.

- Status de integridade da instância
- Métricas de resposta do aplicativo
- Utilização de recursos do sistema
- Eventos recentes de implantação

EC2 otimização de desempenho

Por padrão, eb migrate seleciona o tipo de instância c5.2xlarge para fornecer uma ótima experiência inicial com o Elastic Beanstalk. Você pode substituir esse comportamento com o --instance-type argumento:

```
PS C:\migrations_workspace> eb migrate `
    --instance-type "t3.large"
```

Para ambientes de produção, considere esses fatores ao selecionar um tipo de instância:

- Requisitos de memória de seus aplicativos
- Requisitos de CPU para processamento de cargas de trabalho
- necessidades de desempenho de rede
- Metas de otimização de custos

Configuração de volume do EBS

Por padrão, o Elastic Beanstalk criará somente um volume de dispositivo de bloco `C:\` raiz () para seu ambiente. Você pode transmitir volumes adicionais de snapshot do Amazon Elastic Block Store com a --ebs-snapshots opção:

```
PS C:\migrations_workspace> eb migrate `
    --ebs-snapshots "snap-123456789abc"
```

Para ver exemplos de como você pode configurar mapeamentos de dispositivos de blocos com o Elastic Beanstalk, consulte o artigo do blog Personalizar volumes efêmeros e do EBS em ambientes do Elastic Beanstalk.

Para aplicativos com altos requisitos de armazenamento, considere as seguintes opções:

- Usando volumes do EBS para dados persistentes
- Implementando o Amazon S3 para conteúdo estático
- Usando o Amazon FSx para Windows File Server para sistemas de arquivos compartilhados

Problemas e soluções comuns de

Evento: Falta a instalação do Web Deploy

Se você encontrar erros relacionados ao fato de o Web Deploy não ter sido encontrado, instale o Web Deploy 3.6 ou posterior a partir do Microsoft Web Platform Installer. O exemplo a seguir exibe uma possível mensagem de erro.

```
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: problemas de permissão durante a migração

Se você encontrar erros relacionados à permissão, verifique se está executando a CLI do EB com privilégios administrativos. O exemplo a seguir exibe uma possível mensagem de erro.

