

Guida per l'utente

Amazon CodeCatalyst



Amazon CodeCatalyst: Guida per l'utente

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

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

Table of Contents

.....	xxiii
Che cos'è Amazon CodeCatalyst?	1
Cosa posso fare con? CodeCatalyst	1
Come posso iniziare con CodeCatalyst?	2
Scopri di più su CodeCatalyst	2
Come migrare da CodeCatalyst	3
Migrazione dei tuoi repository	3
Migrazione del repository in un CodeCatalyst repository GitLab	3
Migrazione del repository in un repository CodeCatalyst GitHub	3
Migrazione generica verso un altro provider di repository	3
Estrazione dei dati da CodeCatalyst	5
Scarica artefatti	5
Scarica gli allegati del tuo numero	6
Scarica il codice sorgente di un'azione	6
Eliminazione dei dati da CodeCatalyst	7
Richiedi al team di assistenza di eliminare i dati per tuo conto	7
Elimina il tuo CodeCatalyst spazio	7
Elimina un progetto	8
Eliminare un archivio di origine	9
Eliminare un blueprint personalizzato	10
Elimina gli allegati del tuo problema	10
Eliminazione di un file in ambienti di sviluppo a cui si accede tramite Amazon CodeCatalyst	11
Eliminazione di un ambiente di sviluppo per il tuo spazio	11
Eliminazione delle connessioni degli account	11
Rimuovere un account da uno CodeCatalyst spazio nella Console AWS di gestione	12
Eliminazione di un segreto	13
Eliminazione di un team	14
Eliminazione di una flotta predisposta	14
Eliminazione di un archivio di pacchetti	15
Concetti	16
AWS Spazi ID Builder in CodeCatalyst	17
Spazi che supportano la federazione delle identità in CodeCatalyst	17
Progetti	17

Blueprint	17
Connessioni agli account	18
Connessioni VPC	18
AWS ID del costruttore	19
Profili utente in CodeCatalyst	19
Archivi di origine	19
Commit	20
Ambienti di sviluppo	20
Flussi di lavoro	21
Azioni	21
Problemi	21
Token di accesso personali () PATs	22
Connessioni personali	22
Roles	22
Configura e accedi a CodeCatalyst	23
Creare un nuovo spazio e un nuovo ruolo di sviluppo (a partire senza invito)	25
Creazione di un nuovo spazio e ruoli IAM	26
Accettazione di un invito e creazione di un AWS Builder ID	32
Accettazione di un invito e creazione di un Builder ID AWS	33
Accesso con un AWS Builder ID	34
Dispositivi attendibili	35
Accesso con SSO	35
Visualizzazione di tutti gli spazi e i progetti per un utente	35
Visualizzazione e gestione dei CodeCatalyst profili	36
Visualizzazione del tuo CodeCatalyst profilo	37
Visualizzazione del profilo di un altro utente CodeCatalyst	37
Aggiornamento di un profilo	38
Modifica di una CodeCatalyst password associata a un AWS Builder ID	39
Configurazione per l'utilizzo AWS CLI di CodeCatalyst	40
Tutorial introduttivi	42
Tutorial: Creazione di un progetto con il moderno modello di applicazione web a tre livelli	43
Prerequisiti	45
Fase 1: Creare il moderno progetto di applicazione web a tre livelli	46
Fase 2: Invita qualcuno al tuo progetto	47
Passaggio 3: Crea problemi su cui collaborare e tenere traccia del lavoro	48
Passaggio 4: Visualizza il tuo repository di origine	48

Passaggio 5: creare un ambiente di sviluppo con un ramo di test e apportare una rapida modifica al codice	50
Passaggio 6: Visualizza il flusso di lavoro che crea l'applicazione moderna	52
Passaggio 7: chiedi agli altri di rivedere le modifiche	55
Passaggio 8: chiudere il problema	58
Pulizia delle risorse	59
Documentazione di riferimento	60
Tutorial: Iniziare con un progetto vuoto	61
Prerequisiti	62
Crea un progetto vuoto	62
Crea un repository di sorgenti	62
Crea un flusso di lavoro per creare, testare e implementare una modifica al codice	64
Invita qualcuno al tuo progetto	64
Crea problemi su cui collaborare e monitora il lavoro	65
Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa	66
Prerequisiti	67
Utilizzo di Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di funzionalità	68
Aggiungi un riepilogo generato automaticamente durante la creazione di una pull request	72
Crea un riepilogo dei commenti lasciati sulle modifiche al codice in una pull request	75
Crea un problema e assegnalo ad Amazon Q	76
Crea un problema e chiedi ad Amazon Q di eseguire delle attività consigliate per risolverlo ...	84
Eseguire la pulizia delle risorse	85
Organizza le risorse con gli spazi	86
Creare uno spazio	88
Modifica di uno spazio	91
Eliminazione di uno spazio	91
Monitoraggio dell'attività per utenti e risorse in uno spazio	92
Consentire l'accesso alle AWS risorse con connected Account AWS	93
Aggiungere un Account AWS a uno spazio	94
Aggiungere ruoli IAM alle connessioni degli account	98
Aggiungere la connessione all'account e i ruoli IAM all'ambiente di distribuzione	100
Visualizzazione delle connessioni degli account	101
Eliminazione delle connessioni all'account (in CodeCatalyst)	101
Configurazione di un account di fatturazione per uno spazio	103
Configurazione dei ruoli IAM per gli account connessi	103

Ruolo CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i>	104
Ruolo AWSRoleForCodeCatalystSupport	105
Creazione di un ruolo IAM e utilizzo della policy CodeCatalyst di fiducia	106
Concessione delle autorizzazioni di spazio agli utenti	107
Visualizzazione dei membri in uno spazio	108
Invitare un utente direttamente in uno spazio	109
Annullamento di un invito per uno spazio	110
Modifica del ruolo di un membro dello spazio	111
Rimuovere un membro dello spazio	112
Rimuovere o modificare il ruolo di un utente con il ruolo di amministratore dello spazio	112
Consentire l'accesso allo spazio tramite team	114
Creare un team	115
Visualizzazione di un team	117
Assegnazione di ruoli spaziali a un team	117
Assegnazione di ruoli di progetto a un team a livello di spazio	118
Aggiungere direttamente un utente a un team	119
Rimuovere direttamente un utente da un team	120
Aggiungere un gruppo SSO a un team	120
Eliminazione di un team	121
Consentire l'accesso allo spazio per le risorse della macchina	122
Accesso allo spazio di visualizzazione per le risorse della macchina	123
Disabilitazione dell'accesso allo spazio per le risorse della macchina	123
Consentire l'accesso allo spazio per le risorse della macchina	124
Amministrazione degli ambienti di sviluppo per uno spazio	124
Visualizzazione degli ambienti di sviluppo per il tuo spazio	126
Modifica di un ambiente di sviluppo per il tuo spazio	126
Interruzione di un ambiente di sviluppo per il tuo spazio	127
Eliminazione di un ambiente di sviluppo per il tuo spazio	127
Quote per gli spazi	128
Organizza il lavoro con i progetti	130
Creare un progetto	131
Creare un progetto vuoto	131
Creazione di un progetto con un repository collegato di terze parti	132
Creare un progetto con un blueprint	137
Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint	139

Le migliori pratiche per utilizzare i blueprint con i progetti	142
Aggiungere risorse e attività ai progetti creati	142
Ottenere un elenco di progetti	143
Visualizzazione delle attività del progetto e degli ambienti di sviluppo	143
Visualizzazione di tutti i progetti in uno spazio	144
Visualizzazione delle impostazioni del progetto	145
Passare a un progetto diverso in CodeCatalyst	145
Eliminazione di un progetto	145
Concessione agli utenti delle autorizzazioni relative al progetto	146
Ottenere un elenco dei membri e dei relativi ruoli nel progetto	146
Invitare un utente a un progetto	148
Annullamento di un invito	149
Rimuovere un utente dal progetto	149
Accettare o rifiutare un invito per un progetto	150
Consentire l'accesso al progetto tramite team	150
Aggiungere un team a un progetto	151
Assegnazione di ruoli di progetto a un team	152
Rimuovere un ruolo di progetto per un team	152
Consentire l'accesso al progetto per le risorse della macchina	153
Visualizzazione dell'accesso al progetto per le risorse della macchina	154
Disabilitazione dell'accesso al progetto per le risorse della macchina	154
Abilitazione dell'accesso al progetto per le risorse della macchina	155
Quote per progetti	156
Invio di notifiche	156
Come funzionano le notifiche?	157
Guida introduttiva alle notifiche di Slack	159
Invio di notifiche via email e su Slack	162
Configura progetti con progetti	168
Creazione di un progetto con un blueprint	169
Aggiungere un blueprint in un progetto per integrare le risorse	170
Dissociazione di un blueprint da un progetto	173
Modifica delle versioni del blueprint in un progetto	173
Modifica della descrizione di un blueprint in un progetto	176
Utilizzo della gestione del ciclo di vita come utente del blueprint	177
Utilizzo della gestione del ciclo di vita su progetti esistenti	178
Utilizzo della gestione del ciclo di vita su più blueprint in un progetto	178

Gestione dei conflitti nelle richieste pull del ciclo di vita	178
Disattivazione delle modifiche alla gestione del ciclo di vita	178
Sovrascrivere la gestione del ciclo di vita di un blueprint in un progetto	179
Creazione di un progetto completo con progetti	179
Blueprint disponibili	180
Ricerca di informazioni sul modello del progetto	183
Standardizzazione dei progetti con progetti personalizzati	183
Concetti relativi ai progetti personalizzati	185
Guida introduttiva ai progetti personalizzati	188
Tutorial: creazione e aggiornamento di un'applicazione React	194
Conversione degli archivi di origine in blueprint personalizzati	202
Lavorare con la gestione del ciclo di vita come autore di progetti	204
Sviluppo di un modello personalizzato per soddisfare i requisiti del progetto	210
Pubblicazione di un blueprint personalizzato in uno spazio	249
Impostazione delle autorizzazioni di pubblicazione per un blueprint personalizzato	254
Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali	255
Modifica delle versioni del catalogo per un blueprint personalizzato	256
Visualizzazione di dettagli, versioni e progetti di un blueprint personalizzato	256
Rimozione di un blueprint personalizzato da un catalogo di space blueprint	257
Eliminazione di un blueprint o di una versione personalizzati pubblicati	258
Aggiungere dipendenze, gestire le discrepanze e aggiornare strumenti e componenti	260
Contribuisci	263
Quote per i progetti	263
Archivia e collabora sul codice con i repository di origine	264
Concetti relativi al repository di origine	266
Progetti	17
Archivi di origine	266
Ambienti di sviluppo	20
Token di accesso personali () PATs	22
Rami	267
Rami predefiniti	268
Commit	20
Richieste pull	268
Revisioni	269
Flussi di lavoro	21
Configurazione	270

Installa Git	270
Crea un token di accesso personale	271
Guida introduttiva agli archivi di origine	271
Creazione di un progetto con un blueprint	272
Visualizzazione degli archivi di un progetto	275
Creazione di un ambiente di sviluppo	276
Creazione di una richiesta di pull	278
Unione di una richiesta di pull	280
Visualizzazione del codice distribuito	281
Pulizia delle risorse	282
Memorizzazione del codice sorgente nei repository	282
Creazione di un repository di sorgenti	283
Clonazione di un repository Git esistente in un repository sorgente	285
Collegamento di un repository di sorgenti	288
Visualizzazione di un repository di origine	291
Modifica delle impostazioni per un archivio di sorgenti	292
Clonazione di un repository di sorgenti	294
Eliminazione di un archivio di origine	296
Organizzazione del codice sorgente: lavoro con le filiali	297
Creare un ramo	298
Gestione del ramo predefinito	298
Gestisci le azioni consentite con le regole di filiale	300
Comandi Git per filiali	303
Visualizzazione delle filiali e dei dettagli	304
Eliminazione di un ramo	305
Gestione dei file di codice sorgente	306
Creare o aggiungere un file	306
Visualizzazione di un file	309
Visualizzazione della cronologia delle modifiche apportate a un file	310
Modifica di un file	311
Ridenominazione o eliminazione di un file	312
Revisione del codice con richieste pull	312
Creazione di una richiesta di pull	314
Visualizzazione delle richieste pull	318
Gestione dei requisiti per l'unione con le regole di approvazione	320
Revisione di una richiesta di pull	322

Aggiornamento di una richiesta di pull	325
Unione di una richiesta di pull	327
Chiusura di una pull request	330
Comprendere le modifiche al codice sorgente con i commit	331
Visualizzazione dei commit verso un ramo	332
Modifica della modalità di visualizzazione dei commit (console) CodeCatalyst	333
Quote per gli archivi di origine	333
Scrivi e modifica il codice con Dev Environments	339
Creazione di un ambiente di sviluppo	340
Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo	341
Creazione di un ambiente di sviluppo in CodeCatalyst	341
Creazione di un ambiente di sviluppo in un IDE	345
Interruzione di un ambiente di sviluppo	345
Ripresa del lavoro in un ambiente di sviluppo	346
Modifica di un ambiente di sviluppo	348
Eliminazione di un ambiente di sviluppo	349
Connessione a un ambiente di sviluppo tramite SSH	350
Configurazione e utilizzo di un devfile	352
Modifica di un devfile	353
Funzionalità Devfile supportate da CodeCatalyst	354
Esempio di devfile per un ambiente di sviluppo	355
Risoluzione dei problemi relativi a un devfile del repository utilizzando la modalità di ripristino	356
Specificare immagini devfile universali	356
Comandi Devfile	361
Eventi Devfile	362
Componenti Devfile	363
Associazione di una connessione VPC a un ambiente di sviluppo	363
Quote per ambienti di sviluppo	365
Pubblica e condividi pacchetti software	366
Concetti sui pacchetti	367
Pacchetti	367
Namespace dei pacchetti	367
Versioni del pacchetto	368
Asset	368
Archivi di pacchetti	368

Archivi upstream	369
Archivi Gateway	369
Configurazione e utilizzo degli archivi di pacchetti	370
Creazione di un archivio di pacchetti	370
Connessione a un archivio di pacchetti	371
Eliminazione di un archivio di pacchetti	371
Configurazione e utilizzo dei repository upstream	372
Aggiungere un repository upstream	373
Modifica dell'ordine di ricerca dei repository upstream	374
Richiesta di una versione del pacchetto con repository upstream	374
Rimozione di un repository upstream	377
Connessione a archivi esterni pubblici	378
Archivi di pacchetti esterni supportati e relativi repository gateway	379
Pubblicazione e modifica di pacchetti	380
Pacchetti di pubblicazione	380
Visualizzazione dei dettagli della versione del pacchetto	382
Eliminazione di una versione del pacchetto	382
Aggiornamento dello stato di una versione del pacchetto	383
Modifica dei controlli di origine dei pacchetti	385
Utilizzo di npm	390
Configurazione e utilizzo di npm	391
gestione dei tag npm	400
Usare Maven	401
Configurazione e utilizzo di Gradle Groovy	402
Configurazione e utilizzo di mvn	411
Pacchetti di pubblicazione con curl	419
Utilizzo dei checksum e delle istantanee di Maven	421
Usando NuGet	423
Utilizzo CodeCatalyst con Visual Studio	423
Configurazione e utilizzo di nuget o dotnet	425
NuGet normalizzazione del nome del pacchetto, della versione e del nome dell'asset	429
NuGet compatibilità	430
Uso di Python	430
Configurazione di pip e installazione di pacchetti Python	431
Configurazione di Twine e pubblicazione di pacchetti Python	434
Normalizzazione dei nomi dei pacchetti in Python	436

Compatibilità con Python	437
Quote per i pacchi	437
Crea, testa e distribuisce con i flussi di lavoro	438
Informazioni sul file di definizione del flusso di lavoro	438
Utilizzo degli editor visivi e YAML della CodeCatalyst console	440
Alla scoperta dei flussi di lavoro	442
Visualizzazione dei dettagli sull'esecuzione del flusso di lavoro	443
Fasi successive	444
Concetti sui flussi di lavoro	444
Flussi di lavoro	444
File di definizione del flusso di lavoro	445
Azioni	445
Gruppi di azione	445
Artifacts	445
Calcolo	446
Ambienti	446
Cancelli	446
Report	446
Esecuzioni	447
Origini	447
Variabili	447
Trigger del flusso di lavoro	447
Guida introduttiva ai flussi di lavoro	448
Prerequisiti	449
Fase 1: Crea e configura il tuo flusso di lavoro	449
Fase 2: Salvate il flusso di lavoro con un commit	451
Fase 3: Visualizza i risultati della corsa	451
(Opzionale) Fase 4: pulizia	452
Creazione con flussi di lavoro	452
Come posso creare un'applicazione?	453
Vantaggi dell'azione di compilazione	454
Alternative all'azione di compilazione	454
Aggiungere l'azione di compilazione	454
Visualizzazione dei risultati dell'azione di compilazione	456
Tutorial: caricare artefatti su Amazon S3	456
YAML: crea e testa azioni	465