```
[ERROR] Access to the path 'C:\inetpub\wwwroot\web.config' is denied.
```

Evento: problemas de identidade do pool de aplicativos

Se o aplicativo falhar ao iniciar devido a problemas de identidade do pool de aplicativos, crie um script personalizado para configurar as identidades do pool de aplicativos, conforme mostrado emthe section called “Configurações personalizadas do pool de aplicativos”.

Evento: erros de configuração do certificado SSL

Se as associações HTTPS não funcionarem, verifique se você especificou um ARN de certificado ACM válido usando o parâmetro option. eb mibrate `--ssl-certificates`

Evento: Tempo limite de criação do ambiente

Se a criação do ambiente expirar, verifique se há falhas específicas na criação de recursos nos AWS CloudFormation eventos no AWS Management Console. As causas comuns incluem problemas de configuração da VPC ou limites de serviço.

Obter suporte

Se você encontrar problemas que não consegue resolver, antes de entrar em contato, AWS Support reúna as seguintes informações:

- ID do ambiente (`eb status`)
- Registros do aplicativo (`eb logs --zip`)
- Artefatos de migração de `.\migrations\latest\`
- Configuração do IIS de origem (saída de`eb migrate explore --verbose`)
- Mensagens de erro detalhadas

Para obter mais informações sobre a solução de problemas do Elastic Beanstalk, consulte. Solucionar problemas com o ambiente Elastic Beanstalk

Comparando as opções de migração: EB CLI vs. AWS Application Migration Service

AWS oferece vários caminhos para migrar aplicativos do Windows para a nuvem. Esta seção compara duas opções principais: o eb migrate comando na CLI do EB AWS Application Migration Service e (MGN). Compreender as diferenças entre essas abordagens ajudará você a escolher a estratégia de migração mais adequada às suas necessidades específicas.

Comparação das opções de migração

Recurso	CLI DA WEB () eb migrate	AWS Application Migration Service (MGN)
Foco principal	Migração em nível de aplicativo de sites e aplicativos do IIS	Nova hospedagem em nível de servidor de máquinas inteiras (servidores físicos, virtuais ou em nuvem)
Mais adequado para	Aplicativos IIS que você deseja migrar diretamente para o Elastic Beanstalk com o mínimo de reconfiguração	Migrações em grande escala envolvendo muitos servidores ou infraestrutura complexa

Recurso	CLI DA WEB () eb migrate	AWS Application Migration Service (MGN)
Abordagem de descoberta	Detecção em nível de aplicativo de sites, aplicativos e configurações do IIS	Replicação em nível de servidor de máquinas inteiras, incluindo sistema operacional e aplicativos
Ambiente de destino	Cria e configura diretamente ambientes do Elastic Beanstalk otimizados para aplicativos Windows	Cria EC2 instâncias que exigem configuração adicional para funcionar com o Elastic Beanstalk
Preservação da configuração	Preserva automaticamente as configurações específicas do IIS (sites, pools de aplicativos, vinculaçõ es)	Preserva toda a configuração do servidor, que pode incluir component es desnecessários
Modelo de implantação	Cria um ambiente limpo do Elastic Beanstalk com seus aplicativos implantados usando as melhores práticas do Elastic Beanstalk	Cria uma réplica do seu servidor de origem que pode exigir otimização para operações na nuvem
Escala de migração	Ideal para migrações direcionadas de aplicativos específicos	Projetado para migrações em grande escala de muitos servidores
Etapas de pós-migração	Mínimo; o ambiente está pronto para uso com as ferramentas de gerenciamento do Elastic Beanstalk	Requer etapas adicionais para integração com o Elastic Beanstalk, como a execução de ações de pós-lançamento do SSM

Quando usar cada opção de migração

Escolha eb migrate quando você tem os seguintes requisitos:

- Você deseja migrar aplicativos IIS específicos em vez de servidores inteiros
- Seu objetivo é adotar o Elastic Beanstalk como sua plataforma de gerenciamento de aplicativos
- Você quer aproveitar os recursos da plataforma gerenciada do Elastic Beanstalk, como fácil escalabilidade, implantação e monitoramento

- Você prefere uma implantação limpa que siga as AWS melhores práticas para operações nativas em nuvem
- Você deseja minimizar o trabalho de configuração pós-migração

Escolha AWS Application Migration Service quando você tem os seguintes requisitos:

- Você precisa migrar um grande número de servidores
- Você tem configurações de servidor complexas que devem ser preservadas com exatidão
- Seus aplicativos têm problemas de compatibilidade que exigem a manutenção do ambiente exato do servidor
- Você deseja “levantar e deslocar” com o mínimo de alterações em seus aplicativos
- Você planeja refatorar ou otimizar seus aplicativos após a migração

Comparação do fluxo de trabalho de migração

Fluxo de trabalho do EB CLI (eb migrate):

1. Instale a CLI do EB em seu servidor IIS de origem ou em um bastion host.
2. Execute eb migrate para descobrir aplicativos do IIS.
3. O comando empacota seus aplicativos e configurações.
4. Um ambiente do Elastic Beanstalk é criado com os recursos apropriados.
5. Seus aplicativos são implantados no novo ambiente.
6. Você pode gerenciar imediatamente seus aplicativos usando as ferramentas do Elastic Beanstalk.

AWS Application Migration Service fluxo de trabalho:

1. Instale o Agente AWS de Replicação nos servidores de origem.
2. Configure e teste a replicação de dados.
3. Inicie instâncias de teste para verificar a funcionalidade.
4. Agende a transição para. AWS
5. Inicie instâncias de produção.
6. Execute ações pós-lançamento para otimizar para a nuvem.
7. Se o Elastic Beanstalk for a plataforma de destino, é necessária uma configuração adicional para a integração com o Elastic Beanstalk.

Conclusão

O Elastic Beanstalk é o destino preferido para AWS aplicativos da plataforma Windows, oferecendo um ambiente gerenciado que simplifica a implantação, o dimensionamento e o gerenciamento. O eb migrate comando fornece um caminho direto para o Elastic Beanstalk para aplicativos IIS, com descoberta e configuração automáticas que preservam as configurações do seu aplicativo.

Embora AWS Application Migration Service ofereça recursos poderosos para migrações de servidores em grande escala, ele requer etapas adicionais para se integrar ao Elastic Beanstalk. Para a maioria das migrações de aplicativos do IIS, nas quais o Elastic Beanstalk é a eb migrate plataforma de destino, oferece uma abordagem mais simplificada que se alinha ao modelo de serviço gerenciado do Elastic Beanstalk.

Escolha a abordagem de migração que melhor atenda às suas necessidades específicas, considerando fatores como escala, complexidade e a arquitetura de estado final desejada. AWS

Para obter mais informações sobre AWS Application Migration Service, consulte O que é AWS Application Migration Service? no Guia do AWS Application Migration Service usuário.

Solucionar problemas com o ambiente Elastic Beanstalk

Este capítulo fornece orientações para a solução de problemas com o ambiente do Elastic Beanstalk. Ele fornece as informações a seguir.

- Uma introdução à ferramenta AWS Systems Manager, além de um procedimento para executar um runbook predefinido do Elastic Beanstalk que gera recomendações e etapas de solução de problemas.
- Orientação geral sobre ações que você poderá tomar e recursos que poderá visualizar se o status do seu ambiente se deteriorar.
- Dicas de solução de problemas mais específicas por categoria de assunto.

> **Note**
>
> Se a integridade do seu ambiente mudar para vermelho, recomendamos usar primeiro a ferramenta AWS Systems Manager , que inclui runbooks predefinidos para solucionar problemas do Elastic Beanstalk. Para obter mais informações, consulte o Usar a ferramenta Systems Manager.

Tópicos

- Usando runbooks do AWS Systems Manager Elastic Beanstalk
- Orientação geral para solução de problemas no ambiente Elastic Beanstalk
- Ambientes que acessam segredos e parâmetros com variáveis de ambiente
- Criação de ambiente e execução de instâncias
- Implantações
- Integridade
- Configuração
- Solução de problemas de contêineres do Docker
- Perguntas frequentes
- Erros comuns
- Erros de implantação

Usando runbooks do AWS Systems Manager Elastic Beanstalk

Você pode usar o Systems Manager para solucionar problemas em seus ambientes do Elastic Beanstalk. Para ajudar você a começar rapidamente, o Systems Manager fornece runbooks de automação predefinidos para o Elastic Beanstalk. Um runbook de automação é um tipo de documento do Systems Manager que define ações a serem executadas nas instâncias e em outros recursos da AWS do seu ambiente.

O documento `AWSSupport-TroubleshootElasticBeanstalk` é um runbook de automação desenvolvido para ajudar a identificar vários problemas comuns que podem degradar seu ambiente do Elastic Beanstalk. Para fazer isso, ele verifica os componentes do seu ambiente, incluindo os seguintes: EC2 instâncias, VPC, AWS CloudFormation pilha, balanceadores de carga, grupos de Auto Scaling e configuração de rede associada a regras de grupos de segurança, tabelas de rotas e. ACLs

Ele também oferece a opção de fazer upload de arquivos de log agrupados do seu ambiente para o AWS Support.

Para obter mais informações, consulte `AWSSupport-TroubleshootElasticBeanstalk` na Referência do runbook do AWS Systems Manager Automation.

Usar o Systems Manager para executar o runbook do **`AWSSupport-TroubleshootElasticBeanstalk`**

> **Note**
>
> Execute esse procedimento no mesmo Região da AWS local em que seu ambiente do Elastic Beanstalk está localizado.

1. Abra o console do AWS Systems Manager.
2. No painel de navegação, em Gerenciamento de alterações, escolha Automação.
3. Escolha Execute automation.
4. Na guia Pertencente à Amazon, na caixa de pesquisa Documentos de automação, insira `AWSSupport-TroubleshootElasticBeanstalk`.
5. Selecione o AWSSupport-TroubleshootElasticBeanstalkcartão e escolha Avançar.
6. Selecione Executar.

7. Na seção Parâmetros de entrada:

   a. No menu AutomationAssumeRolesuspenso, selecione o ARN da função que permite que o Systems Manager execute ações em seu nome.

   b. ApplicationNameEm, insira o nome do aplicativo Elastic Beanstalk.

   c. Em Nome do ambiente, insira o ambiente do Elastic Beanstalk.

   d. (Opcional) Para o S3 UploaderLink, insira um link se um AWS Support Engineer tiver fornecido um link do S3 para a coleta de registros.

8. Escolha Executar.

   Se alguma das etapas falhar, selecione o link na coluna ID da etapa da etapa que falhou. Fazer isso exibe uma página Detalhes da execução para a etapa. A VerificationErrorMessageseção exibirá um resumo das etapas que requerem atenção. Por exemplo, `IAMPermissionCheck` pode exibir uma mensagem de aviso. Nesse caso, você pode verificar se a função selecionada no AutomationAssumeRolemenu suspenso tem as permissões necessárias.

Depois que todas as etapas forem concluídas com êxito, o resultado fornecerá etapas de solução de problemas e recomendações para restaurar seu ambiente a um estado íntegro.

Orientação geral para solução de problemas no ambiente Elastic Beanstalk

As mensagens de erro podem ser exibidas na página Eventos no console, em logs ou na página Integridade. Você também pode tomar medidas para se recuperar de um ambiente degradado causado por uma alteração recente. Se a integridade do seu ambiente mudar para vermelho, tente o seguinte:

- Se uma operação em seu ambiente retornar um erro contendo o textoThe `stack` *`stack_id`* `associated with environment` *`environment-ID`* `is in` *`stack-status`* `state`, consulte Recuperando seu ambiente do Elastic Beanstalk de um estado inválido para obter ajuda para solução de problemas.
- Se uma operação em seu ambiente retornar um erro que contenha o texto`Environment` *`environment-name`* `associated CloudFormation stack` *`stack_arn`* `does not exist`, encerre seu ambiente e crie outro.
- Analise os eventos recentes do ambiente. As mensagens do Elastic Beanstalk sobre problemas de implantação, carregamento e configuração geralmente aparecem aqui.

- Analise o recente histórico de alterações do ambiente. O histórico de alterações lista todas as alterações de configuração feitas nos seus ambientes e inclui outras informações, como qual usuário do IAM fez alterações e quais parâmetros de configuração foram definidos.
- Extraia os logs para visualizar as entradas recentes do arquivo de log. Os logs do servidor Web contêm informações sobre as solicitações de entrada e os erros.