Test con flussi di lavoro	494
Tipi di rapporti sulla qualità	495
Aggiungere l'azione di test	498
Visualizzazione dei risultati di un'azione di test	499
Ignorare i test non riusciti	500
Integrazione con universal-test-runner	501
Configurazione dei report sulla qualità	503
Le migliori pratiche per i test	509
Proprietà SARIF	513
Implementazione con flussi di lavoro	522
Come posso distribuire un'applicazione?	522
Elenco delle azioni di distribuzione	523
Vantaggi delle azioni di implementazione	524
Alternative all'implementazione delle azioni	524
Distribuzione su Amazon ECS	525
Distribuzione su Amazon EKS	578
Implementazione di uno stack CloudFormation	626
Distribuzione di un'app AWS CDK	679
Avvio di un'app AWS CDK	704
Pubblicazione su Amazon S3	720
Implementazione in e Account AWS VPCs	736
Visualizzazione dell'URL dell'app	749
Rimozione di un obiettivo di distribuzione	753
Monitoraggio dello stato di distribuzione tramite commit	754
Visualizzazione dei registri di distribuzione	756
Visualizzazione delle informazioni sulla distribuzione	757
Creazione di un flusso di lavoro	758
Esecuzione di un workflow	762
Avvio manuale delle esecuzioni	763
L'avvio viene eseguito automaticamente utilizzando i trigger	764
Configurazione dei trigger solo manuali	781
Arresto dell'esecuzione di un flusso di lavoro	783
Autorizzare l'esecuzione di un flusso di lavoro	783
Richiesta di approvazioni per l'esecuzione del flusso di lavoro	787
Configurazione del comportamento di accodamento delle esecuzioni	800
Memorizzazione nella cache dei file tra le esecuzioni	807

Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro	811
Configurazione delle azioni del flusso di lavoro	816
Tipi di operazione	816
Aggiungere un'azione	820
Rimuovere un'azione	822
Sviluppo di un'azione personalizzata	823
Raggruppamento delle azioni in gruppi di azione	824
Azioni di sequenziamento	826
Condivisione di artefatti e file tra azioni	831
Specificare la versione dell'azione da utilizzare	845
Elenco delle versioni d'azione disponibili	847
Visualizzazione del codice sorgente di un'azione	848
Integrazione con Actions GitHub	849
Configurazione delle immagini di calcolo e di runtime	887
Tipi di calcolo	888
Flotte di calcolo	889
Proprietà del parco veicoli su richiesta	890
Proprietà del parco veicoli assegnate	891
Creazione di una flotta predisposta	892
Modifica di una flotta predisposta	893
Eliminazione di una flotta fornita	894
Assegnazione di una flotta o di un calcolo a un'azione	894
Condivisione dell'elaborazione tra le azioni	896
Specificazione delle immagini dell'ambiente di runtime	903
Connessione dei repository di origine ai flussi di lavoro	912
Specificare l'archivio di origine di un file di workflow	913
Specificare l'archivio di origine di un'azione di workflow	914
Riferimento ai file del repository di origine	916
Variabili: 'BranchName' e 'CommitId'	916
Collegamento degli archivi di pacchetti ai flussi di lavoro	917
Tutorial: Estrai da un archivio di pacchetti	918
Specificazione degli archivi di CodeCatalyst pacchetti nei flussi di lavoro	934
Utilizzo dei token di autorizzazione nelle azioni del flusso di lavoro	936
Esempi: archivi di pacchetti nei flussi di lavoro	938
Richiamo di una funzione Lambda	940
Quando utilizzare questa azione	941

Immagine di runtime utilizzata dall'azione 'invoke'AWS Lambda	941
Esempio: richiamare una funzione Lambda	941
Aggiungere l'AWS Lambda azione «invoca»	943
Variabili - 'invoca'AWS Lambda	945
YAML - azione 'invoca'AWS Lambda	946
Modifica della definizione di un'attività Amazon ECS	961
Quando usare questa azione	961
Come funziona l'azione «Render Amazon ECS task definition»	962
Immagine di runtime utilizzata dall'azione «Render Amazon ECS task definition»	964
Esempio: modificare un taskdef di Amazon ECS	964
Aggiungere l'azione «Render Amazon ECS task definition»	966
Visualizzazione del file di definizione delle attività aggiornato	969
Variabili - 'Renderizza la definizione dell'attività Amazon ECS'	970
YAML - 'Render la definizione dell'attività di Amazon ECS'	971
Utilizzo di variabili nei flussi di lavoro	980
Utilizzo di variabili definite dall'utente	980
Utilizzo di variabili predefinite	993
Mascherare i dati utilizzando segreti	998
Creare un segreto	999
Modificare un segreto	1000
Usare un segreto	1000
Eliminazione di un segreto	1002
Visualizzazione dello stato di un flusso di lavoro	1003
Quote di flusso di lavoro	1003
Stati di esecuzione del flusso di lavoro	1005
Stati del flusso di lavoro	1006
Definizione YAML del flusso di lavoro	1007
Esempio di file di definizione del flusso di lavoro	1008
Linee guida e convenzioni sulla sintassi	1009
Proprietà di primo livello	1011
Tieni traccia e organizza il lavoro con i problemi	1024
Concetti relativi ai problemi	1025
Problemi attivi	1025
Problemi archiviati	1026
Assegnatario	1026
Campi personalizzati	1026

Stima	1026
Problema	1026
Etichetta	1026
Priorità	1027
Stato e categorie di stato	1027
Processi	1027
Visualizzazioni	1027
Monitoraggio del lavoro con problemi	1028
Creazione di un problema	1028
Stima di un problema	1033
Redazione e collaborazione sui problemi	1034
Individuazione e visualizzazione dei problemi	1042
Progresso di un problema	1045
Archiviazione di un problema	1047
Problemi di esportazione	1048
Organizzazione del lavoro con arretrati, etichette e bacheche	1048
Categorizzazione del lavoro con le etichette	1049
Organizzazione del lavoro con campi personalizzati	1050
Il monitoraggio funziona con stati personalizzati	1051
Configurazione della stima dello sforzo da risolvere	1052
Abilitazione o disabilitazione di più assegnatari	1053
Creazione di una visualizzazione dei problemi	1054
Quote per le emissioni	1054
Configura identità, autorizzazioni e accesso in CodeCatalyst	1056
Concessione dell'accesso con ruoli utente	1057
Comprendere i ruoli degli utenti per spazi e progetti	1057
Visualizzazione delle autorizzazioni disponibili per ogni ruolo	1060
Visualizzazione e modifica dei ruoli utente	1105
Concedi agli utenti l'accesso al repository con token di accesso personali	1107
Creando PATs	1108
Visualizzazione PATs	1110
Eliminazione PATs	1111
.....	1113
Creazione di connessioni personali	1113
Eliminazione delle connessioni personali	1115
Configura il tuo AWS Builder ID per accedere con l'autenticazione a più fattori (MFA)	1116

Come registrare un dispositivo per l'utilizzo con l'autenticazione a più fattori	1117
Applicazioni di autenticazione	1119
Modifica dei dispositivi MFA	1120
Sicurezza	1121
Privacy dei dati	1122
Protezione dei dati	1122
CodeCatalyst e Identity and Access Management	1125
Convalida della conformità	1180
Resilienza	1181
Sicurezza dell'infrastruttura	1181
Analisi della configurazione e delle vulnerabilità	1181
I tuoi dati e la tua privacy in Amazon CodeCatalyst	1181
Le migliori pratiche per le azioni del flusso di lavoro	1182
Comprendere il modello CodeCatalyst di fiducia	1183
Monitoraggio degli eventi e delle chiamate API tramite la registrazione	1186
Monitoraggio delle chiamate API Account AWS utilizzando la registrazione AWS	
CloudTrail	1188
Accesso agli eventi registrati utilizzando la registrazione degli eventi	1196
Quote per identità, autorizzazione e accesso	1199
Risoluzione dei problemi	1201
Problemi di registrazione	1201
Problemi di accesso	1202
Problemi con la disconnessione	1203
Errore «I get a role does not exist» per un flusso di lavoro non riuscito	1203
Ricevo un errore di ruolo a causa di un flusso di lavoro non riuscito	1203
Devo aggiornare il ruolo IAM nel flusso di lavoro di un progetto	1204
Ho una richiesta di revisione per il mio GitHub account dopo aver creato una connessione personale	1204
Come posso compilare un modulo di supporto?	1205
Aggiungi funzionalità ai progetti con estensioni	1206
Estensioni di terze parti disponibili	1206
Integrazione dei GitHub repository in CodeCatalyst	1207
Integrazione dei repository Bitbucket in CodeCatalyst	1208
GitLab Integrazione dei repository in CodeCatalyst	1209
Integrazione dei problemi di Jira in CodeCatalyst	1210
Concetti di estensioni	1210

Estensioni	1210
CodeCatalyst catalogo	1210
Connessione e collegamento	1211
Guida rapida: installazione di estensioni, connessione di provider e collegamento di risorse ..	1211
Passaggio 1: installa un'estensione di terze parti dal catalogo CodeCatalyst	1214
Passaggio 2: Collega il tuo provider di terze parti al tuo CodeCatalyst spazio	1214
Passaggio 3: collega le risorse di terze parti al tuo progetto CodeCatalyst	1218
Fasi successive	1221
Installazione di un'estensione in uno spazio	1222
Disinstallazione di un'estensione in uno spazio	1223
Connessione di GitHub account, aree di lavoro Bitbucket, utenti e siti Jira GitLab	1224
Disconnessione di GitHub account, aree di lavoro Bitbucket, utenti e siti Jira GitLab	1228
Repository di collegamento, repository Bitbucket, GitHub repository di progetti e progetti Jira GitLab	1230
Collegamento di risorse di fornitori terzi collegati	1232
Collega un repository di terze parti durante la creazione del progetto	1238
GitHub Repository scollegati, repository Bitbucket, repository di progetti e progetti Jira GitLab	1238
Visualizzazione di repository di terze parti e ricerca dei problemi di Jira in CodeCatalyst	1240
Visualizzazione di repository di terze parti in CodeCatalyst	1241
Sto cercando i problemi di Jira in CodeCatalyst	1242
Avvio automatico di un flusso di lavoro eseguito dopo eventi di repository di terze parti	1242
L'aggiunta di trigger per avviare le esecuzioni del flusso di lavoro	1243
Limitazione dell'accesso IP con fornitori di repository di terze parti	1244
Indirizzi IP utilizzati dall'estensione degli archivi di terze parti	1245
Bloccare le fusioni di terze parti quando i flussi di lavoro falliscono	1245
Collegamento dei problemi di Jira alle richieste pull	1246
Visualizzazione degli CodeCatalyst eventi nei problemi di Jira	1248
Cerca codice, problemi, progetti e utenti	1249
Perfezionamento della query di ricerca	1250
Perfezionamento per tipo	1250
Perfezionamento per campo	1251
Perfezionamento con operatori booleani	1251
Perfezionamento per progetto	1251
Considerazioni sull'utilizzo della ricerca	1252
Riferimento ai campi ricercabili	1253

Risoluzione dei problemi	1259
Risoluzione dei problemi generali di accesso	1259
Ho dimenticato la mia password	1259
Amazon CodeCatalyst non è disponibile in tutto o in parte	1260
Non riesco a creare un progetto in CodeCatalyst	1260
Risoluzione dei problemi di supporto	1260
Ricevo un errore quando accedo Supporto ad Amazon CodeCatalyst	1260
Non riesco a creare casi di supporto tecnico per il mio spazio	1261
Il mio account per i casi di assistenza non è più collegato al mio spazio in CodeCatalyst ...	1261
Non riesco ad aprire un caso di supporto per un altro Servizio AWS in Supporto Amazon CodeCatalyst	1262
Amazon CodeCatalyst non è disponibile in tutto o in parte	1260
Non riesco a creare un progetto in CodeCatalyst	1260
Non riesco ad accedere al mio spazio BID come nuovo utente o non posso essere aggiunto come nuovo utente SSO a causa del nome utente troncato	1263
L'aggiunta di un utente SSO al mio spazio federato come nuovo utente ha creato un utente duplicato	1264
Voglio inviare un feedback in CodeCatalyst	1264
Risoluzione dei problemi relativi ai repository di origine	1265
Ho raggiunto lo spazio di archiviazione massimo disponibile e visualizzo avvisi o errori	1265
Ricevo un errore durante il tentativo di clonare o eseguire il push su un repository di CodeCatalyst sorgenti Amazon	1266
Ricevo un errore durante il tentativo di eseguire il commit o il push su un repository di CodeCatalyst sorgenti Amazon	1267
Ho bisogno di un repository di sorgenti per il mio progetto	1267
Il mio repository di origine è nuovo di zecca ma contiene un commit	1267
Voglio un ramo diverso come ramo predefinito	1268
Ricevo e-mail sull'attività nelle pull request	1268
Ho dimenticato il mio token di accesso personale (PAT)	1268
Una pull request non mostra le modifiche che mi aspetto	1268
Una pull request mostra lo stato Non unibile	1269
Risoluzione dei problemi relativi a progetti e blueprint	1270
API Java con dipendenze del AWS Fargate blueprint mancanti per apache-maven-3.8.6 ..	1270
Il moderno flusso di lavoro del blueprint di applicazioni Web a tre livelli OnPullRequestnon riesce a causa di un errore di autorizzazione per Amazon CodeGuru	1271
Stai ancora cercando di risolvere il tuo problema?	1274

Risoluzione dei problemi degli ambienti di sviluppo	1274
La creazione del mio ambiente di sviluppo non è riuscita a causa di un problema con le quote	1275
Non riesco a trasferire le modifiche dal mio ambiente di sviluppo a un ramo specifico di un repository	1276
Il mio ambiente di sviluppo non è stato ripreso	1276
Il mio ambiente di sviluppo si è disconnesso	1276
Il mio ambiente di sviluppo connesso al VPC non è riuscito	1276
Non riesco a trovare la cartella in cui si trova il mio progetto	1277
Non riesco a connettermi al mio ambiente di sviluppo tramite SSH	1277
Non riesco a connettermi al mio ambiente di sviluppo tramite SSH perché manca la mia configurazione SSH locale	1277
Non riesco a connettermi al mio ambiente di sviluppo tramite SSH perché ho problemi con il mio profilo for the AWS Configcodecatalyst	1277
Non riesco a creare un ambiente di sviluppo quando ho effettuato l'accesso CodeCatalyst utilizzando un account Single Sign-on	1278
Risoluzione dei problemi IDEs	1278
Risoluzione dei problemi relativi ai file dev	1280
Risoluzione dei flussi di lavoro	1282
Come posso correggere i messaggi «Workflow is inactive»?	1283
Come posso correggere gli errori «La definizione del flusso di lavoro presenta <i>n</i> errori»? ..	1284
Come posso correggere gli errori «Impossibile individuare le credenziali» e "ExpiredToken"?	1286
Come posso correggere gli errori «Impossibile connettersi al server»?	1288
Perché mancano CodeDeploy dei campi nell'editor visivo?	1288
Come posso correggere gli errori relativi alle funzionalità IAM?	1289
Come posso correggere gli errori di «npm install»?	1291
Perché più flussi di lavoro hanno lo stesso nome?	1294
Posso archiviare i miei file di definizione del flusso di lavoro in un'altra cartella?	1295
Come faccio ad aggiungere azioni in sequenza al mio flusso di lavoro?	1295
Perché il mio flusso di lavoro viene convalidato correttamente ma fallisce in fase di esecuzione?	1295
L'individuazione automatica non rileva alcun rapporto relativo alla mia azione	1296
La mia azione non riesce sui report rilevati automaticamente dopo aver configurato i criteri di successo	1297
L'individuazione automatica genera report che non voglio	1297

L'individuazione automatica genera molti report di piccole dimensioni per un singolo framework di test	1297
I flussi di lavoro elencati in CI/CD non corrispondono a quelli del repository di origine	1298
Non riesco a creare o aggiornare i flussi di lavoro	1299
Risoluzione dei problemi	1299
Non riesco a scegliere un assegnatario per il mio problema	1299
Risoluzione dei problemi di ricerca	1300
Non riesco a trovare un utente nel mio progetto	1300
Non vedo cosa sto cercando nel mio progetto o nel mio spazio	1300
Il numero di risultati di ricerca continua a cambiare quando navigo tra le pagine	1301
La mia query di ricerca non è stata completata	1301
Risoluzione dei problemi delle estensioni	1301
Non riesco a visualizzare le modifiche a un archivio di terze parti collegato o a cercare i risultati di tali modifiche	1301
Risoluzione dei problemi relativi agli account associati	1302
La mia richiesta di Account AWS connessione riceve un errore relativo al token non valido	1302
Il flusso di lavoro CodeCatalyst del mio progetto Amazon non funziona con un errore per l'account configurato, l'ambiente o il ruolo IAM	1303
Ho bisogno di un account, un ruolo e un ambiente associati per creare un progetto	1304
Non riesco ad accedere alla pagina Amazon CodeCatalyst Spaces nel Console di gestione AWS	1305
Voglio un account diverso come account di fatturazione	1305
Il flusso di lavoro del mio progetto fallisce a causa di un errore nel nome di connessione ...	1306
Risoluzione dei problemi AWS CLI relativi all'SDK	1306
Ricevo un errore quando aws codecatalyst accedo da una riga di comando o da un terminale che dice che si tratta di una scelta non valida	1306
Ricevo un errore relativo alle credenziali quando eseguo i comandi aws codecatalyst	1307
Comprendere lo stato attuale del servizio con il rapporto sullo stato di CodeCatalyst salute	1308
CodeCatalyst concetti del rapporto sullo stato di salute	1308
Incidente	1309
Stato	1309
Funzionalità impattate	1309
Aggiornato il	1309
Supporto per Amazon CodeCatalyst	1310
Fatturazione Supporto per Amazon CodeCatalyst	1310
Configurazione dello spazio Supporto per Amazon CodeCatalyst	1313

Accesso al supporto per CodeCatalyst in Console di gestione AWS	1314
Creazione di una CodeCatalyst richiesta di supporto in CodeCatalyst	1315
Risoluzione di un caso di supporto in CodeCatalyst	1318
Riapertura di una richiesta di supporto in CodeCatalyst	1318
Quote	1320
Cronologia dei documenti	1322
AWS Glossario	1355

Amazon non CodeCatalyst è più aperta a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta [Come migrare da CodeCatalyst](#).