- Conecte-se a uma instância e verifique os recursos do sistema.
- Reverta para uma versão anterior válida da aplicação.
- Desfaça as alterações de configuração recentes ou restaure uma configuração salva.
- Implante um novo ambiente. Se ele parecer íntegro, execute uma troca de CNAME para rotear o tráfego para o novo ambiente e continuar depurando o anterior.

Ambientes que acessam segredos e parâmetros com variáveis de ambiente

Evento: falha na implantação da instância ao obter um ou mais segredos

Essa mensagem indica que o Elastic Beanstalk não conseguiu buscar um ou mais dos segredos especificados durante a implantação do seu aplicativo.

- Verifique se os recursos especificados pelos valores do ARN na configuração da variável de ambiente existem.
- Confirme se sua função de perfil da instância do EC2 Elastic Beanstalk tem as permissões necessárias do IAM para acessar os recursos.
- Se esse evento foi acionado por meio da `RestartAppServer` operação, depois que o problema for corrigido, tente fazer a `RestartAppServer` chamada novamente para resolver o problema.
- Se o evento foi acionado por meio de uma `UpdateEnvironment` chamada, repita a `UpdateEnvironment` operação.

Para ver exemplos desses comandos, veja AWS CLI exemplos do Elastic Beanstalk. Para obter mais informações sobre as ações da API para essas operações, consulte a Referência AWS Elastic Beanstalk da API.

Evento: a implantação da instância detectou um ou mais valores de ambiente de várias linhas, que não são compatíveis com essa plataforma

As variáveis de várias linhas não são compatíveis com as plataformas Amazon Linux 2, exceto as plataformas Docker gerenciadas pelo Docker e pelo ECS. Para ver as opções disponíveis para continuar, consulteValores de várias linhas.

Evento: CreateEnvironment falha quando um segredo é especificado

Quando uma `CreateEnvironment` falha e você tem segredos como variáveis de ambiente, você precisa resolver o problema subjacente e usá-lo `UpdateEnvironment` para concluir a configuração do ambiente. Não use`RestartAppServer`, pois não será suficiente para melhorar o meio ambiente nessa situação. Para ver exemplos desses comandos, veja AWS CLI exemplos do Elastic Beanstalk. Para obter mais informações sobre as ações da API para essas operações, consulte a Referência AWS Elastic Beanstalk da API.

Criação de ambiente e execução de instâncias

Evento: falha ao executar o ambiente

Este evento ocorre quando o Elastic Beanstalk tenta executar um ambiente e encontra falhas nesse processo. Os eventos anteriores na página Eventos irão alertá-lo para a causa raiz do problema.

Evento: operação de criação do ambiente concluída, mas com tempos limites de comando excedidos. Tente aumentar o tempo limite.

A implantação do seu aplicativo poderá levar muito tempo se você usar arquivos de configuração que executam comandos na instância, fizer download de arquivos grandes ou instalar pacotes. Aumente o tempo limite do comando para dar ao aplicativo mais tempo para iniciar a execução durante as implantações.

Evento: Falha na criação dos seguintes recursos: [AWSEBInstanceLaunchWaitCondition]

Essa mensagem indica que as EC2 instâncias da Amazon do seu ambiente não comunicaram ao Elastic Beanstalk que foram lançadas com sucesso. Isso pode ocorrer se as instâncias não têm conectividade com a Internet. Se você configurou o ambiente para executar instâncias em uma sub-rede da VPC privada, a sub-rede precisará ter um NAT para permitir que as instâncias se conectem ao Elastic Beanstalk.

Evento: um perfil de serviço é necessário nessa região. Adicione uma opção de perfil de serviço ao ambiente.

O Elastic Beanstalk usa uma função de serviço para monitorar os recursos no ambiente e oferecer suporte às atualizações gerenciadas da plataforma. Consulte Gerenciar funções de serviço do Elastic Beanstalk para obter mais informações.

Implantações

Problema: o aplicativo fica indisponível durante as implantações

Como o Elastic Beanstalk usa um processo de atualização "drop-in", pode haver alguns segundos de tempo de inatividade. Use as implantações contínuas para minimizar o efeito das implantações em seus ambientes de produção.

Evento: Falha ao criar a versão do aplicativo AWS Elastic Beanstalk

Seu pacote de origem do aplicativo pode ser muito grande ou você pode ter atingido a cota de versões do aplicativo.

Evento: operação de atualização do ambiente concluída, mas com tempos limites de comando excedidos. Tente aumentar o tempo limite.

A implantação do seu aplicativo poderá levar muito tempo se você usar arquivos de configuração que executam comandos na instância, fizer download de arquivos grandes ou instalar pacotes. Aumente o tempo limite do comando para dar ao aplicativo mais tempo para iniciar a execução durante as implantações.

Integridade

Evento: utilização da CPU excede 95%

Tente executar mais instâncias ou escolher outro tipo de instância.

Evento: Elastic Load Balancer awseb- *myapp* não tem instâncias íntegras

Se o aplicativo parece estar funcionando, verifique se o URL de verificação de integridade do aplicativo está configurado corretamente. Caso contrário, verifique os logs de ambiente e tela de integridade para obter mais informações.

Evento: Elastic Load Balancer awseb - Não pode ser encontrado *myapp*

O balanceador de carga do seu ambiente pode ter sido removido out-of-band. Somente faça alterações nos recursos do ambiente com as opções de configuração e extensibilidade fornecidas pelo Elastic Beanstalk. Reconstrua o ambiente ou inicie um novo.

Evento: falha na inicialização da EC2 instância. Aguardando a inicialização de uma nova EC2 instância...

A disponibilidade para o tipo de instância do seu ambiente pode estar baixa ou você pode ter atingido a cota de instâncias da sua conta. Verifique o painel de integridade do serviço para garantir que o serviço Elastic Compute Cloud (Amazon EC2) seja ecológico ou solicite um aumento de cota.

Configuração

Evento: A pilha *`stack_id`* associada ao ambiente *`environment-ID`* está no *`stack-status`* estado

A AWS CloudFormation pilha subjacente do seu ambiente pode estar no status *_FAILED. Esse status deve ser corrigido para continuar as operações do Elastic Beanstalk em seu ambiente. Para obter mais informações, consulte Recuperando seu ambiente do Elastic Beanstalk de um estado inválido.

Evento: você não pode configurar um ambiente do Elastic Beanstalk com valores para as ambas as opções, Elastic Load Balancing Target (Destino do balanceamento de carga elástico) e Application Healthcheck URL (URL para verificação de integridade da aplicação)

A opção `Target` no namespace `aws:elb:healthcheck` está obsoleta. Remova o namespace da opção `Target` de seu ambiente e tente atualizar novamente.

Evento: o ELB não pode ser conectado a várias sub-redes na mesma AZ

Esta mensagem poderá ser visualizada se você tentar mover um load balancer entre sub-redes na mesma zona de disponibilidade. A alteração de sub-redes no load balancer requer movê-lo para fora da(s) zona(s) de disponibilidade original(is) e, em seguida, de volta à original com as sub-redes desejadas. Durante o processo, todas as suas instâncias serão migradas entre elas AZs, causando um tempo de inatividade significativo. Em vez disso, considere criar um novo ambiente e executar uma troca CNAME.

Solução de problemas de contêineres do Docker

Evento: falha ao extrair imagem do Docker: mais recentes: nome do repositório inválido (), apenas [a-z0-9-_.] são permitidos. Siga os logs para obter mais detalhes.

Verifique a sintaxe do arquivo `dockerrun.aws.json` usando um validador JSON. Verifique também o conteúdo do dockerfile em relação aos requisitos descritos em Preparar a imagem do Docker para implantação no Elastic Beanstalk

Evento: Nenhuma diretiva EXPOSE encontrada no Dockerfile, abortar a implantação

O arquivo `Dockerfile` ou `dockerrun.aws.json` não declara a porta do contêiner. Use a instrução `EXPOSE` (`Dockerfile`) ou `Ports` bloco (arquivo `dockerrun.aws.json`) para expor uma porta do tráfego de entrada.

Evento: falha ao baixar as credenciais *`repository`* de autenticação do *`bucket name`*

O `dockerrun.aws.json` fornece um bucket and/or S3 de par de EC2 chaves inválido para o `.dockercfg` arquivo. Ou o perfil da instância não tem GetObject autorização para o bucket do S3. Verifique se o `.dockercfg` arquivo contém um bucket e um par de EC2 chaves válidos do S3. Conceda permissões da ação `s3:GetObject` à função do IAM no perfil da instância. Para detalhes, consulte Gerenciar perfis de instância do Elastic Beanstalk

Evento: Falha na execução de atividade, porque: AVISO: arquivo de configuração de autorização inválido

Seu arquivo de autenticação (`config.json`) não está formatado corretamente. Consulte Usar imagens de um repositório privado no Elastic Beanstalk

Perguntas frequentes

Pergunta: Como posso alterar o URL do meu aplicativo de myapp.us-west-2.elasticbeanstalk.com para www.myapp.com?

Em um servidor DNS, registre um registro CNAME, como **`www.mydomain.com CNAME mydomain.elasticbeanstalk.com`**.

Pergunta: Como especifico uma zona de disponibilidade exclusiva para minha aplicação do Elastic Beanstalk?

Você pode escolher uma zona de disponibilidade específica usando a CLI APIs, o plug-in Eclipse ou o plug-in do Visual Studio. Para obter instruções sobre como usar o console do Elastic Beanstalk para especificar uma zona de disponibilidade, consulte Auto Scaling de suas instâncias do ambiente do Elastic Beanstalk.

Pergunta: Como posso alterar o tipo de instância do meu ambiente?

Para alterar o tipo de instância do ambiente, acesse a página de configuração do ambiente e escolha Edit (Editar) na categoria de configuração Instances (Instâncias). Selecione um novo tipo de instância e escolha Apply (Aplicar) para atualizar o ambiente. Depois disso, o Elastic Beanstalk encerra todas as instâncias em execução e as substitui por novas.

Pergunta: Como determino se alguém fez alterações de configuração em um ambiente?

Para ver essas informações, no painel de navegação do console do Elastic Beanstalk, escolha Change history (Alterar histórico) para exibir uma lista de alterações de configuração para todos os ambientes. Essa lista inclui a data e a hora da alteração, o parâmetro de configuração e o valor para o qual ele foi alterado e o usuário do IAM que fez a alteração. Para obter mais informações, consulte Histórico de alterações.

Pergunta: posso impedir que os volumes do Amazon EBS sejam excluídos quando as instâncias são encerradas?

As instâncias do ambiente usam o Amazon EBS para armazenamento; no entanto, o volume raiz é excluído quando uma instância é encerrada pelo Auto Scaling. Nós não recomendamos que você armazene o estado ou outros dados em suas instâncias. Se necessário, você pode evitar que os volumes sejam excluídos com AWS CLI: `$ aws ec2 modify-instance-attribute -b '/dev/sdc=<vol-id>:false` conforme descrito na AWS CLI Referência.

Pergunta: como faço para excluir informações pessoais da aplicação do Elastic Beanstalk?

AWS os recursos que seu aplicativo Elastic Beanstalk usa podem armazenar informações pessoais. Quando um ambiente é encerrado, o Elastic Beanstalk encerra os recursos que ele criou. Os recursos que você adicionou usando os arquivos de configuração também são encerrados. No entanto, se você criou AWS recursos fora do seu ambiente do Elastic Beanstalk e os associou ao seu aplicativo, talvez seja necessário verificar manualmente se as informações pessoais que seu aplicativo pode ter armazenado não foram retidas. Ao longo deste guia do desenvolvedor, sempre que discutimos a criação de recursos adicionais, também mencionamos quando você deve pensar em excluí-los.

Erros comuns

Este tópico lista as mensagens de erro comuns encontradas ao usar a EB CLI e as possíveis soluções. Se você encontrar uma mensagem de erro que não aparece aqui, use os links Feedback para relatá-la.

ERRO: Ocorreu um erro ao usar o comando git. Código do erro: 128 Erro: fatal: Não é um nome de objeto válido HEAD

Causa: essa mensagem de erro aparece quando você inicializou um repositório Git, mas ainda não confirmou. A EB CLI procura a revisão HEAD quando a pasta do seu projeto contém um repositório Git.

Solução: adicione os arquivos na pasta do seu projeto à área de preparação e confirme:

```
~/my-app$ git add .
~/my-app$ git commit -m "First commit"
```

ERRO: Esta ramificação não tem um ambiente padrão. Você deve especificar um ambiente digitando “eb status my-env-name" ou definir um ambiente padrão digitando “eb use my-env-name”.

Causa: ao criar uma nova ramificação em git, ela não é anexada a um ambiente Elastic Beanstalk por padrão.

Solução: execute eb list para ver uma lista dos ambientes disponíveis. Em seguida, execute eb use ***env-name*** para usar um dos ambientes disponíveis.

ERRO: Plataformas 2.0+ exigem um perfil de serviço. Você pode fornecer uma com a opção --service-role

Causa: se você especificar um nome de ambiente com eb create (por exemplo, eb create my-env), a EB CLI não tentará criar uma função de serviço para você. Se você não tem a função de serviço padrão, o erro acima é mostrado.

Solução: execute eb create sem um nome de ambiente e siga os prompts para criar a função de serviço padrão.

Erros de implantação

Sua implantação do Elastic Beanstalk pode responder com uma resposta 404 (se o aplicativo falhar na inicialização) ou 500 (se o aplicativo falhar durante o tempo de execução). Para solucionar

muitos problemas comuns, você pode usar a CLI do EB para verificar o status da sua implantação, visualizar seus registros, obter acesso à EC2 sua instância com SSH ou abrir a página do console de gerenciamento AWS do seu ambiente de aplicativos.

Para usar a EB CLI para ajudar a solucionar problemas de implantação

1. Corra eb status para ver o status de sua implantação atual e a integridade de seus EC2 hosts. Por exemplo:

```
$ 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**
>
> O uso do switch `--verbose` fornece informações sobre o status das suas instâncias em execução. Sem ele, o eb status imprime somente informações gerais sobre seu ambiente.

2. Execute eb health para visualizar informações de integridade do seu 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
```

O exemplo acima mostra um ambiente com cinco instâncias em que a implantação da versão "v2" falhou na terceira instância. Após uma implantação com falha, a versão esperada será redefinida para a última versão bem-sucedida que, neste caso, é o "Aplicativo de exemplo" da primeira implantação. Consulte Usar a EB CLI para monitorar a integridade do ambiente para obter mais informações.

3. Execute eb logs para fazer download e visualizar os logs associados à implantação do aplicativo.

```
$ eb logs
```

4. Execute eb ssh para se conectar com a EC2 instância que está executando seu aplicativo e examine-a diretamente. Na instância, o aplicativo implantado está no diretório `/opt/python/current/app`, e seu ambiente Python está em `/opt/python/run/venv/`.
5. Execute eb console para visualizar o ambiente da aplicação no Console de Gerenciamento da AWS. Você pode usar a interface da Web para examinar facilmente diversos aspectos da implantação, incluindo configuração, status, eventos e logs do aplicativo. Você também pode fazer download das versões atuais ou anteriores do aplicativo que você implantou no servidor.

Recursos do Elastic Beanstalk

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com este serviço.

- Referência da API do Elastic Beanstalk Uma descrição abrangente de todos os SOAP e Query. APIs Além disso, ele contém uma lista de todos os tipos de dados SOAP.
- elastic-beanstalk-samples on GitHub — Um GitHub repositório com exemplos de arquivos de configuração do Elastic Beanstalk (.ebextensions). O `README.md` arquivo do repositório tem links para GitHub repositórios adicionais com aplicativos de amostra.
- Perguntas técnicas frequentes sobre o Elastic Beanstalk: as principais perguntas dos desenvolvedores sobre esse produto.
- AWS Elastic Beanstalk Notas de lançamento — Detalhes sobre novos recursos, atualizações e correções nas versões de serviço, plataforma, console e EB CLI do Elastic Beanstalk.

- Aulas e workshops — Links para cursos especializados e baseados em funções, além de laboratórios individualizados para ajudar a aprimorar suas AWS habilidades e ganhar experiência prática.
- AWS Centro do desenvolvedor — explore tutoriais, baixe ferramentas e saiba mais sobre eventos para AWS desenvolvedores.
- AWS Ferramentas do desenvolvedor — Links para ferramentas do desenvolvedor SDKs, kits de ferramentas do IDE e ferramentas de linha de comando para desenvolver e gerenciar AWS aplicativos.
- Centro de recursos de introdução — Saiba como configurar seu aplicativo Conta da AWS, participar da AWS comunidade e lançar seu primeiro aplicativo.
- Tutoriais práticos — Siga os tutoriais para iniciar seu step-by-step primeiro aplicativo no. AWS
- AWS Whitepapers — Links para uma lista abrangente de AWS white papers técnicos, abrangendo tópicos como arquitetura, segurança e economia e criados por arquitetos de AWS soluções ou outros especialistas técnicos.
- AWS Support Center — O hub para criar e gerenciar seus AWS Support casos. Também inclui links para outros recursos úteis, como fóruns, informações técnicas FAQs, status de integridade do serviço AWS Trusted Advisor e.
- Suporte— A principal página da web com informações sobre Suporte um one-on-one canal de suporte de resposta rápida para ajudá-lo a criar e executar aplicativos na nuvem.

- Entrar em contato: um ponto central de contato para consultas relativas a faturas da AWS , contas, eventos, uso abusivo e outros problemas.
- AWS Termos do site — Informações detalhadas sobre nossos direitos autorais e nossa marca registrada; sua conta, licença e acesso ao site; e outros tópicos.

Aplicações de exemplo

Veja a seguir os links de download para os aplicativos de exemplo que são implantados como parte do Saiba como começar a usar o Elastic Beanstalk.

> **Note**
>
> Alguns exemplos usam recursos que podem ter sido lançados após a versão da plataforma que você está usando. Se houver falha na execução do exemplo, tente atualizar sua plataforma para uma versão mais recente, como descrito em the section called “Plataformas compatíveis”.

- Docker: docker.zip
- Docker com vários contêineres — 2.zip docker-multicontainer-v
- Docker pré-configurado (Glassfish) — 1.zip docker-glassfish-v
- Go: go.zip
- Corretto: corretto.zip
- Tomcat: tomcat.zip
- .NET Core no Linux — dotnet-core-linux.zip
- .NET Core — dotnet-asp-windows.zip
- Node.js: nodejs.zip
- PHP: php.zip
- Python: python.zip
- Ruby: ruby.zip

AWS SDK para Java

O AWS SDK para Java fornece uma API Java que você pode usar para criar aplicativos que usam serviços de AWS infraestrutura. Com o AWS SDK para Java, você pode começar em minutos com um único pacote disponível para download que inclui a biblioteca AWS Java, exemplos de código e documentação.

AWS SDK para Java Isso requer o J2SE Development Kit. Você pode fazer download do software Java mais recente em http://developers.sun.com/downloads/. O SDK também requer pacotes de terceiros Apache Commons (Codec e Logging) e Saxon-HE, que estão incluídos no diretório de terceiros do SDK. HTTPClient

Para obter mais informações, consulte AWS SDK para Java. Para obter mais informações sobre o SDK, código de exemplo, documentação, ferramentas e recursos adicionais, consulte o Java on AWS Developer Center.

AWS SDK para .NET

Com o AWS SDK para .NET, você pode começar rapidamente com um único pacote disponível para download que inclui a AWS biblioteca.NET, exemplos de código e documentação.

Para obter mais informações, consulte AWS SDK for .NET. Para as versões suportadas do .NET Framework e do Visual Studio, consulte o Guia do Desenvolvedor AWS SDK para .NET.

Para obter mais informações sobre o SDK, código de exemplo, documentação, ferramentas e recursos adicionais, consulte o.NET no AWS Developer Center.

AWS Toolkit for Visual Studio

Com o AWS Toolkit for Visual Studio plug-in, você pode implantar um aplicativo.NET existente no Elastic Beanstalk. Você também pode criar projetos usando os AWS modelos pré-configurados com o. AWS SDK para .NET

Para obter os pré-requisitos e informações de instalação, consulte AWS Toolkit for Visual Studio. Para começar a criar a aplicação do Elastic Beanstalk usando o Visual Studio, consulte Implantar aplicações .NET Windows com o Elastic Beanstalk.

AWS SDK para JavaScript em Node.js

Com o AWS SDK do Node.js, você pode começar rapidamente com um único pacote disponível para download que inclui a biblioteca AWS Node.js, exemplos de código e documentação. JavaScript

Para obter mais informações, consulte o AWS SDK JavaScript em Node.js.

AWS SDK para PHP

Com o AWS SDK para PHP, você pode começar rapidamente com um único pacote disponível para download que inclui a biblioteca AWS PHP, exemplos de código e documentação.

Para obter mais informações, consulte o AWS SDK for PHP Para obter mais informações sobre o SDK, código de exemplo, documentação, ferramentas e recursos adicionais, consulte o PHP no AWS Developer Center.

AWS SDK for Python (Boto)

Com o AWS SDK for Python (Boto), você pode começar rapidamente com um único pacote disponível para download que inclui a biblioteca AWS Python, exemplos de código e documentação. Você pode criar aplicativos em Python com base nisso, eliminando a complexidade da codificação diretamente nas interfaces de serviços da web. APIs