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

Che cos'è Amazon CodeCatalyst?

Amazon CodeCatalyst è un servizio integrato per i team di sviluppo software che adottano pratiche di integrazione e distribuzione continue nel loro processo di sviluppo software. CodeCatalyst mette tutti gli strumenti di cui hai bisogno in un unico posto. Puoi pianificare il lavoro, collaborare sul codice e creare, testare e distribuire applicazioni con strumenti integration/continuous delivery (CI/CD) (continui). Puoi anche integrare AWS le risorse con i tuoi progetti collegando il tuo spazio Account AWS al tuo CodeCatalyst spazio. Gestendo tutte le fasi e gli aspetti del ciclo di vita delle applicazioni con un unico strumento, è possibile fornire software in modo rapido e sicuro.

In CodeCatalyst primo luogo, si crea uno spazio per rappresentare l'azienda, il reparto o il gruppo e quindi si creano progetti che contengono le risorse necessarie per supportare i team e le attività di sviluppo. CodeCatalyst risorse sono strutturate all'interno di progetti che vivono all'interno di spazi. Per aiutare i team a iniziare rapidamente, CodeCatalyst fornisce modelli di progetto basati sul linguaggio o sugli strumenti. Quando crei un progetto da un blueprint di progetto, il progetto include risorse come un archivio di sorgenti con codice di esempio, script di compilazione, azioni di distribuzione, server virtuali o risorse serverless e altro ancora.

Cosa posso fare con? CodeCatalyst

Tu e il tuo team di sviluppo potete CodeCatalyst occuparvi di ogni aspetto dello sviluppo del software, dalla pianificazione del lavoro alla distribuzione delle applicazioni. Puoi usare CodeCatalyst per:

- Iterazione e collaborazione sul codice: collabora con il tuo team sul codice con branch, unioni, richieste pull e commenti nei tuoi repository di codice sorgente. Crea ambienti di sviluppo per lavorare rapidamente sul codice senza dover clonare o configurare connessioni ai repository.
- Crea, testa e distribuisce la tua applicazione con flussi di lavoro: configura i flussi di lavoro con azioni di compilazione, test e distribuzione per gestire l'integrazione e la distribuzione continue delle tue applicazioni. Puoi avviare i flussi di lavoro manualmente o configurarli per l'avvio automatico in base a eventi come l'invio di codice o la creazione o la chiusura di richieste pull.
- Dai priorità al lavoro del tuo team con il monitoraggio dei problemi: utilizza i problemi per creare arretrati e monitorare lo stato delle attività in corso con le bacheche. Creare e mantenere un buon portafoglio di articoli su cui il team possa lavorare è una parte importante dello sviluppo del software.
- Configura il monitoraggio e le notifiche: monitora l'attività del team e lo stato delle risorse e configura le notifiche per rimanere aggiornato sulle modifiche importanti.

Come posso iniziare con CodeCatalyst?

Se non disponi di uno spazio o desideri imparare a configurare e gestire uno spazio, ti consigliamo di iniziare con l'[Amazon CodeCatalyst Administrator Guide](#).

Se sei alle prime armi con un progetto o uno spazio, ti consigliamo di iniziare con:

- Revisione del [CodeCatalyst concetti](#)
- [Creare uno spazio](#)
- Crea il tuo primo progetto seguendo i passaggi indicati [Tutorial: creazione di un progetto con il moderno modello di applicazione web a tre livelli](#)

Scopri di più su CodeCatalyst

Ulteriori informazioni sulle funzionalità di sono disponibili CodeCatalyst in questa guida per l'utente e nelle seguenti risorse:

- [AWS DevOps Articoli di blog su Amazon CodeCatalyst](#)
- [La guida di riferimento per le CodeCatalyst API di Amazon](#)
- [Guida per gli sviluppatori di Amazon CodeCatalyst Action Development Kit](#)
- [CodeCatalyst DOMANDE FREQUENTI](#)
- [Testimonianze](#)

Come migrare da CodeCatalyst

Dopo un'attenta valutazione, abbiamo deciso di chiudere l'accesso di nuovi clienti ad Amazon CodeCatalyst a partire dal 7 novembre 2025. CodeCatalyst I clienti Amazon esistenti possono continuare a utilizzare il servizio con gli spazi esistenti, ma non saranno in grado di crearne di nuovi. AWS continua a investire in sicurezza e disponibilità per Amazon CodeCatalyst, ma non abbiamo intenzione di introdurre nuove funzionalità.

I clienti possono migrare manualmente i propri dati da Amazon CodeCatalyst ad altri fornitori. Questo documento descrive i metodi di base per migrare, estrarre o eliminare i dati dalle console CodeCatalyst e di AWS gestione. Le risorse e i dati creati in altri servizi AWS o in servizi 3P tramite la CodeCatalyst console dovranno essere eliminati tramite tali servizi per evitare l'addebito di addebiti (se applicabile).

I clienti possono prendere in considerazione la migrazione a [GitLab Duo con Amazon Q](#), che ha annunciato la disponibilità generale il 17 aprile 2025. Questa nuova offerta è un prodotto integrato che unisce la DevSecOps piattaforma con le funzionalità [GitLabdi](#) intelligenza artificiale generativa di Amazon Q. Gitlab Duo con Amazon Q integra le funzionalità degli agenti Amazon Q direttamente GitLab nella DevSecOps piattaforma per accelerare attività complesse e in più fasi durante l'intero ciclo di vita dello sviluppo del software.

Migrazione dei tuoi repository

Migrazione del repository in un CodeCatalyst repository GitLab

[Utilizzando l'URL prerequisito in combinazione con le credenziali del repository Git HTTPS, segui le indicazioni contenute nella documentazione relativa all'importazione GitLab del codice sorgente da un repository tramite URL.](#)

Migrazione del repository in un repository CodeCatalyst GitHub

Utilizzando l'URL prerequisito in combinazione con le credenziali del repository Git HTTPS, segui le indicazioni contenute nella documentazione relativa all'[importazione GitHub](#) del codice sorgente.

Migrazione generica verso un altro provider di repository

1. Clona il repository CodeCatalyst

Clona il CodeCatalyst repository Amazon sul tuo computer locale usando Git. Se utilizzi HTTPS, puoi farlo eseguendo il seguente comando:

```
git clone --mirror https://your-aws-repository-url your-aws-repository
```

Sostituiscilo `your-aws-repository-url` con l'URL del tuo CodeCatalyst repository Amazon.
`your-aws-repository` Sostituiscilo con un nome per questo repository.

Esempio:

```
git clone https://git-codecatalyst.us-east-2.amazonaws.com/v1/repos/MyDemoRepo my-demo-repo
```

2. Configura un nuovo puntatore al repository remoto

Accedi alla directory del tuo repository Amazon CodeCatalyst clonato. Quindi, aggiungi l'URL del repository dal nuovo provider di repository come telecomando:

```
git remote add <provider name> <provider-repository-url>
```

`<provider name>` Sostituiscilo con il nome del provider che preferisci. (Esempio: gitlab)

Sostituisci `< provider-repository-url >` con l'URL del repository del tuo nuovo fornitore di repository.

3. Invia il tuo repository locale al nuovo repository remoto:

Questo invierà tutti i rami e i tag all'archivio del nuovo fornitore di repository. Il nome del provider deve corrispondere al nome del provider indicato nella fase 2.

```
git push <provider name> --mirror
```

Note:

- L'archivio remoto deve essere vuoto
- L'archivio remoto può avere filiali protette che non consentono l'invio forzato, a seconda del provider. In tal caso, devi accedere al nuovo provider di repository e disabilitare le protezioni delle filiali per consentire il force push.

4. Verifica la migrazione

Una volta completato il push, verifica che tutti i file, i rami e i tag siano stati migrati correttamente al nuovo provider di repository. Puoi farlo sfogliando il tuo repository online o clonandolo in un'altra posizione e controllandolo localmente.

5. Aggiornamento remoto URLs (opzionale)

Se prevedi di continuare a lavorare con l'archivio migrato localmente, potresti voler aggiornare l'URL remoto in modo che punti all'archivio del nuovo provider anziché ad Amazon. CodeCatalyst A questo proposito, utilizzare il seguente comando:

```
git remote set-url origin <provider-repository-url>
```

Sostituisci < provider-repository-url > con l'URL del repository del tuo nuovo fornitore di repository.

Estrazione dei dati da CodeCatalyst

Scarica artefatti

Puoi scaricare e ispezionare gli artefatti generati dalle azioni del tuo flusso di lavoro Amazon CodeCatalyst. Esistono due tipi di artefatti che puoi scaricare:

- **Artefatti di origine:** un artefatto che contiene un'istantanea del contenuto del repository di origine così com'era all'avvio dell'esecuzione.
- **Artefatti del flusso di lavoro:** un artefatto definito nella proprietà Outputs del file di configurazione del flusso di lavoro.

Per scaricare gli artefatti generati dal flusso di lavoro:

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.

5. Sotto il nome del flusso di lavoro, scegli Esegui.
6. Nella cronologia delle esecuzioni, nella colonna Run ID, scegli una corsa. Ad esempio, RUN-95A4D.
7. Sotto il nome della corsa, scegli Artefatti.
8. Accanto a un artefatto, scegli Scarica. Verrà scaricato un file di archivio. Il nome del file è composto da sette caratteri casuali.
9. Estrai l'archivio utilizzando un'utilità di estrazione dell'archivio di tua scelta.

Scarica gli allegati del tuo numero

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegli il problema per il quale desideri gestire gli allegati. Per assistenza nell'individuazione del problema, consulta [Ricerca e visualizzazione dei problemi](#)
3. Per scaricare un allegato, scegli il menu con i puntini di sospensione accanto all'allegato che desideri scaricare e scegli Scarica.

Scarica il codice sorgente di un'azione

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Trova l'azione di cui desideri visualizzare il codice:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - b. Scegli il nome di qualsiasi flusso di lavoro o creane uno. Per informazioni sulla creazione di un flusso di lavoro, consulta [Creazione di un flusso di lavoro](#).
 - c. Scegli Modifica.
 - d. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
 - e. Nell'elenco a discesa, scegli Amazon CodeCatalyst to view CodeCatalyst, CodeCatalyst Labs e azioni di terze parti.
 - f. Cerca un'azione e scegline il nome. Non scegliere il segno più (+).

Vengono visualizzati i dettagli sull'azione.

4. Nella finestra di dialogo dei dettagli dell'azione, nella parte inferiore, scegli Scarica.

Viene visualizzata una pagina che mostra il bucket Amazon S3 in cui risiede il codice sorgente dell'azione. Per informazioni su Amazon S3, consulta [What is Amazon S3?](#) nella Guida per l'utente di Amazon Simple Storage Service.

Eliminazione dei dati da CodeCatalyst

Prima di eliminare i dati da CodeCatalyst, informa il team sulla migrazione del servizio e verifica che non siano ancora necessarie risorse. Una volta eliminati, i dati e le risorse non possono essere recuperati.

Richiedi al team di assistenza di eliminare i dati per tuo conto

Un amministratore dello spazio può richiedere al team di assistenza di eliminare uno spazio per suo conto contattandoci tramite il Support Center nella CodeCatalyst console. Gli amministratori dello spazio devono essere autenticati nella CodeCatalyst console per richiedere l'eliminazione dello spazio. Dopo aver inviato la richiesta, il team di assistenza ti contatterà per confermare la richiesta prima di agire per tuo conto.

Elimina il tuo CodeCatalyst spazio

È possibile eliminare uno spazio per rimuovere l'accesso a tutte le risorse dello spazio. È necessario disporre del ruolo di amministratore dello spazio per eliminare uno spazio.

Nota: non è possibile annullare l'eliminazione di uno spazio e non è possibile recuperare i dati una volta eliminato uno spazio.

Dopo aver eliminato uno spazio, tutti i membri dello spazio non saranno in grado di accedere alle risorse dello spazio. Anche la fatturazione delle risorse spaziali verrà interrotta e tutti i flussi di lavoro richiesti dagli archivi di sorgenti di terze parti verranno interrotti.

Se appartieni a più di uno spazio, scegli lo spazio nella barra di navigazione in alto.

Per eliminare uno spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Elimina.
4. Digita **delete** per confermare l'eliminazione.
5. Scegli Elimina.

 Note

Se appartieni a più di uno spazio, verrai reindirizzato alla pagina di panoramica dello spazio. Se appartieni a uno spazio, verrai reindirizzato alla pagina di creazione dello spazio.

Se elimini uno spazio ma appartieni a più di uno spazio, verrai reindirizzato alla pagina di panoramica dello spazio. Se appartieni a uno spazio, verrai reindirizzato alla pagina di creazione dello spazio.

Se hai creato risorse in altri servizi AWS o in servizi di terze parti dalla CodeCatalyst console, dovrai accedere a tali servizi singolarmente per disattivare le risorse dall'account di fatturazione in cui sono state create. L'eliminazione di uno spazio eliminerà solo i CodeCatalyst dati e le risorse.

Elimina un progetto

È possibile eliminare un progetto per rimuovere tutti gli accessi alle risorse del progetto. È necessario disporre del ruolo di amministratore dello spazio o amministratore del progetto per eliminare un progetto. Dopo aver eliminato un progetto, i membri del progetto non saranno in grado di accedere alle risorse del progetto e tutti i flussi di lavoro richiesti dagli archivi di origine di terze parti verranno interrotti.

Per eliminare il progetto:

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Passa allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Seleziona Delete project (Elimina progetto).

5. Invio **delete** per confermare l'eliminazione.
6. Seleziona Delete project (Elimina progetto).

Se hai creato risorse in altri servizi AWS o in servizi di terze parti nell'ambito di un CodeCatalyst progetto, dovrai utilizzare i singoli servizi per disattivarli dall'account di fatturazione con cui sono state create. L'eliminazione di uno spazio eliminerà solo i CodeCatalyst dati e le risorse.

Eliminare un archivio di origine

Puoi eliminare i tuoi repository di origine per un CodeCatalyst progetto Amazon. L'eliminazione di un repository di origine elimina anche tutte le informazioni sul progetto memorizzate nel repository. Se alcuni flussi di lavoro dipendono dall'archivio di origine, tali flussi di lavoro verranno eliminati dall'elenco dei flussi di lavoro del progetto dopo l'eliminazione del repository. I problemi che fanno riferimento all'archivio di origine non verranno eliminati o modificati, ma tutti i collegamenti all'archivio di origine aggiunti ai problemi falliranno una volta eliminato il repository.

Importante: l'eliminazione di un repository di origine non può essere annullata. Dopo aver eliminato un repository di origine, non è più possibile clonarlo, estrarre dati da esso o inviargli dati. L'eliminazione di un repository di origine non elimina alcuna copia locale di tale repository (repository locali). Per eliminare un repository locale, utilizzate gli strumenti di gestione delle directory e dei file del computer locale.

Nota: non è possibile eliminare un repository collegato nella CodeCatalyst console. Per eliminare un repository collegato, scegli il link nell'elenco dei repository per aprirlo nel servizio che lo ospita, quindi eliminalo. Per ulteriori informazioni, consulta la documentazione del servizio che ospita l'archivio collegato.

Per rimuovere un repository collegato da un progetto, consulta [Unlinking GitHub repository, Bitbucket repository, GitLab project repository e Jira projects in. CodeCatalyst](#)

Per eliminare un repository di origine

1. Passa al progetto che contiene il repository di origine che desideri eliminare.
2. Nella pagina di riepilogo del progetto, scegli il repository che desideri dall'elenco, quindi scegli Visualizza archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli il nome del repository dall'elenco dei repository di origine per il progetto.
3. Nella home page del repository, scegli Altro, scegli Gestisci impostazioni, quindi scegli Elimina archivio.

4. Esamina le informazioni sul branch, sulla pull request e sul relativo flusso di lavoro per assicurarti di non eliminare un repository ancora in uso o con del lavoro incompiuto. Se vuoi continuare, digita delete, quindi scegli Elimina.

Eliminare un blueprint personalizzato

Quando elimini un blueprint dal tuo CodeCatalyst spazio Amazon, tutti i tuoi accessi vengono rimossi alle risorse del progetto blueprint o della versione del blueprint. Quando elimini un blueprint, i membri del progetto non saranno in grado di accedere alle risorse del progetto e tutti i flussi di lavoro richiesti dagli archivi di origine di terze parti verranno interrotti.

Se si elimina un blueprint, ciò non influisce sul progetto a cui è applicato il blueprint. Le risorse del blueprint non vengono rimosse dal progetto.

Importante: per eliminare un blueprint personalizzato pubblicato o la versione del catalogo di un blueprint personalizzato dal proprio spazio, è necessario accedere con un account con il ruolo di amministratore dello spazio o Power user nello spazio.

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri eliminare un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Nella tabella Impostazioni, scegli il pulsante di opzione relativo al blueprint personalizzato che desideri eliminare, quindi scegli Elimina blueprint.
5. Inserisci delete per confermare l'eliminazione della versione del catalogo dei blueprint.
6. Scegli Elimina.

Elimina gli allegati del tuo problema

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegli il problema per il quale desideri gestire gli allegati. Per assistenza nell'individuazione del problema, consulta [Ricerca e visualizzazione dei problemi](#).
3. Per rimuovere un allegato, scegli il menu con i puntini di sospensione accanto all'allegato che desideri rimuovere e scegli Elimina.

Eliminazione di un file in ambienti di sviluppo a cui si accede tramite Amazon CodeCatalyst

Puoi eliminare i file in un ambiente di sviluppo, localmente sul tuo computer o in un ambiente di sviluppo integrato (IDE). Non puoi eliminare file nella CodeCatalyst console Amazon.

Eliminazione di un ambiente di sviluppo per il tuo spazio

Per ulteriori informazioni sulle considerazioni relative all'eliminazione di un ambiente di sviluppo, vedi [Eliminazione](#) di un ambiente di sviluppo.

È necessario disporre del ruolo di amministratore dello spazio per visualizzare questa pagina e gestire gli ambienti di sviluppo a livello di spazio. Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

Per eliminare un ambiente di sviluppo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio
3. Scegli Impostazioni, quindi scegli Dev Environments.
4. Scegli il selettore accanto all'ambiente di sviluppo che desideri gestire. Scegli Elimina.
5. Invio **delete** per confermare l'eliminazione di Dev Environment.
6. Scegli Elimina.

Eliminazione delle connessioni degli account

Puoi eliminare una connessione all'account nella CodeCatalyst console che hai aggiunto in precedenza al tuo spazio. Dopo l'eliminazione di una connessione all'account, non è possibile ricollegare la connessione ed è necessario stabilirne una nuova.

Per il tuo CodeCatalyst spazio deve essere designato un account di fatturazione, anche se l'utilizzo dello spazio non supererà il livello gratuito. Prima di poter rimuovere uno spazio per un account che è un account di fatturazione designato, dovrai aggiungere un altro account per lo spazio. Se desideri eliminare il tuo account di fatturazione per lo spazio, devi eliminare lo spazio. Consulta [la sezione Gestione della fatturazione](#) nella Amazon CodeCatalyst Administrator Guide.

Per gestire le connessioni dell'account per il tuo spazio, devi avere il ruolo di amministratore dello spazio o Power user.

Un account che è stato rimosso può essere aggiunto nuovamente in un secondo momento, ma è necessario creare una nuova connessione tra l'account e lo spazio. Dovrai riassociare tutti i ruoli IAM all'account.

Per eliminare una connessione all'account

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
3. In Amazon CodeCatalyst Display name, scegli il selettore accanto alla connessione dell'account che desideri rimuovere.
4. Scegli Rimuovi Account AWS. Conferma l'eliminazione inserendo il nome nel campo, quindi scegli Rimuovi.

Viene visualizzato un banner di successo e la connessione all'account viene rimossa dall'elenco delle connessioni.

Rimuovere un account da uno CodeCatalyst spazio nella Console AWS di gestione

Puoi utilizzare la pagina for CodeCatalyst in AWS per rimuovere un account che è stato aggiunto a uno spazio. Per questa procedura, utilizzando le autorizzazioni amministrative per l'account specifico che stai gestendo, accedi alla pagina Amazon CodeCatalyst Spaces nella Console di AWS gestione per rimuovere un AWS account dal tuo spazio. Per rimuovere un account che è un account di fatturazione designato per il tuo CodeCatalyst spazio, assicurati di specificare prima un nuovo account di fatturazione.

Un account che è stato rimosso può essere aggiunto nuovamente in un secondo momento, ma devi creare una nuova connessione tra l'account e lo spazio. Dovrai riassociare tutti i ruoli IAM all'account aggiunto.

È necessario designare un account di fatturazione per il tuo CodeCatalyst spazio, anche se l'utilizzo dello spazio non supererà il livello gratuito. Prima di poter rimuovere uno spazio per un account che è un account di fatturazione designato, dovrai aggiungere un altro account per lo spazio.

È necessario disporre del ruolo di amministratore dello spazio o utente esperto per gestire le connessioni degli account per il proprio spazio.

Per rimuovere un account aggiunto

1. Nel Console di gestione AWS, assicurati di aver effettuato l'accesso con lo stesso account che desideri gestire.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Scegli Impostazioni, quindi scegli Fatturazione.
4. Visualizza le informazioni sull'account di fatturazione sulla pagina per assicurarti che l'account che desideri rimuovere non sia l'account di fatturazione designato per lo spazio.
5. Scegli Gestisci la fatturazione. AWS Questo apre Amazon CodeCatalyst Spaces in Console di gestione AWS. Se ti viene richiesto di accedere, accedi a AWS, quindi scegli nuovamente il pulsante per caricare la pagina.
6. Nella pagina Amazon CodeCatalyst Spaces, scegli lo spazio con l'account che desideri rimuovere. Viene visualizzata la pagina dei dettagli dello spazio.
7. Scegli Rimuovi spazio.
8. In Rimuovi CodeCatalyst spazio da questo account, inserisci il nome dello spazio per confermare. Scegli Rimuovi.

Eliminazione di un segreto

Utilizzare la procedura seguente per eliminare un segreto e l'identificatore di riferimento segreto. Prima di eliminare un segreto, si consiglia di rimuovere l'identificatore di riferimento del segreto da tutte le azioni del flusso di lavoro. Se elimini il segreto senza eliminare l'identificatore di riferimento, l'azione avrà esito negativo alla successiva esecuzione.

Per eliminare l'identificatore di riferimento di un segreto da un flusso di lavoro

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli YAML.
6. Cerca nel flusso di lavoro la seguente stringa:


```
${Secrets.}
```

In questo modo vengono trovati tutti gli identificatori di riferimento di tutti i segreti.

7. Elimina l'identificatore di riferimento del segreto scelto o sostituiscilo con un valore in chiaro.
8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Eliminare il segreto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Secrets.
3. Nell'elenco dei segreti, scegli il segreto che desideri eliminare.
4. Scegli Elimina.
5. Inserisci **delete** per confermare l'eliminazione.
6. Scegli Elimina.

Eliminazione di un team

Puoi eliminare un team che non ti serve più. Quando elimini un team, le autorizzazioni associate verranno rimosse per tutti i membri del team da tutti i progetti e le risorse dello spazio. È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Eliminare un team

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. In Azioni, scegli Elimina team. Questo cambia il ruolo dell'intero team.
4. Scegli Elimina.

Eliminazione di una flotta predisposta

Utilizza le seguenti istruzioni per eliminare una flotta predisposta.

Prima di eliminare una flotta assegnata, rimuovila da tutte le azioni eliminando la proprietà Fleet dal codice YAML dell'azione. Qualsiasi azione che continui a fare riferimento a una flotta predisposta dopo la sua eliminazione avrà esito negativo alla successiva esecuzione dell'azione.

Per eliminare una flotta predisposta

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Compute.
2. Nell'elenco Provisioned fleet, scegli il parco veicoli che desideri eliminare.
3. Scegli Elimina.
4. Inserisci **delete** per confermare l'eliminazione.
5. Scegli Elimina.

Eliminazione di un archivio di pacchetti

Effettuare le seguenti operazioni per eliminare un archivio di pacchetti in CodeCatalyst

Per eliminare un archivio di pacchetti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa al progetto che contiene l'archivio dei pacchetti che desideri eliminare.
3. Dal riquadro di navigazione, scegli Pacchetti.
4. Nella pagina Package repository, scegli il repository che desideri eliminare.
5. Scegli Elimina.
6. Esamina le informazioni fornite sugli effetti dell'eliminazione di un archivio di pacchetti.
7. Entra **delete** nel campo di immissione e scegli Elimina.

Tutte le altre risorse archiviate in Amazon CodeCatalyst verranno eliminate quando il tuo spazio verrà eliminato. Ciò non include le risorse e i dati creati in altri servizi AWS o in servizi 3P tramite la CodeCatalyst console. Tutte le risorse create nei servizi esterni alla CodeCatalyst console dovranno essere eliminate tramite tali servizi per evitare l'addebito di addebiti.

Se hai altre domande, contattaci all'indirizzo aws-codecatalyst-service@amazon.com o tramite il Support Center nella CodeCatalyst console Amazon.

CodeCatalyst concetti

Acquisisci familiarità con i concetti chiave per velocizzare la collaborazione e lo sviluppo di applicazioni in Amazon CodeCatalyst. Questi concetti includono termini usati nel controllo del codice sorgente, nell'integrazione continua e nella distribuzione continua (CI/CD) e nella modellazione e configurazione dei processi di rilascio automatizzati.

Per ulteriori informazioni concettuali, vedere i seguenti argomenti:

- [Concetti relativi al repository di origine](#)
- [Concetti sui flussi di lavoro](#)

Argomenti

- [AWS Spazi ID Builder in CodeCatalyst](#)
- [Spazi che supportano la federazione delle identità in CodeCatalyst](#)
- [Progetti](#)
- [Blueprint](#)
- [Connessioni agli account](#)
- [Connessioni VPC](#)
- [AWS ID del costruttore](#)
- [Profili utente in CodeCatalyst](#)
- [Archivi di origine](#)
- [Commit](#)
- [Ambienti di sviluppo](#)
- [Flussi di lavoro](#)
- [Azioni](#)
- [Problemi](#)
- [Token di accesso personali \(\) PATs](#)
- [Connessioni personali](#)
- [Roles](#)

AWS Spazi ID Builder in CodeCatalyst

L'amministratore dello spazio invita gli utenti a CodeCatalyst farlo inviando e-mail di invito individuali dalla pagina dei membri. Utenti invitati o registrati per CodeCatalyst creare il proprio AWS Builder ID. Il profilo è gestito in AWS Builder ID e viene visualizzato come nome utente e informazioni del profilo nelle impostazioni utente in CodeCatalyst.

Spazi che supportano la federazione delle identità in CodeCatalyst

Utenti che sono stati aggiunti agli utenti e ai gruppi SSO per l'istanza IAM Identity Center, gestiti nell'archivio di identità e invitati nel tuo spazio tramite IAM Identity Center. L'amministratore dello spazio sincronizza la pagina CodeCatalyst dei membri per gli ultimi aggiornamenti. Gli utenti accedono utilizzando il portale di accesso SSO configurato nell'istanza aziendale IAM Identity Center. Gli spazi che supportano la federazione delle identità sono collegati all'istanza dell'archivio di identità tramite l'applicazione Identity Center e la relativa mappatura all'ID dell'archivio di identità.

Progetti

Un progetto rappresenta uno sforzo collaborativo CodeCatalyst che supporta i team e le attività di sviluppo. Dopo aver creato un progetto, puoi aggiungere, aggiornare o rimuovere utenti e risorse, personalizzare la dashboard del progetto e monitorare lo stato di avanzamento del lavoro del team. Puoi avere più progetti all'interno di uno spazio.

Per ulteriori informazioni sui progetti, vedere [Organizza il lavoro con i progetti in CodeCatalyst](#).

Blueprint

Un blueprint è un sintetizzatore di progetti che genera ed estende automaticamente i file di supporto delle applicazioni e le dipendenze, oltre a creare il CodeCatalyst progetto nella console. Scegliete un tipo di progetto da una selezione di progetti in CodeCatalyst, visualizzate il file README e visualizzate in anteprima l'archivio del progetto e le risorse che verranno generate. Il progetto viene generato dalla configurazione di base specificata dal blueprint. Periodicamente si esegue la sintesi nel blueprint del progetto, che aggiorna i file di progetto, ad esempio le dipendenze software, e rigenera le risorse. I progetti utilizzano uno strumento chiamato Projen per sintetizzare i progetti sincronizzando gli ultimi aggiornamenti del progetto e generando file di supporto. Questi file possono includere `package.json`, `Makefile`, `eslint`, e altro in base al tipo e alla lingua dell'applicazione.

I blueprint di progetto possono generare file che supportano AWS risorse come costrutti CDK, CloudFormation modelli e modelli. AWS Serverless Application Model

Per ulteriori informazioni sui blueprint di progetto, vedere. [Creazione di un progetto completo con CodeCatalyst progetti](#)

Connessioni agli account

Una connessione all'account associa uno CodeCatalyst spazio al tuo Account AWS. Dopo aver configurato la connessione all'account, Account AWS viene reso disponibile lo spazio. Puoi quindi aggiungere ruoli IAM in CodeCatalyst modo che possa accedere alle risorse del tuo Account AWS. Puoi anche utilizzare questi ruoli per le tue azioni CodeCatalyst di flusso di lavoro.

Puoi limitare i progetti e le risorse che hanno accesso alle connessioni degli account abilitando le connessioni agli account con restrizioni del progetto. Sono connesse connessioni ad account con restrizioni del progetto a Account AWS cui possono accedere solo progetti specifici nello spazio. Ciò consente ai team presenti in uno spazio di limitare l'uso delle AWS risorse integrate Account AWS per progetto. Ad esempio, l'account utilizzato per i flussi di lavoro di distribuzione e le connessioni VPC in progetti specifici sarà disponibile solo con una connessione account limitata al progetto. Per ulteriori informazioni, consulta [Configurazione](#) delle connessioni di account con restrizioni del progetto.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#)

Connessioni VPC

Una connessione VPC è una CodeCatalyst risorsa che contiene tutte le configurazioni necessarie al flusso di lavoro per accedere a un VPC. Gli amministratori dello spazio possono aggiungere le proprie connessioni VPC nella console CodeCatalyst Amazon per conto dei membri dello spazio. Aggiungendo una connessione VPC, i membri dello spazio possono eseguire azioni di flusso di lavoro e creare ambienti di sviluppo che rispettano le regole di rete e possono accedere alle risorse nel VPC associato.

Per ulteriori informazioni sulle connessioni VPC, consulta Managing [Amazon Virtual Private Clouds](#) nella CodeCatalyst Administrator Guide.

AWS ID del costruttore

Un AWS Builder ID è un'identità personale che puoi utilizzare per registrarti e accedere ad altre CodeCatalyst applicazioni partecipanti. Non è la stessa cosa di un Account AWS. Il tuo AWS Builder ID gestisce i metadati come l'alias utente e l'indirizzo email. Il tuo AWS Builder ID è un'identità unica che supporta gli utenti in tutti gli spazi di. CodeCatalyst Per informazioni sull'accesso al tuo profilo AWS Builder ID, consulta [Aggiornamento di un profilo](#) Per ulteriori informazioni su AWS Builder ID, consulta [AWS Builder ID](#) nel. Riferimenti generali di AWS

Per ulteriori informazioni sulla registrazione e l'accesso, consulta [Configura e accedi a CodeCatalyst](#)

Profili utente in CodeCatalyst

Puoi accedere al tuo profilo CodeCatalyst utente scegliendo l'opzione del profilo dal menu a discesa sotto le iniziali di accesso in qualsiasi pagina. CodeCatalyst Puoi creare token di accesso personali (PATs) dalla pagina del tuo profilo, ma puoi visualizzarli o eliminarli solo utilizzando il. PATs AWS CLI Il tuo nome utente è l'alias che hai scelto al momento della registrazione. Non puoi cambiare il tuo nome utente. Per visualizzare la pagina del profilo di un altro CodeCatalyst utente, vai alla scheda Membri del tuo progetto e scegli l'utente appropriato.

Puoi accedere al tuo AWS Builder ID visualizzando il tuo CodeCatalyst profilo e poi scegliendo di andare a AWS Builder ID. Verrai reindirizzato alla pagina del tuo profilo AWS Builder ID. Il nome completo, l'indirizzo email e la password del tuo profilo sono gestiti dal tuo ID AWS Builder e puoi modificare tali informazioni utilizzando la pagina AWS Builder ID. Hai inserito queste informazioni al momento della registrazione. Quando sei pronto per configurare MFA per utilizzare un'applicazione di autenticazione per l'accesso, utilizzerai la AWS pagina Builder ID. Per ulteriori informazioni sulla visualizzazione del profilo AWS Builder ID, consulta [Aggiornamento di un profilo](#)

Per ulteriori informazioni sulla registrazione e l'accesso, consulta [Configura e accedi a CodeCatalyst](#).