A all-in-one biblioteca oferece Python amigável para desenvolvedores APIs e oculta muitas das tarefas de nível inferior associadas à programação para a AWS nuvem, incluindo autenticação, novas tentativas de solicitação e tratamento de erros. O SDK fornece exemplos práticos em Python de como usar as bibliotecas para criar aplicativos.

Para obter informações sobre o Boto, o código de exemplo, a documentação, ferramentas e recursos adicionais, consulte Python no Centro de desenvolvedores do AWS.

AWS SDK para Ruby

Com a biblioteca, os exemplos de código e a documentação do AWS Ruby, você pode criar aplicativos Ruby com base nisso, eliminando a complexidade da codificação diretamente nas interfaces de serviços da web. APIs

A all-in-one biblioteca oferece Ruby fácil de usar para desenvolvedores APIs e oculta muitas das tarefas de nível inferior associadas à programação para a AWS nuvem, incluindo autenticação,

novas tentativas de solicitação e tratamento de erros. O SDK fornece exemplos práticos em Ruby de como usar as bibliotecas para criar aplicativos.

Para obter informações sobre o SDK, o código de exemplo, a documentação, ferramentas e recursos adicionais, consulte Ruby no Centro de desenvolvedores do AWS.

Histórico do documento

A tabela a seguir descreve as alterações importantes no Guia do desenvolvedor do AWS Elastic Beanstalk desde abril de 2024.

Alteração	Descrição	Data
Tópico revisado: O que é AWS Elastic Beanstalk?	Novo diagrama de arquitetura e introdução condensada para maior clareza.	5 de junho de 2025
Tópico revisado: Configurando o EB CLI	Instruções de instalação condensadas e focadas.	5 de junho de 2025
Tópico revisado: PHP Quickstart	Foi adicionada uma etapa para atualizar e reimplant ar o aplicativo e aprimorou algumas instruções.	5 de junho de 2025
Tópico revisado: Tutorial de introdução	Simplificou o tutorial de introdução para maior clareza e agilidade na integração.	5 de junho de 2025
Conteúdo reorganizado	Estrutura de conteúdo reorganizada para destacar informações essenciais e melhorar a capacidade de descoberta.	5 de junho de 2025
Novo tópico: Gerenciando grupos EC2 de segurança	O Elastic Beanstalk adiciona suporte ao ambiente de exclusão do grupo de segurança padrão. EC2	30 de abril de 2025
Novo tópico: Migração de aplicativos IIS para o Elastic Beanstalk	O Elastic Beanstalk adiciona recursos para migrar seus aplicativos Windows IIS para o. AWS Elastic Beanstalk	18 de abril de 2025

Novo tópico: Usando o Elastic AWS Secrets ManagerAWS Systems Manager Beanstalk com um Parameter Store	O Elastic Beanstalk adiciona suporte AWS Secrets Manager para referenciar AWS Systems Manager segredos e parâmetros do Parameter Store com variáveis de ambiente.	31 de março de 2025
AWS atualizações de políticas gerenciadas para`AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy` .	Permissões atualizadas na política AWS gerenciada.	27 de fevereiro de 2025
As configurações de lançamento foram preteridas em favor dos modelos de lançamento	Este tópico foi revisado para explicar como os ambientes do Elastic Beanstalk são afetados pela descontinuação das configurações de lançamento do EC2 Amazon Auto Scaling em favor dos modelos de lançamento. Também revisamos outros conteúdos relacionados às configurações e modelos de lançamento.	23 de janeiro de 2025
Novo tópico: Usando o EC2 Fast Launch com ramificações da plataforma Windows	As versões da plataforma Windows do Elastic Beanstalk AMIs incluem EC2 base com o Fast Launch ativado.	22 de janeiro de 2025
Novo tópico: Estratégia de alocação de instâncias spot	O Elastic Beanstalk adiciona suporte à configuração da Estratégia de Alocação Spot durante a criação do ambiente.	15 de janeiro de 2025

AdministratorAccess-AWSElasticBeanstalk AWS política gerenciada	Permissões atualizadas na política AWS gerenciada.	11 de dezembro de 2024
Movimentação das recomendações do EB para ambientes de primeira onda do Graviton arm64 para o capítulo Conteúdo arquivado	Movimentação das recomendações do EB para ambientes de primeira onda do Graviton arm64 para o capítulo Conteúdo arquivado	5 de outubro de 2024
Modelos de inicialização	Novo tópico: Modelos de inicialização. Este tópico explica como o Amazon EC2 Auto Scaling está eliminando gradualmente as configurações de lançamento em favor dos modelos de lançamento.	1.º de outubro de 2024
QuickStartfor Python	Novo tópico: QuickStart para Python	24 de setembro de 2024
Permissões necessárias do bucket do Amazon S3 para políticas restritivas de endpoint de VPC	Novo tópico: Permissões de bucket do Amazon S3 necessárias para políticas restritivas de endpoint de VPC	18 de setembro de 2024
QuickStart para Java no Tomcat	Novo tópico: QuickStart para o aplicativo Java JSP executado no Tomcat	12 de setembro de 2024
QuickStart para Java	Novo tópico: QuickStart para Java	12 de setembro de 2024

Usando o Amazon EC2 Systems Manager para repositórios privados do Docker	Tópico restabelecido: Usando o Amazon EC2 Systems Manager para repositór ios privados do Docker. Substitui o tópico Usando o AWS Secrets Manager para repositórios privados do Docker até que a integraçã o do Elastic Beanstalk com o SSM seja suportada.	8 de setembro de 2024
EB CLI 2.6 (retirada) para o capítulo Conteúdo arquivado	Transferência do EB CLI 2.6 (retirado) para o capítulo Conteúdo arquivado	15 de agosto de 2024
API do EB (retirada) para o capítulo Conteúdo arquivado	Transferência da API do EB (retirada) para o capítulo Conteúdo arquivado	15 de agosto de 2024
Implantar aplicações do Elastic Beanstalk a partir de contêineres do Docker	Atualizado e reorganizado: Implantar aplicações do Elastic Beanstalk a partir de contêineres do Docker	15 de agosto de 2024
Conteúdo arquivado	Novo capítulo: Conteúdo arquivado	15 de agosto de 2024
QuickStart para Windows ASP.NET	Novo tópico: QuickStart para Windows ASP.NET	5 de julho de 2024
QuickStart para o.NET Core no Windows	Novo tópico: QuickStart para o.NET Core no Windows	28 de junho de 2024
QuickStart para Docker	Novo tópico: QuickStart para Docker	19 de junho de 2024

Impedir o acesso ao bucket do Amazon S3 entre ambientes	Novo tópico: Impedir o acesso ao bucket do Amazon S3 entre ambientes	12 de junho de 2024
QuickStart para o.NET Core no Linux	Novo tópico: QuickStart para o.NET Core no Windows	28 de maio de 2024
QuickStart para PHP	Novo tópico: QuickStart para PHP	10 de maio de 2024
QuickStart para Node.js	Novo tópico: QuickStart para Node.js	5 de maio de 2024
QuickStart para Go	Novo tópico: QuickStart for Go	5 de maio de 2024
Cronograma de lançamentos da plataforma Elastic Beanstalk	Adição de um novo tópico que inclui uma programação de Próximos lançamentos de ramificações de plataforma. Transferência de Programação de ramificações desativadas da plataforma e Histórico de ramificações de plataforma retiradas para este tópico.	1º. de maio de 2024
`AWSElasticBeanstalkRoleCore` AWS política gerenciada	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
`AWSElasticBeanstalkManagedUpdatesServiceRolePolicy` AWS política gerenciada	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024

AWSElasticBeanstalkManagedUpdatesInternalServiceRolePolicy AWS política gerenciada	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
AWSElasticBeanstalkMaintenance AWS política gerenciada	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024
AWSElasticBeanstalkInternalMaintenanceRolePolicy AWS política gerenciada	Permissões atualizadas na política AWS gerenciada.	30 de abril de 2024

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.