Archivi di origine

Un repository di sorgenti consente di archiviare in modo sicuro codice e file per il progetto. Memorizza anche la cronologia delle versioni dei file. Per impostazione predefinita, un archivio dei sorgenti viene condiviso con gli altri utenti del CodeCatalyst progetto. Puoi avere più di un archivio di sorgenti per un progetto. È possibile creare archivi di sorgenti per progetti in CodeCatalyst oppure scegliere di collegare un archivio di sorgenti esistente ospitato da un altro servizio se tale servizio è supportato da

un'estensione installata. Ad esempio, è possibile collegare un GitHub repository a un progetto dopo aver installato l'GitHub estensione Repositories. Per ulteriori informazioni, consultare [Memorizzazione del codice sorgente nei repository per un progetto in CodeCatalyst](#) e [Guida rapida: installazione di estensioni, connessione di provider e collegamento di risorse in CodeCatalyst](#).

I repository di origine sono anche il luogo in cui vengono archiviate le informazioni di configurazione per il CodeCatalyst progetto, ad esempio il file di configurazione che definisce gli attributi e le azioni del flusso di lavoro CI/CD. Se si crea il progetto utilizzando un blueprint, verrà creato un repository di origine con le informazioni di configurazione del progetto memorizzate al suo interno. Se si crea un progetto vuoto, è necessario creare un archivio di origine prima di poter creare risorse che richiedono informazioni di configurazione, come i flussi di lavoro.

Per ulteriori concetti che possono aiutarti a lavorare con gli archivi dei sorgenti e il controllo del codice sorgente, consulta. [Concetti relativi al repository di origine](#)

Commit

Un commit è una modifica a un file o a un set di file. Nella CodeCatalyst console Amazon, un commit salva le modifiche e le invia a un repository di origine. Il commit include informazioni sulla modifica, tra cui l'identità dell'utente che ha apportato la modifica, l'ora e la data della modifica, il titolo del commit e qualsiasi messaggio incluso sulla modifica. Per ulteriori informazioni, consulta [Comprendere le modifiche al codice sorgente con i commit in Amazon CodeCatalyst](#).

Nel contesto di un repository di origine in CodeCatalyst, i commit sono istantanee delle modifiche al contenuto del repository. Ogni volta che un utente esegue il commit e invia una modifica, CodeCatalyst salva le informazioni che includono chi ha eseguito la modifica, la data e l'ora del commit e le modifiche apportate nell'ambito del commit. Puoi anche aggiungere tag Git ai commit per aiutare a identificare commit specifici.

Per ulteriori informazioni sui commit, consulta. [Comprendere le modifiche al codice sorgente con i commit in Amazon CodeCatalyst](#)

Ambienti di sviluppo

Un ambiente di sviluppo è un ambiente di sviluppo basato su cloud che puoi utilizzare CodeCatalyst per lavorare rapidamente sul codice archiviato nei repository di origine del tuo progetto. Gli strumenti di progetto e le librerie di applicazioni inclusi nel tuo ambiente di sviluppo sono definiti da un devfile nel repository di origine del progetto. Se non avete un devfile nel vostro repository di origine, verrà

applicato automaticamente un devfile predefinito. Il devfile predefinito include strumenti per i linguaggi e i framework di programmazione più utilizzati. Per impostazione predefinita, un ambiente di sviluppo è configurato per avere un processore a 2 core, 4 GB di RAM e 16 GiB di storage persistente.

Flussi di lavoro

Un flusso di lavoro è una procedura automatizzata che descrive come creare, testare e distribuire il codice come parte di un sistema di integrazione e distribuzione continua (CI/CD). Un flusso di lavoro definisce una serie di passaggi, o azioni, da eseguire durante l'esecuzione di un flusso di lavoro. Un flusso di lavoro definisce anche gli eventi, o trigger, che causano l'avvio del flusso di lavoro. Per configurare un flusso di lavoro, è necessario creare un file di definizione del flusso di lavoro utilizzando l'editor [visivo o YAML](#) della CodeCatalyst console.

Tip

Per una rapida occhiata a come potresti utilizzare i flussi di lavoro in un progetto, [crea un progetto con un blueprint](#). Ogni blueprint implementa un flusso di lavoro funzionante che puoi rivedere, eseguire e sperimentare.

Per ulteriori informazioni sui flussi di lavoro, consulta [Crea, testa e distribuisce con i flussi di lavoro](#).

Azioni

Un'azione è l'elemento costitutivo principale di un flusso di lavoro e definisce un'unità logica di lavoro, o attività, da eseguire durante l'esecuzione di un flusso di lavoro. In genere, un flusso di lavoro include più azioni eseguite in sequenza o in parallelo a seconda di come le hai configurate.

Per ulteriori informazioni sulle azioni, consulta [Configurazione delle azioni del flusso di lavoro](#).

Problemi

Un problema è un record che tiene traccia del lavoro relativo al progetto. Puoi creare un problema per una funzionalità, un'attività, un bug o qualsiasi altro documento relativo al tuo progetto. Se utilizzi lo sviluppo agile, un problema può anche descrivere un'epopea o una storia utente.

Per ulteriori informazioni sui problemi, consulta [Tieni traccia e organizza il lavoro con problemi in CodeCatalyst](#)

Token di accesso personali () PATs

Un token di accesso personale (PAT) è simile a una password. È associato all'identità utente e può essere utilizzato in tutti gli spazi e i progetti di CodeCatalyst. Si utilizza PATs per accedere a CodeCatalyst risorse che includono ambienti di sviluppo integrati (IDEs) e repository di sorgenti basati su Git. PATs ti rappresentano CodeCatalyst e puoi gestirle nelle tue impostazioni utente. Un utente può avere più di un PAT. I token di accesso personali vengono visualizzati solo una volta. Come procedura consigliata, assicurati di archivarli in modo sicuro sul tuo computer locale. Per impostazione predefinita, PATs scadono dopo un anno.

Per ulteriori informazioni su PATs, vedere [Concedi agli utenti l'accesso al repository con token di accesso personali](#).

Connessioni personali

Una connessione personale è un'autorizzazione tra la tua CodeCatalyst identità e il tuo provider di origine esterno, ad esempio GitHub. Le connessioni personali vengono utilizzate per consentire a un CodeCatalyst utente di aggiungere repository di sorgenti di terze parti. Ad esempio, è possibile connettere un GitHub repository a uno CodeCatalyst spazio. Un'applicazione connettore installata viene installata nell' GitHub account per essere utilizzata con i repository designati dal proprietario dell'account. È possibile creare una connessione personale per un'identità utente (CodeCatalyst alias) in tutti gli spazi per un tipo di provider specifico, ad esempio. GitHub Le connessioni personali sono associate al tuo AWS Builder ID o al tuo utente SSO.

Per ulteriori informazioni, consulta [Accesso alle GitHub risorse con connessioni personali](#).

Roles

Un ruolo definisce l'accesso di un utente alle risorse per un progetto o uno spazio e le azioni che l'utente può intraprendere. Scegli il ruolo di un utente quando lo inviti a un progetto. Esistono ruoli a livello di spazio e ruoli a livello di progetto in. CodeCatalyst Un utente con un ruolo amministrativo al livello corretto può modificare i ruoli assegnati. Ad esempio, un utente con il ruolo di amministratore di progetto per un progetto ha il pieno controllo su quel progetto e può modificare i ruoli degli utenti in quel progetto. Per informazioni sui ruoli disponibili e sulle autorizzazioni di ciascun ruolo, vedere [Concessione dell'accesso con ruoli utente](#).

Per ulteriori informazioni sui ruoli, consulta [Concessione dell'accesso con ruoli utente](#).

Configura e accedi a CodeCatalyst

È possibile configurare due tipi di spazio CodeCatalyst: spazi che supportano gli utenti AWS Builder ID e creazione di uno spazio che supporti la federazione delle identità, in cui gli utenti e i gruppi SSO vengono gestiti in IAM Identity Center. Gli utenti in uno spazio AWS Builder ID accedono CodeCatalyst con il proprio ID AWS Builder e gli utenti in uno spazio configurato per la federazione delle identità accedono CodeCatalyst utilizzando il portale SSO dell'azienda associata allo spazio.

Note

CodeCatalyst i nomi utente hanno una lunghezza minima di 3 e una lunghezza massima di 100 caratteri. I nomi utente forniti più lunghi di 100 caratteri verranno troncati. Ciò può comportare un nome utente che sembra essere un duplicato di un altro nome utente di 100 caratteri. Per ulteriori informazioni, consulta [Non riesco ad accedere al mio spazio BID come nuovo utente o non posso essere aggiunto come nuovo utente SSO a causa del nome utente troncato](#).

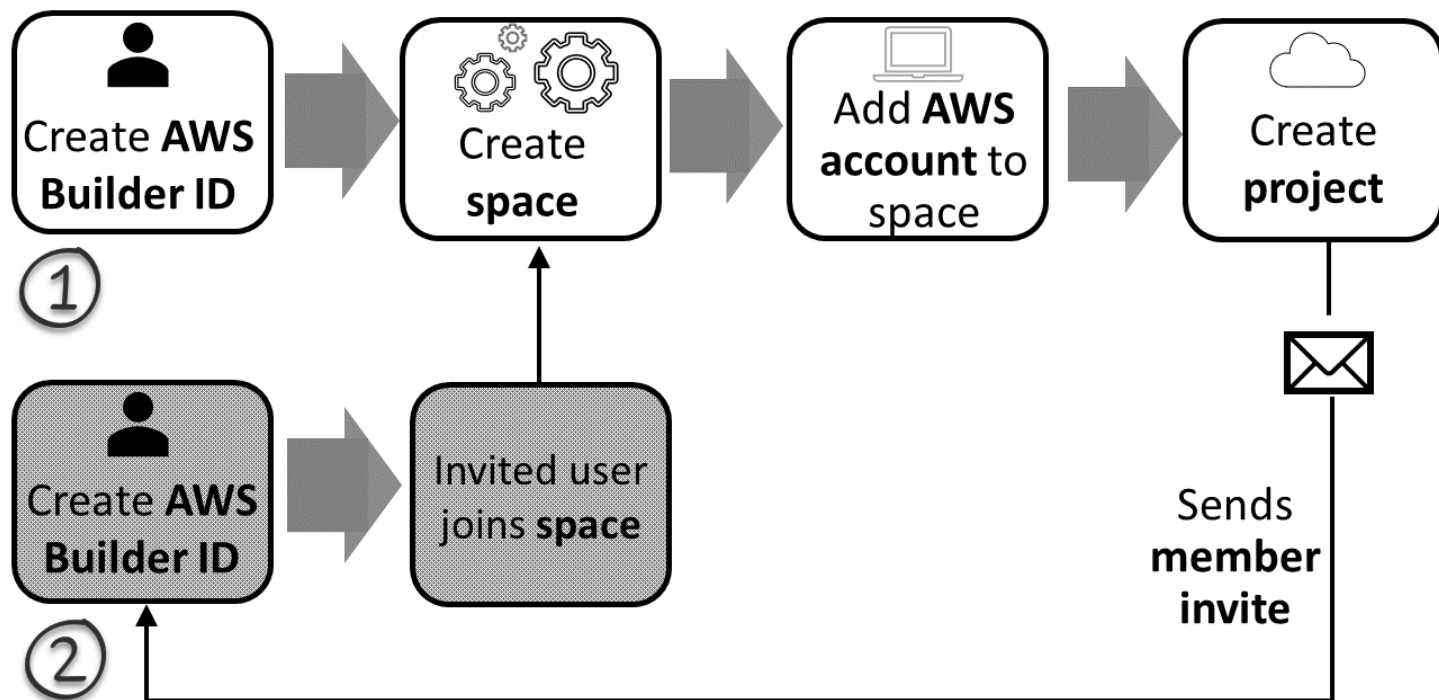
I passaggi per configurare e amministrare uno spazio AWS Builder ID sono forniti in questa guida. Per lavorare con uno spazio CodeCatalyst AWS Builder ID, devi configurarlo CodeCatalyst utilizzando le impostazioni utente e l'ID AWS Builder che usi per accedere. CodeCatalyst

I passaggi per configurare e amministrare uno spazio che supporti la federazione delle identità sono disponibili nella Guida per l'CodeCatalyst amministratore. Per lavorare con spazi configurati per la federazione delle identità, consulta [Configurazione e amministrazione CodeCatalyst degli spazi](#) nella Amazon CodeCatalyst Administrator Guide.

Questa sezione fornisce due percorsi comuni per configurare l'utilizzo in Amazon CodeCatalyst con uno spazio ID AWS Builder: creare uno spazio e un progetto come primo utente e accettare un invito a uno spazio o progetto esistente. Questi flussi di lavoro di configurazione sono necessariamente molto diversi. Il diagramma seguente mostra entrambi i processi di registrazione come segue:

1. Nel primo caso, crei e configuri uno spazio per la tua azienda, il team o il gruppo e crei un progetto prima di invitare altri a utilizzare queste risorse. Un Account AWS modulo deve essere fornito ai fini della fatturazione, dove puoi comunque impostare come impostazione predefinita il livello gratuito.

2. Nel secondo caso, se ti iscrivi CodeCatalyst accettando un invito a un progetto, qualcun altro ha già creato uno spazio e un progetto per te. Tuttavia, vorrai comunque configurare il tuo profilo in modo da essere pronto per iniziare a lavorare con gli altri.



Tip

CodeCatalyst utilizza gli spazi per raggruppare progetti e risorse. Quando ti iscrivi per la prima volta CodeCatalyst, ti verrà richiesto di creare uno spazio oltre a un progetto.

Sia che ti registri per creare uno spazio e un progetto o per accettare un invito, creerai un AWS Builder ID che utilizzerai per accedere. CodeCatalyst Per creare un AWS Builder ID, fornisci il nome completo, la password e l'indirizzo email che usi per accedere alle AWS applicazioni. CodeCatalyst Dopo questo punto, utilizzerai l'email e la password per accedere. Puoi utilizzare questo AWS Builder ID anche per accedere ad altre applicazioni che utilizzano le credenziali AWS Builder ID.

In CodeCatalyst e in AWS Builder ID, viene generato un profilo in base alle informazioni di accesso. Il tuo profilo contiene CodeCatalyst le tue preferenze per la lingua e le impostazioni di notifica nei tuoi CodeCatalyst progetti.

 Tip

Se riscontri problemi durante la registrazione al tuo CodeCatalyst profilo Amazon, segui i passaggi indicati in quella pagina. Se hai bisogno di ulteriore assistenza, consulta [Problemi di registrazione](#).

Argomenti

- [Creazione di un nuovo ruolo spaziale e di sviluppo \(a partire senza invito\)](#)
- [Accettazione di un invito e creazione di un AWS Builder ID](#)
- [Accesso con un AWS Builder ID](#)
- [Accesso con SSO](#)
- [Visualizzazione di tutti gli spazi e i progetti per un utente](#)
- [Visualizzazione e gestione dei CodeCatalyst profili](#)
- [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#)

Creazione di un nuovo ruolo spaziale e di sviluppo (a partire senza invito)

Puoi iscriverti ad Amazon CodeCatalyst senza un invito a uno spazio o progetto esistente. Quando lo farai, creerai uno spazio e un progetto dopo aver creato il tuo AWS Builder ID. Come parte della creazione di uno spazio, dovrai aggiungerne uno Account AWS per scopi di fatturazione.

 Tip

Se riscontri problemi durante la registrazione al tuo CodeCatalyst profilo Amazon, segui i passaggi indicati in quella pagina. Se hai bisogno di ulteriore assistenza, consulta [Problemi di registrazione](#).

Ecco un possibile flusso per un utente che inizia CodeCatalyst senza un invito a un progetto o a uno spazio.

Mary Major è una sviluppatrice interessata CodeCatalyst e decide di provarlo. Accede alla CodeCatalyst console e sceglie l'opzione per registrarsi e creare un AWS Builder ID. Mary fornisce

un indirizzo email e una password per creare il suo AWS Builder ID. Potrà usare il suo AWS Builder ID per accedere ad altre CodeCatalyst applicazioni. Quando le viene chiesto di scegliere un alias, specifica MaryMajor come nome CodeCatalyst utente che verrà visualizzato CodeCatalyst e che gli altri membri del progetto useranno @mention Mary.

Successivamente, Mary viene automaticamente indirizzata a creare uno spazio. Come parte di questo flusso, a Mary viene chiesto di Account AWS associare uno allo spazio che sta creando in modo da poter vedere il codice di esempio nella creazione e distribuzione del suo primo progetto. Aggiunge queste informazioni e crea il suo spazio, dove sceglie l'opzione di creare un ruolo di sviluppo in anteprima che può essere utilizzato per i progetti nel suo nuovo spazio. Mary sceglie di creare un progetto, quindi visualizza un elenco di progetti per i progetti. Dopo aver esaminato le informazioni per i progetti disponibili, decide di provare il modello di applicazione Web Modern a tre livelli per il suo primo progetto. Compila i campi obbligatori e crea il progetto. Non appena il progetto è pronto, viene indirizzata a una pagina di riepilogo del progetto che include le attività recenti, nonché collegamenti al codice del progetto e al flusso di lavoro che crea e distribuisce automaticamente quel codice. Esplora sia il codice che il flusso di lavoro, inclusa la visualizzazione dell'applicazione web di esempio distribuita. Apprezzando ciò che vede, decide di invitare alcuni dei suoi colleghi al progetto per iniziare a esplorare. CodeCatalyst

Quando ha un momento, Mary configura il suo ID AWS Builder per l'accesso CodeCatalyst con l'autenticazione a più fattori (MFA). Con la MFA configurata, Mary può accedere CodeCatalyst utilizzando una combinazione della sua CodeCatalyst password e un passcode o token da un'app di autenticazione di terze parti approvata.

Creazione di un nuovo spazio e ruoli IAM

Segui questi passaggi per registrarti al tuo CodeCatalyst profilo Amazon, creare uno spazio e aggiungere un account, un ruolo di supporto e un ruolo di sviluppatore per il tuo spazio.

La procedura finale crea e aggiunge il ruolo di sviluppatore. Il ruolo di sviluppatore è un ruolo AWS IAM che consente ai CodeCatalyst flussi di lavoro di accedere alle AWS risorse. Il ruolo di sviluppatore è un ruolo di servizio utilizzato per la gestione Servizi AWS e verrà creato nell'account a cui è stato effettuato l'accesso. Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Il ruolo avrà un nomeCodeCatalystWorkflowDevelopmentRole-*spaceName*. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere[Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName*del servizio](#).

 Note

Come procedura consigliata in materia di sicurezza, assegna l'accesso amministrativo solo agli utenti amministrativi e agli sviluppatori che devono gestire l'accesso alle AWS risorse dello spazio.

Prima di iniziare, devi essere pronto a fornire un Account AWS ID per un account con privilegi amministrativi. Tieni a portata di mano il tuo ID a 12 cifre Account AWS . Per informazioni su come trovare il tuo Account AWS ID, vedi Il [tuo Account AWS ID e il suo alias](#).

Per registrarti come nuovo utente

1. Prima di iniziare a utilizzare la CodeCatalyst console Console di gestione AWS, apri e assicurati di aver effettuato l'accesso con lo stesso Account AWS che desideri utilizzare per creare lo spazio.
2. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Nella pagina di benvenuto, scegli Iscriviti. Viene visualizzata la pagina Crea il tuo ID AWS Builder. Il tuo AWS Builder ID è un'identità che crei per accedere. Non è la stessa cosa di un Account AWS.
4. In Il tuo indirizzo e-mail, inserisci l'indirizzo e-mail a cui desideri associarti CodeCatalyst. Quindi scegli Successivo.
5. In Il tuo nome, inserisci il nome e il cognome che desideri vengano visualizzati nelle applicazioni in cui utilizzi il tuo AWS Builder ID. Gli spazi sono consentiti. Questo sarà il nome del tuo profilo AWS Builder ID, ad esempio Mary Major. Puoi cambiare il nome in un secondo momento.

Scegli Next (Successivo). Viene visualizzata la pagina di verifica dell'e-mail.

6. Un codice di verifica verrà inviato all'indirizzo e-mail specificato. Inserisci questo codice nel campo Codice di verifica, quindi scegli Verifica. Se non ricevi il codice dopo 5 minuti e non riesci a trovarlo nella cartella spam o posta indesiderata, scegli Reinvia codice.
7. Una volta verificato il codice, inserisci una password che soddisfi i requisiti in Password e Conferma password.

Seleziona la casella di controllo per confermare l'accordo con il AWS cliente e i Termini di AWS servizio, quindi scegli Crea AWS Builder ID.

8. Nella pagina Crea il tuo CodeCatalyst alias, inserisci un alias in cui desideri utilizzare come identificatore utente univoco. CodeCatalyst Sceglie una versione abbreviata del tuo nome senza

spazi, ad esempio. MaryMajor CodeCatalyst Gli altri utenti lo useranno per @mention you nei commenti e nelle pull request. Il tuo CodeCatalyst profilo conterrà sia il nome completo del tuo ID AWS Builder che il tuo CodeCatalyst alias. Non puoi modificare il tuo CodeCatalyst alias in un secondo momento.

Il tuo nome completo e il tuo alias verranno visualizzati in aree diverse di CodeCatalyst. Ad esempio, il nome del tuo profilo viene visualizzato per l'attività elencata nel feed delle attività, ma i membri del progetto useranno il tuo alias to @mention you.

Scegli Next (Successivo). La pagina viene aggiornata per mostrare la sezione Crea il tuo CodeCatalyst spazio.

9. In Assegna un nome al tuo spazio, inserisci il nome del tuo spazio. Non puoi modificarlo in un secondo momento.

 Note

I nomi degli spazi devono essere univoci CodeCatalyst. Non è possibile riutilizzare i nomi degli spazi eliminati.

10. Nel menu a Regione AWS discesa, scegli la regione in cui desideri archiviare lo spazio e i dati del progetto. Non puoi cambiarlo in un secondo momento.
11. Scegli Next (Successivo). La pagina si aggiorna per mostrare la pagina per aggiungere un Account AWS. Questo account verrà utilizzato come account di fatturazione per lo spazio.
12. In Account AWS ID, inserisci l'ID a dodici cifre dell'account che desideri connettere al tuo spazio.

Nel token di verifica dell'AWS account, copia l'ID del token generato. Il token viene copiato automaticamente per te, ma potresti volerlo archiviare mentre approvi la richiesta di AWS connessione.

13. Scegli Vai alla AWS console per verificare.
14. La pagina Verify Amazon CodeCatalyst Space si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

Nel Console di gestione AWS, assicurati di scegliere lo stesso Regione AWS luogo in cui desideri creare il tuo spazio.

Per accedere direttamente alla pagina, accedi ad Amazon CodeCatalyst Spaces nella Console di gestione AWS pagina <https://console.aws.amazon.com/codecatalyst/> iniziale/.

Il campo relativo al token di verifica in Console di gestione AWS viene compilato automaticamente con il token generato in CodeCatalyst.

15. (Facoltativo) In Livelli a pagamento autorizzati, scegli Autorizza livelli a pagamento (Standard, Enterprise) per attivare i livelli a pagamento per il tuo account di fatturazione.

 Note

Ciò non comporta l'upgrade del livello di fatturazione a un livello a pagamento. Tuttavia, questo lo configura Account AWS in modo che tu possa modificare il livello di fatturazione per il tuo spazio in qualsiasi momento. CodeCatalyst Puoi attivare i livelli a pagamento in qualsiasi momento. Senza apportare questa modifica, lo spazio è in grado di utilizzare solo il livello gratuito.

16. Scegli Verifica spazio.

Viene visualizzato un messaggio di verifica dell'account che indica che l'account è stato aggiunto allo spazio.

17. Rimani sulla pagina Verify Amazon CodeCatalyst Space. Scegli il seguente link: Per aggiungere ruoli IAM per questo spazio, visualizza i dettagli dello spazio.

La pagina delle connessioni con i dettagli CodeCatalyst dello spazio si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

18. Torna alla CodeCatalyst pagina, quindi scegli Avanti.
19. Durante la creazione dello spazio viene visualizzato un messaggio di stato. Quando lo spazio viene creato, viene visualizzato CodeCatalyst il seguente messaggio: Lo spazio è pronto. L'ultimo passaggio è creare un progetto. . Puoi effettuare una delle seguenti operazioni:
 - Scegli Skip per ora.
 - Scegli Crea il tuo primo progetto per il tuo spazio. Per un tutorial che mostra come creare un progetto con un blueprint, vedi [Tutorial: creazione di un progetto con il moderno modello di applicazione web a tre livelli](#)

 Note

Se viene visualizzato un errore di autorizzazione o un banner, aggiorna la pagina e prova a visualizzarla nuovamente.

Per creare e aggiungere CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Prima di iniziare a utilizzare la CodeCatalyst console, apri il Console di gestione AWS, quindi assicurati di aver effettuato l'accesso con lo stesso Account AWS nome del tuo spazio.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
4. Scegli il link relativo alla Account AWS posizione in cui desideri creare il ruolo. Viene visualizzata la pagina dei Account AWS dettagli.
5. Scegli Gestisci ruoli da Console di gestione AWS.

La pagina Aggiungi il ruolo IAM CodeCatalyst allo spazio Amazon si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

6. Scegli Crea il ruolo di amministratore CodeCatalyst dello sviluppo in IAM. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di fiducia per il ruolo di sviluppo. Il ruolo avrà un nomeCodeCatalystWorkflowDevelopmentRole-*spaceName*. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere[Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName*del servizio](#).

 Note

Questo ruolo è consigliato solo per gli account sviluppatore e utilizza la politica AdministratorAccess AWS gestita, che gli consente l'accesso completo alla creazione di nuove politiche e risorse Account AWS.

7. Scegli Crea ruolo di sviluppo.

8. Nella pagina delle connessioni, in Ruoli IAM disponibili per CodeCatalyst, visualizza il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo nell'elenco dei ruoli IAM aggiunti al tuo account.
9. Per tornare al tuo spazio, scegli Vai su Amazon CodeCatalyst.

Per creare e aggiungere CodeCatalyst `AWSRoleForCodeCatalystSupport`

1. Prima di iniziare a utilizzare la CodeCatalyst console, apri il Console di gestione AWS, quindi assicurati di aver effettuato l'accesso con lo stesso Account AWS nome del tuo spazio.
2. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
3. Scegli il link relativo alla Account AWS posizione in cui desideri creare il ruolo. Viene visualizzata la pagina dei Account AWS dettagli.
4. Scegli Gestisci ruoli da Console di gestione AWS.

La pagina Aggiungi il ruolo IAM CodeCatalyst allo spazio Amazon si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

5. In Dettagli CodeCatalyst dello spazio, scegli Aggiungi ruolo CodeCatalyst Support. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di fiducia per il ruolo di sviluppo in anteprima. Il ruolo avrà un nome `AWSRoleForCodeCatalystSupport` con un identificatore univoco aggiunto. Per ulteriori informazioni sul ruolo e sulla politica dei ruoli, vedere [Comprensione del ruolo `AWSRoleForCodeCatalystSupport` del servizio](#)
6. Nella pagina Aggiungi ruolo per CodeCatalyst Support, lascia selezionata l'impostazione predefinita, quindi scegli Crea ruolo.
7. In Ruoli IAM disponibili per CodeCatalyst, visualizza il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo nell'elenco dei ruoli IAM aggiunti al tuo account.
8. Per tornare al tuo spazio, scegli Vai su Amazon CodeCatalyst.

Dopo aver creato il tuo ID AWS Builder, creato il tuo primo spazio e aggiunto un account, puoi creare un progetto. Per ulteriori informazioni, consulta [Creare un progetto](#). Se è la prima volta che lo usi CodeCatalyst, ti consigliamo di iniziare con [Tutorial: creazione di un progetto con il moderno modello di applicazione web a tre livelli](#).

Accettazione di un invito e creazione di un AWS Builder ID

Puoi iscriverti ad Amazon CodeCatalyst come parte dell'accettazione di un invito a un progetto o a uno spazio. Come parte dell'accettazione dell'invito, ti verrà richiesto di creare un AWS Builder ID. Utilizzerai il tuo AWS Builder ID per accedere alle risorse in CodeCatalyst.

Tip

Se hai bisogno di ulteriore assistenza, consulta [Problemi di registrazione](#).

Ecco un possibile flusso per un utente che inizia CodeCatalyst con un invito a un progetto o a uno spazio.

Saanvi Sarkar è uno sviluppatore che ha ricevuto un invito a partecipare a un CodeCatalyst progetto come amministratore del progetto. Saanvi accetta l'invito, che apre la pagina di accesso per CodeCatalyst. Saanvi sceglie di registrarsi e fornisce un indirizzo e-mail e una password per creare il suo Builder ID. AWS Saanvi potrà usare il suo AWS Builder ID per accedere ad altre applicazioni. Successivamente, potrà modificare il suo profilo per cambiare l'indirizzo email o la password di accesso. Quando viene chiesto di scegliere un alias, Saanvi specifica `SaanviSarkar` come CodeCatalyst alias che verrà visualizzato in CodeCatalyst e che gli altri membri del progetto useranno per @mention Saanvi. Dopo la registrazione, Saanvi potrà utilizzare le sue credenziali di accesso anche per altre applicazioni che utilizzano le credenziali Builder ID. AWS.

Al termine della registrazione, Saanvi si unisce automaticamente al progetto e allo spazio specificati nell'invito. L'invito fornisce anche autorizzazioni predeterminate per i suoi ruoli nel progetto e nello spazio. Nelle impostazioni del progetto, l'alias di Saanvi viene visualizzato nell'elenco dei membri con il ruolo assegnato nel progetto. Per lavorare con i repository dei sorgenti in CodeCatalyst, Saanvi impiega qualche minuto per creare un token di accesso personale (PAT). Il PAT verrà utilizzato in CodeCatalyst per l'autenticazione quando si apportano modifiche alle azioni che richiedono un token di autenticazione.

Quando Saanvi lavora a un progetto, il suo alias verrà elencato nel registro delle attività lavorative del progetto. I problemi e i commenti di Saanvi mostreranno il suo alias, dove gli altri membri del progetto potranno inviarle un @mention nelle risposte. Per @mention un altro membro del progetto, Saanvi cerca il suo alias sul suo profilo in CodeCatalyst.

Quando ha un momento, Saanvi configura il suo AWS Builder ID per l'accesso a CodeCatalyst con l'autenticazione a più fattori (MFA). Con l'MFA configurata, Saanvi può accedere a CodeCatalyst.

utilizzando una combinazione della sua CodeCatalyst password e un passcode o token da un'app di autenticazione di terze parti approvata.

Accettazione di un invito e creazione di un Builder ID AWS

Quando vieni invitato a un progetto o a uno spazio su Amazon CodeCatalyst, riceverai un'email da `notify@codecatalyst.aws` che ti chiede di accettare l'invito. Se disponi già di un AWS Builder ID e hai effettuato l'accesso CodeCatalyst, selezionando **Accetta invito** si aprirà automaticamente il progetto o lo spazio in una scheda del browser. Se non hai effettuato l'accesso alla console ma disponi di un AWS Builder ID, verrai indirizzato alla pagina di accesso. Per ulteriori informazioni, consulta [Accesso con un AWS Builder ID](#).

Se non disponi di un AWS Builder ID, selezionando **Accetta invito** verrai indirizzato alla pagina di accesso, dove dovrai scegliere l'opzione per creare il tuo ID Builder. AWS

Important

Accettando un invito e aprendo i repository in Dev Environments, gli script possono essere eseguiti con accesso alle tue credenziali. CodeCatalyst Assicurati di fidarti della fonte prima di procedere.

Per accettare un invito e creare un AWS Builder ID

1. Nell'e-mail di invito, scegli **Accetta invito**.
2. Nella pagina di accesso, scegli **Non sei registrato? Crea il tuo AWS Builder ID**.

Tip

Il tuo AWS Builder ID è un'identità che crei per accedere. Non è la stessa cosa di un Account AWS.

3. Nella pagina **Crea il tuo ID AWS Builder**, in **Indirizzo e-mail**, inserisci l'indirizzo email che desideri utilizzare per il tuo ID AWS Builder.

Nel campo **Il tuo nome**, fornisci il nome e il cognome che desideri vengano visualizzati nelle applicazioni in cui utilizzi il tuo AWS Builder ID. Gli spazi sono consentiti. Questo sarà il nome del tuo profilo AWS Builder ID, ad esempio **Mary Major**. Puoi cambiare il nome in un secondo momento.

Scegli Next (Successivo).

Un codice di verifica verrà inviato all'indirizzo e-mail specificato. Inserisci questo codice nel campo Codice di verifica, quindi scegli Verifica. Se non ricevi il codice dopo 5 minuti e non riesci a trovarlo nella cartella spam o posta indesiderata, scegli Reinvia codice.

4. Una volta verificato il codice, inserisci una password che soddisfi i requisiti in Password e Conferma password.
5. Scegli Crea AWS Builder ID.
6. Nella pagina Crea il tuo alias, inserisci un alias in cui desideri utilizzare come identificatore utente univoco. CodeCatalyst Sceglie una versione abbreviata del tuo nome senza spazi, ad esempio. MaryMajor CodeCatalyst Gli altri utenti lo useranno per @mention you nei commenti e nelle pull request. Il tuo CodeCatalyst profilo conterrà sia il nome completo del tuo ID AWS Builder che il tuo CodeCatalyst alias. Non puoi cambiare il tuo alias. CodeCatalyst

Il tuo nome completo e il tuo alias verranno visualizzati in aree diverse di CodeCatalyst. Ad esempio, il nome del tuo profilo viene visualizzato per l'attività elencata nel feed delle attività, ma i membri del progetto useranno il tuo alias to @mention you.

Scegli Crea alias. Verrai indirizzato al progetto o allo spazio a cui sei stato invitato.

Accesso con un AWS Builder ID

Segui questi passaggi per accedere al tuo CodeCatalyst profilo Amazon.

Note

Hai già registrato un dispositivo per l'autenticazione a più fattori (MFA)? Ti consigliamo vivamente di configurare l'MFA in Amazon CodeCatalyst per aumentare la tua sicurezza. Per ulteriori informazioni, consulta [Come registrare un dispositivo per l'utilizzo con l'autenticazione a più fattori](#).

Per accedere con il tuo AWS Builder ID

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Inserisci il tuo indirizzo e-mail. Facoltativamente, scegli Salva il mio indirizzo e-mail se desideri salvare il tuo indirizzo e-mail per accessi futuri. Scegli Continua.

3. Inserisci la password. Selezionare Sign in (Accedi). Se non ricordi la password, segui i passaggi indicati. [Ho dimenticato la mia password](#)

Dispositivi attendibili

Dopo aver scelto l'opzione Questo è un dispositivo affidabile nella pagina di accesso, Amazon CodeCatalyst considera autorizzati tutti gli accessi futuri da quel dispositivo. Amazon non CodeCatalyst offrirà un'opzione per inserire un codice MFA fintanto che utilizzi quel dispositivo affidabile. Alcune eccezioni includono l'accesso da un nuovo browser o quando al dispositivo è stato assegnato un indirizzo IP sconosciuto.

Accesso con SSO

Segui questi passaggi per utilizzare l'SSO per accedere ad Amazon CodeCatalyst.

Per accedere invece con il tuo AWS Builder ID, consulta. [Accesso con un AWS Builder ID](#)

Per accedere con SSO

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. In Scegli un'opzione di accesso, scegli Usa Single Sign-On (SSO).
3. Nel nome dell'applicazione AWS Identity Center, inserisci il nome dell'applicazione fornito dall'amministratore della federazione delle identità.
4. Scegli Continua con IAM Identity Center.

Visualizzazione di tutti gli spazi e i progetti per un utente

Puoi visualizzare un elenco dei tuoi spazi e progetti nella home page dell'utente. La home page dell'utente mostra un elenco di ogni spazio a cui l'utente appartiene, il ruolo dell'utente in quello spazio, ad esempio l'amministratore dello spazio, e i progetti in ogni spazio a cui l'utente appartiene.

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel browser, inserisci il seguente indirizzo: <https://codecatalyst.aws/home>

EnchantedForest
Space administrator

[Manage AWS Builder ID](#) [Create space](#)

Filter spaces

Projects (2)

- WildWaves**
 - Pull requests
 - Workflows
 - Source repositories
 - Environments
- FracturedFairyTales**
 - Pull requests
 - Workflows
 - Source repositories
 - Environments

org
Space administrator

[Create project](#)

Projects (4)

- migration**
 - Pull requests
 - Workflows
 - Source repositories
 - Environments
- test**
 - Pull requests
 - Workflows
 - Source repositories
 - Environments
- 12597**
 - Pull requests
 - Workflows
 - Source repositories
 - Environments

AnyCompany
Space member

[Create project](#)

Projects (1)

- newproject**
 - Pull requests
 - Workflows
 - Source repositories
 - Environments

- Scegli lo spazio o il progetto che desideri aprire. Se non vedi uno spazio o un progetto che ti aspettavi di vedere, potresti dover accedere come utente diverso.

Visualizzazione e gestione dei CodeCatalyst profili

Puoi visualizzare i profili utente in Amazon CodeCatalyst per ottenere informazioni come indirizzi e-mail e CodeCatalyst alias. Puoi anche aggiornare il tuo profilo e il tuo ID AWS Builder. Se dimentichi la password, puoi richiederne la reimpostazione.

Visualizzazione del tuo CodeCatalyst profilo

Al momento della registrazione fornisci informazioni che verranno utilizzate come credenziali per accedere ad Amazon CodeCatalyst e che verranno gestite nel tuo profilo. Ciò include il nome, il soprannome e l'indirizzo e-mail che utilizzi per accedere. CodeCatalyst

Note

Il AWS Builder ID Nickname non è il tuo CodeCatalyst alias. Hai selezionato il tuo CodeCatalyst alias al momento della registrazione.

Per visualizzare il tuo profilo CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. In alto a destra, scegli la freccia accanto all'icona con la tua prima iniziale, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.
3. Per aggiornare l'indirizzo e-mail o la password del AWS Builder ID o per configurare l'MFA, scegli AWS Gestisci ID Builder. Si apre la pagina AWS Builder ID.

Visualizzazione del profilo di un altro utente CodeCatalyst

Per visualizzare il CodeCatalyst profilo di un altro utente

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella barra di navigazione laterale, scegli Impostazioni del progetto. Scegli la scheda Membri. Visualizza l'elenco dei membri del tuo CodeCatalyst progetto.
3. Scegli il nome del membro che desideri cercare o @mention. La pagina Le mie impostazioni mostra l'alias, l'indirizzo email e il nome completo dell'utente. Usa l' CodeCatalyst alias per @mention membri del progetto.

Note

Il nickname AWS Builder ID di un utente non è il suo CodeCatalyst alias. Hanno selezionato il loro CodeCatalyst alias al momento della registrazione.

Per visualizzare il profilo di un altro utente nel tuo progetto, scegli il suo nome nell'elenco.

Aggiornamento di un profilo

Nel CodeCatalyst, il tuo profilo è composto da informazioni personali gestite da AWS Builder ID e impostazioni gestite in CodeCatalyst.

- Il nome completo, l'indirizzo email e la password del tuo profilo sono gestiti da AWS Builder ID. Hai inserito queste informazioni al momento della registrazione. Quando configuri MFA per utilizzare un'app di autenticazione per l'accesso all'applicazione, accedi alla pagina CodeCatalyst Builder ID.AWS
- CodeCatalyst le impostazioni per il token di accesso personale (PAT), CodeCatalyst le notifiche e le preferenze di lingua sono gestite nella pagina Le mie impostazioni in. CodeCatalyst Per ulteriori informazioni, consulta [Concedi agli utenti l'accesso al repository con token di accesso personali](#).

Note

Puoi aggiornare il nome completo (nome CodeCatalyst visualizzato) e il nome del tuo AWS Builder ID. Tuttavia, non puoi modificare il tuo CodeCatalyst alias.

Aggiornamento di un ID AWS Builder o di un indirizzo email

Per aggiornare il tuo ID Builder AWS indirizzo email

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. In alto a destra, scegli la freccia accanto all'icona con la tua prima iniziale, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.
3. Nella pagina del profilo, scegli Manage AWS Builder ID. Si apre la pagina AWS Builder ID.
4. Sul lato sinistro della pagina, scegli I miei dati.
5. In Informazioni sul profilo, scegli Modifica per aggiornare il tuo nome o soprannome. Se non hai specificato un soprannome, il campo Soprannome riporta il nome del nome completo. Non è il tuo CodeCatalyst alias.

 Note

Questo aggiorna il nome completo e il nome del AWS Builder ID. Questo non aggiorna il tuo CodeCatalyst alias.

In Informazioni di contatto, scegli Modifica per aggiornare il tuo indirizzo e-mail.

 Note

Questo aggiorna l'indirizzo email che utilizzerai per accedere CodeCatalyst.

Modifica di una CodeCatalyst password associata a un AWS Builder ID

Utilizza le seguenti istruzioni per modificare la CodeCatalyst password Amazon associata al tuo ID AWS Builder.

 Note

Se utilizzi SSO per accedere a CodeCatalyst, contatta l'amministratore per modificare la password.

Per modificare la password CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. In alto a destra, scegli la freccia accanto all'icona con la tua prima iniziale, quindi scegli Profilo utente. Si apre la pagina Le CodeCatalyst mie impostazioni.
3. Nella pagina del profilo, scegli Manage AWS Builder ID. Si apre la pagina AWS Builder ID.
4. Sul lato sinistro della pagina, scegli Sicurezza.
5. Scegli Cambia password e segui le istruzioni.

Configurazione per l'utilizzo AWS CLI di CodeCatalyst

La CodeCatalyst console Amazon è il luogo in cui svolgerai la maggior parte delle tue attività quotidiane. Tuttavia, potresti voler impostare e configurare AWS CLI quando lavori con ambienti di sviluppo, token di accesso personali o registri di eventi. CodeCatalyst È necessario installare AWS CLI e configurare un profilo prima di poterlo utilizzare con. CodeCatalyst

Per configurare AWS CLI il CodeCatalyst

1. Installazione della versione più recente della AWS CLI. Se disponi già di una versione AWS CLI installata, assicurati che sia recente e includa i comandi per CodeCatalyst, e aggiornala se necessario. Per verificare che sia installata una versione che includa CodeCatalyst i comandi, apri un prompt dei comandi ed esegui il comando seguente:

```
aws codecatalyst help
```

Se vedi un elenco di CodeCatalyst comandi, significa che hai una versione che supporta CodeCatalyst. Se il comando non viene riconosciuto, aggiorna la versione di AWS CLI alla versione più recente. Per ulteriori informazioni, vedere [Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l' AWS Command Line Interface utente](#).

2. Esegui il `aws configure` comando per creare un profilo se non ne hai uno o se desideri utilizzare un profilo denominato specificamente per CodeCatalyst. Ti consigliamo di creare un profilo con nome da utilizzare in modo specifico CodeCatalyst, ma puoi anche utilizzare il profilo predefinito. Per ulteriori informazioni, consulta [Nozioni di base sulla configurazione](#).
3. Modifica il `config` file del profilo per aggiungere una sezione a cui connetterti CodeCatalyst come segue. Il file `config` si trova in `~/.aws/config` su Linux o macOS o in `C:\Users\USERNAME\.aws\config` su Windows.

```
[profile codecatalyst]
region = us-west-2
sso_session = codecatalyst

[sso-session codecatalyst]
sso_region = us-east-1
sso_start_url = https://view.awsapps.com/start
sso_registration_scopes = codecatalyst:read_write
```

4. Salvare il file.

5. Prima di provare a eseguire qualsiasi comando, apri un nuovo terminale o un prompt dei comandi di CodeCatalyst ed esegui il comando seguente per richiedere e recuperare le credenziali per eseguire i comandi. `aws codecatalyst codecatalyst` Sostituiscilo con il nome del tuo profilo, se necessario.

```
aws sso login --profile codecatalyst
```

Per visualizzare esempi di codecatalyst comandi, consulta i seguenti argomenti:

- [Concedi agli utenti l'accesso al repository con token di accesso personali](#)
- [Accesso agli eventi registrati utilizzando la registrazione degli eventi](#)

Tutorial introduttivi

Amazon CodeCatalyst offre diversi modelli per aiutarti a iniziare con i tuoi progetti. Puoi anche scegliere di iniziare con un progetto vuoto e aggiungervi risorse. Segui i passaggi di questi tutorial per scoprire alcuni dei modi in cui puoi lavorare. CodeCatalyst

Se è la prima volta che lo usi CodeCatalyst, ti suggeriamo di iniziare con. [Tutorial: creazione di un progetto con il moderno modello di applicazione web a tre livelli](#)

Note

Per seguire questi tutorial, devi prima completare la configurazione. Per ulteriori informazioni, consulta [Configura e accedi a CodeCatalyst](#).

Argomenti

- [Tutorial: creazione di un progetto con il moderno modello di applicazione web a tre livelli](#)
- [Tutorial: iniziare con un progetto vuoto e aggiungere manualmente le risorse](#)
- [Tutorial: Utilizzo delle funzionalità di intelligenza artificiale CodeCatalyst generativa per velocizzare il lavoro di sviluppo](#)

Per ulteriori tutorial incentrati su aree funzionali specifiche in, consulta: CodeCatalyst

- [Guida introduttiva alle notifiche di Slack](#)
- [Guida introduttiva agli archivi di CodeCatalyst origine e al modello di applicazione a pagina singola](#)
- [Guida introduttiva ai flussi di lavoro](#)
- [Guida introduttiva ai blueprint personalizzati](#)
- [Inizia a leggere la Amazon CodeCatalyst Action Developer Guide](#)

Per tutorial approfonditi, consulta:

- [Tutorial: caricare artefatti su Amazon S3](#)
- [Tutorial: Implementazione di un'applicazione serverless](#)
- [Tutorial: distribuire un'applicazione su Amazon ECS](#)

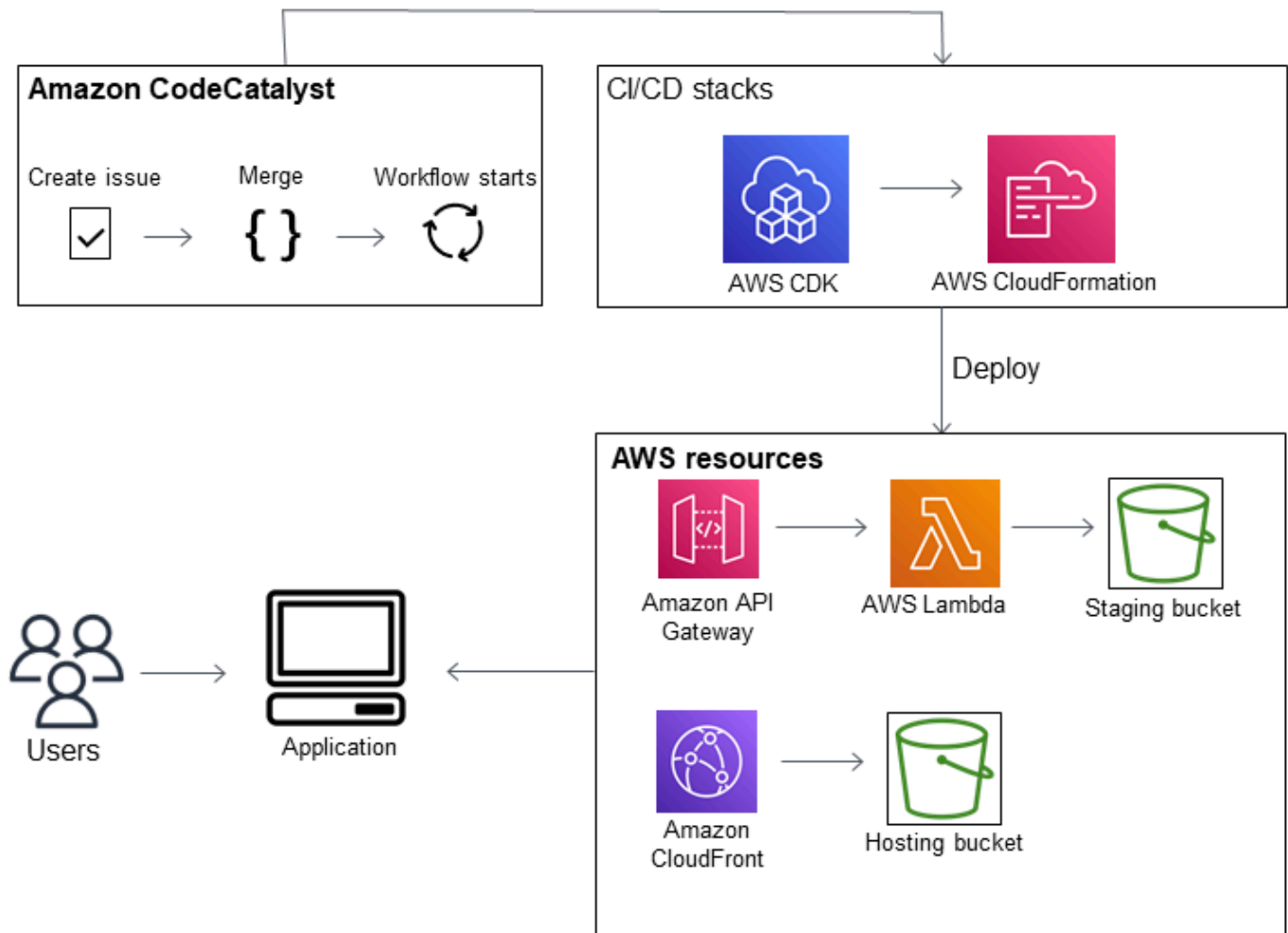
- [Tutorial: distribuzione di un'applicazione su Amazon EKS](#)
- [Tutorial: codice Lint usando un' GitHubazione](#)
- [Tutorial: creazione e aggiornamento di un'applicazione React](#)

Tutorial: creazione di un progetto con il moderno modello di applicazione web a tre livelli

Puoi iniziare più rapidamente a sviluppare software creando un progetto con un blueprint. Un progetto creato con un blueprint include le risorse necessarie, tra cui un archivio di sorgenti per gestire il codice e un flusso di lavoro per creare e distribuire l'applicazione. In questo tutorial, ti illustreremo come utilizzare il modello di applicazione Web Modern a tre livelli per creare un progetto in Amazon. CodeCatalyst Il tutorial include anche la visualizzazione dell'esempio distribuito, l'invito ad altri utenti a lavorarci e l'apporto di modifiche al codice con richieste pull che vengono automaticamente create e distribuite alle risorse connesse Account AWS quando la pull request viene unita. Laddove CodeCatalyst crea il progetto con report, feed di attività e altri strumenti, il blueprint crea AWS le risorse associate al progetto. Account AWS I file blueprint consentono di creare e testare un'applicazione moderna di esempio e di distribuirla nell'infrastruttura di. Cloud AWS

L'illustrazione seguente mostra come CodeCatalyst vengono utilizzati gli strumenti in per creare un problema di tracciamento, unire e generare automaticamente le modifiche, quindi avviare un flusso di lavoro nel CodeCatalyst progetto che esegua azioni per consentire AWS CDK e CloudFormation fornire l'infrastruttura.

Le azioni generano risorse nell'ambiente associato Account AWS e distribuiscono l'applicazione su una AWS Lambda funzione serverless con un endpoint API Gateway. L' AWS Cloud Development Kit (AWS CDK) azione converte uno o più AWS CDK stack in CloudFormation modelli e distribuisce gli stack sul tuo. Account AWS Le risorse nei tuoi stack includono CloudFront risorse Amazon per distribuire contenuti web dinamici, un'istanza Amazon DynamoDB per i dati delle tue applicazioni e i ruoli e le policy che supportano l'applicazione distribuita.



Quando si crea un progetto con il modello Modern Three Tier Web Application Blueprint, il progetto viene creato con le seguenti risorse:

Nel progetto: CodeCatalyst

- Un [archivio di sorgenti](#) con codice di esempio e flusso di lavoro YAML
- Un [flusso di lavoro](#) che crea e distribuisce il codice di esempio ogni volta che viene apportata una modifica al ramo predefinito
- Una bacheca dei problemi e un backlog che puoi utilizzare per pianificare e tenere traccia del lavoro
- Una suite di report di test con report automatici inclusi nel codice di esempio

Nella sezione associata Account AWS:

- Tre AWS CloudFormation stack che creano le risorse necessarie per l'applicazione.

Per maggiori dettagli sulle risorse che verranno create in AWS e CodeCatalyst come parte di questo tutorial, consulta [Documentazione di riferimento](#).

Note

Le risorse e gli esempi inclusi in un progetto dipendono dal blueprint selezionato. Amazon CodeCatalyst offre diversi modelli di progetto che definiscono le risorse relative al linguaggio o al framework definito. Per ulteriori informazioni sui blueprint, consulta. [Creazione di un progetto completo con CodeCatalyst progetti](#)

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare il moderno progetto di applicazione web a tre livelli](#)
- [Fase 2: Invita qualcuno al tuo progetto](#)
- [Passaggio 3: Crea problemi su cui collaborare e tenere traccia del lavoro](#)
- [Passaggio 4: Visualizza il tuo repository di origine](#)
- [Passaggio 5: creare un ambiente di sviluppo con un ramo di test e apportare una rapida modifica al codice](#)
- [Passaggio 6: Visualizza il flusso di lavoro che crea l'applicazione moderna](#)
- [Passaggio 7: chiedi agli altri di rivedere le modifiche](#)
- [Passaggio 8: chiudere il problema](#)
- [Pulizia delle risorse](#)
- [Documentazione di riferimento](#)

Prerequisiti

Per creare un progetto applicativo moderno in questo tutorial, devi aver completato le attività [Configura e accedi a CodeCatalyst](#) seguenti:

- Disponi di un AWS Builder ID a cui accedere CodeCatalyst.

- Appartenete a uno spazio e fatevi assegnare il ruolo di amministratore dello spazio o Power user in quello spazio. Per ulteriori informazioni, consulta [Creare uno spazio](#), [Concessione delle autorizzazioni di spazio agli utenti](#) e [Ruolo di amministratore dello spazio](#).
- Account AWS Associa uno al tuo spazio e ottieni il ruolo IAM che hai creato durante la registrazione. Ad esempio, durante la registrazione, hai la possibilità di scegliere di creare un ruolo di servizio con una politica di ruolo chiamata CodeCatalystWorkflowDevelopmentRole-**spaceName** politica del ruolo. Il ruolo avrà un nome CodeCatalystWorkflowDevelopmentRole-**spaceName** con un identificatore univoco aggiunto. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere. [Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-**spaceName**del servizio](#) Per i passaggi per creare il ruolo, vedere [Creazione del CodeCatalystWorkflowDevelopmentRole-**spaceName**ruolo per il tuo account e il tuo spazio](#).

Fase 1: Creare il moderno progetto di applicazione web a tre livelli

Dopo averlo creato, nel progetto potrai sviluppare e testare il codice, coordinare le attività di sviluppo e visualizzare le metriche del progetto. Il progetto contiene anche gli strumenti e le risorse di sviluppo.

In questo tutorial, utilizzerai il moderno modello di applicazione web a tre livelli per creare un'applicazione interattiva. Il flusso di lavoro creato ed eseguito automaticamente come parte del progetto creerà e distribuirà l'applicazione. Il flusso di lavoro viene eseguito correttamente solo dopo che tutti i ruoli e le informazioni sull'account sono stati configurati per il tuo spazio. Dopo che il flusso di lavoro è stato eseguito correttamente, puoi visitare l'URL dell'endpoint per vedere l'applicazione.

Per creare un progetto con un blueprint

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri creare un progetto.
3. Seleziona Crea progetto.
4. Scegli Inizia con un progetto.
5. Nella barra di ricerca inserisci **modern**.
6. Seleziona il blueprint di applicazione Web moderna a tre livelli, quindi scegli Avanti.
7. In Assegna un nome al progetto, inserisci un nome per il progetto. Per esempio:

MyExampleProject.

 Note

Il nome deve essere univoco nel tuo spazio.

8. In Account, scegli Account AWS quello che hai aggiunto durante la registrazione. Il blueprint installerà le risorse in questo account.
9. In Deployment Role, scegli il ruolo che hai aggiunto durante la registrazione. Ad esempio, scegli `CodeCatalystWorkflowDevelopmentRole-spaceName`.

Se non ci sono ruoli elencati, aggiungine uno. Per aggiungere un ruolo, scegli Aggiungi ruolo IAM e aggiungi il ruolo al tuo Account AWS. Per ulteriori informazioni, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS](#).

10. Nella piattaforma Compute, scegli Lambda.
11. Nell'opzione di hosting frontend, scegli Amplify Hosting. Per informazioni su AWS Amplify, vedi [Cos'è l'hosting? AWS Amplify nella Guida AWS Amplify per l'utente](#).
12. In Deployment Region, inserisci il codice regionale Regione AWS in cui desideri che il blueprint distribuisca l'applicazione Mysfits e le risorse di supporto. Per un elenco dei codici regionali, consulta Endpoint [regionali](#) in. Riferimenti generali di AWS
13. Nel Nome applicazione, lascia il valore predefinito `dimysfitsstring`.
14. (Facoltativo) In Genera anteprima del progetto, scegliete Visualizza codice per visualizzare in anteprima i file sorgente che verranno installati dal blueprint. Scegliete Visualizza flusso di lavoro per visualizzare in anteprima i file di definizione del flusso di lavoro CI/CD che verranno installati dal blueprint. L'anteprima si aggiorna dinamicamente in base alle selezioni effettuate.
15. Seleziona Crea progetto.

Il flusso di lavoro del progetto inizia non appena si crea il progetto. Ci vorrà un po' di tempo per completare la creazione e la distribuzione del codice. Nel frattempo, vai avanti e invita qualcun altro al tuo progetto.

Fase 2: Invita qualcuno al tuo progetto

Ora che hai impostato il tuo progetto, invita altre persone a lavorare con te.

Per invitare qualcuno al tuo progetto

1. Vai al progetto a cui desideri invitare gli utenti.

2. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Nella scheda Membri, scegli Invita.
4. Digita gli indirizzi email delle persone che desideri invitare come utenti del progetto. Puoi digitare più indirizzi e-mail separati da uno spazio o una virgola. Puoi anche scegliere tra membri del tuo spazio che non sono membri del progetto.
5. Scegli il ruolo dell'utente.

Quando hai finito di aggiungere utenti, scegli Invita.

Passaggio 3: Crea problemi su cui collaborare e tenere traccia del lavoro

CodeCatalyst ti aiuta a tenere traccia delle funzionalità, delle attività, dei bug e di qualsiasi altra attività coinvolta nel progetto con problemi. Puoi creare problemi per tenere traccia del lavoro e delle idee necessari. Per impostazione predefinita, quando crei un problema, questo viene aggiunto al backlog. Puoi spostare i problemi in una bacheca in cui tenere traccia dei lavori in corso. Puoi anche assegnare un problema a un membro specifico del progetto.

Per creare un problema per un progetto

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Crea problema.
3. Nel titolo del problema, fornisci un nome per il problema. Facoltativamente, fornisci una descrizione del problema. In questo esempio, usa **make a change in the src/mysfit_data.json file**.
4. Scegli la priorità, la stima, lo stato e le etichette. In Assegnatario, scegli +Aggiungi me per assegnare il problema a te stesso.
5. Scegli Crea problema. Il problema è ora visibile sulla lavagna. Scegli la scheda per spostare il problema nella colonna In corso.

Per ulteriori informazioni, consulta [Tieni traccia e organizza il lavoro con problemi in CodeCatalyst](#).

Passaggio 4: Visualizza il tuo repository di origine

Il tuo blueprint installa un repository di sorgenti che contiene i file per definire e supportare l'applicazione o il servizio. Alcune directory e file degni di nota nel repository dei sorgenti sono:

- `Directory.cloud9`: contiene i file di supporto per l'ambiente di sviluppo. AWS Cloud9
- `.codecatalyst`: contiene il file di definizione del flusso di lavoro per ogni YAML flusso di lavoro incluso nel blueprint.
- `.idea`: contiene i file di supporto per gli ambienti di sviluppo. JetBrains
- `.vscode`: contiene i file di supporto per l'ambiente di sviluppo di Visual Studio Code.
- `Directory CDKStacks`: contiene i file AWS CDK stack che definiscono l'infrastruttura in Cloud AWS
- `directory src`: contiene il codice sorgente dell'applicazione.
- `directory tests`: contiene i file per i test interi e unitari eseguiti come parte del flusso di lavoro CI/CD automatizzato che viene eseguito durante la creazione e il test dell'applicazione.
- `directory web`: contiene il codice sorgente del frontend. Altri file includono file di progetto, ad esempio il `package.json` file che contiene importanti metadati sul progetto, la `index.html` pagina del sito Web, il file per il codice linting e il `.eslintrc.cjs` file per specificare i `tsconfig.json` file root e le opzioni del compilatore.
- `Dockerfile`: descrive il contenitore dell'applicazione.
- `README.md` file — Contiene informazioni di configurazione per il progetto.

Per accedere agli archivi di origine di un progetto

1. Accedete al progetto ed effettuate una delle seguenti operazioni:
 - Nella pagina di riepilogo del progetto, scegli il repository desiderato dall'elenco, quindi scegli Visualizza archivio.
 - Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Nei repository di origine, scegli il nome del repository dall'elenco. Puoi filtrare l'elenco dei repository digitando parte del nome del repository nella barra dei filtri.
2. Nella home page del repository, visualizza il contenuto del repository e le informazioni sulle risorse associate, come il numero di richieste pull e i flussi di lavoro. Per impostazione predefinita, vengono visualizzati i contenuti del ramo predefinito. È possibile modificare la visualizzazione scegliendo un ramo diverso dall'elenco a discesa.

Passaggio 5: creare un ambiente di sviluppo con un ramo di test e apportare una rapida modifica al codice

Puoi lavorare rapidamente sul codice nel tuo repository dei sorgenti creando un ambiente di sviluppo. Per questo tutorial, supponiamo che tu voglia:

- Crea un ambiente di AWS Cloud9 sviluppo.
- Scegli l'opzione per lavorare in un nuovo ramo del ramo principale quando crei l'ambiente di sviluppo.
- Usa il nome test per questo nuovo ramo.

In una fase successiva, utilizzerai Dev Environment per apportare una modifica al codice e creare una pull request.

Per creare un ambiente di sviluppo con un nuovo ramo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri creare un ambiente di sviluppo.
3. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, scegli Archivi di origine e scegli il repository per il quale desideri creare un ambiente di sviluppo.
4. Nella home page del repository, scegli Crea ambiente di sviluppo.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Scegli il repository da clonare, scegli Lavora in un nuovo ramo, inserisci il nome di un ramo nel campo Nome del ramo e scegli un ramo dal quale creare il nuovo ramo dal menu a discesa Crea ramo da.
7. Facoltativamente, aggiungi un alias per Dev Environment.
8. Facoltativamente, scegli il pulsante di modifica della configurazione dell'ambiente di sviluppo per modificare la configurazione di calcolo, archiviazione o timeout dell'ambiente di sviluppo.
9. Scegli Create (Crea) . Durante la creazione dell'ambiente di sviluppo, la colonna di stato dell'ambiente di sviluppo mostrerà Avvio e la colonna di stato verrà visualizzata in esecuzione una volta creato l'ambiente di sviluppo. Si aprirà una nuova scheda con il tuo ambiente di sviluppo nell'IDE di tua scelta. Puoi modificare il codice, eseguire il commit e inviare le modifiche.

In questa sezione, utilizzerai l'applicazione di esempio generata apportando modifiche al codice con richieste pull che vengono automaticamente create e distribuite alle risorse della rete connessa Account AWS quando la pull request viene unita. CodeCatalyst

Per apportare una modifica al file **src/mysfit_data.json**

1. Vai all'ambiente di sviluppo del tuo progetto. In AWS Cloud9, espandi il menu di navigazione laterale per sfogliare i file. `mysfitssrc` Espandi e apri `src/mysfit_data.json`.
2. Nel file, modificate il valore del "Age": campo da 6 a 12. La riga dovrebbe essere simile alla seguente:

```
{
  "Age": 12,
  "Description": "Twilight's personality sparkles like the night sky and is looking for a forever home with a Greek hero or God. While on the smaller side at 14 hands, he is quite adept at accepting riders and can fly to 15,000 feet. Twilight needs a large area to run around in and will need to be registered with the FAA if you plan to fly him above 500 feet. His favorite activities include playing with chimeras, going on epic adventures into battle, and playing with a large inflatable ball around the paddock. If you bring him home, he'll quickly become your favorite little Pegasus.",
  "GoodEvil": "Good",
  "LawChaos": "Lawful",
  "Name": "Twilight Glitter",
  "ProfileImageUri": "https://www.mythicalmysfits.com/images/pegasus_hover.png",
  "Species": "Pegasus",
  "ThumbImageUri": "https://www.mythicalmysfits.com/images/pegasus_thumb.png"
},
```

3. Salvare il file.
4. Passa al repository `mysfits` con il comando. **`cd /projects/mysfits`**
5. Aggiungi, conferma e invia le tue modifiche con i comandi `git add`, `git commit` e `git push`.

```
git add .
git commit -m "make an example change"
git push
```

Passaggio 6: Visualizza il flusso di lavoro che crea l'applicazione moderna

Dopo aver creato il progetto applicativo moderno, CodeCatalyst genera diverse risorse per tuo conto, incluso un flusso di lavoro. Un workflow è una procedura automatizzata definita in un file.yaml che descrive come creare, testare e distribuire il codice.

In questo tutorial, hai CodeCatalyst creato un flusso di lavoro e lo hai avviato automaticamente al momento della creazione del progetto. (Il flusso di lavoro potrebbe essere ancora in esecuzione a seconda di quanto tempo fa hai creato il progetto.) Utilizzate le seguenti procedure per verificare l'avanzamento del flusso di lavoro, esaminare i log generati e i rapporti di test e, infine, accedere all'URL dell'applicazione distribuita.

Per verificare l'avanzamento del flusso di lavoro

1. Nella CodeCatalyst console, nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

Viene visualizzato un elenco di flussi di lavoro. Questi sono i flussi di lavoro generati e avviati dal CodeCatalyst blueprint al momento della creazione del progetto.

2. Osserva l'elenco dei flussi di lavoro. Dovresti vederne quattro:
 - I due flussi di lavoro nella parte superiore corrispondono al test ramo creato in precedenza in [Passaggio 5: creare un ambiente di sviluppo con un ramo di test e apportare una rapida modifica al codice](#). Questi flussi di lavoro sono cloni dei flussi di lavoro sul ramo. main non ApplicationDeploymentPipeline è attivo perché è configurato per l'uso con il ramo. main Il OnPullRequest flusso di lavoro non è stato eseguito perché non è stata effettuata alcuna pull request.
 - I due flussi di lavoro nella parte inferiore corrispondono al main ramo creato quando è stato eseguito il blueprint in precedenza. Il ApplicationDeploymentPipeline flusso di lavoro è attivo e ha un'esecuzione in corso (o completata).

Note

Se l'ApplicationDeploymentPipeline esecuzione fallisce con un Build @cdk_bootstrap o un DeployBackend errore, potrebbe essere perché in precedenza è stata eseguita l'applicazione Web Modern a tre livelli, che ha lasciato vecchie risorse in conflitto con il blueprint corrente. Dovrai eliminare queste vecchie risorse e quindi eseguire nuovamente il flusso di lavoro. Per ulteriori informazioni, consulta [Pulizia delle risorse](#).

3. Scegli il `ApplicationDeploymentPipeline` flusso di lavoro associato al `main` ramo, in basso. Questo flusso di lavoro è stato eseguito utilizzando il codice sorgente del `main` ramo.

Viene visualizzato un diagramma del flusso di lavoro. Il diagramma mostra diversi blocchi, ognuno dei quali rappresenta un'attività o un'azione. La maggior parte delle azioni è disposta verticalmente, con le azioni in alto che precedono quelle sottostanti. Le azioni disposte fianco a fianco vengono eseguite in parallelo. Le azioni raggruppate devono essere eseguite tutte correttamente prima che l'azione sottostante possa iniziare.

I blocchi principali sono:

- `WorkflowSource`— Questo blocco rappresenta il tuo repository di origine. Mostra, tra le altre informazioni, il nome del repository di origine (`mysfits`) e il commit che ha avviato automaticamente l'esecuzione del flusso di lavoro. CodeCatalyst ha generato questo commit quando hai creato il tuo progetto.
 - `Compila`: questo blocco rappresenta un raggruppamento di due azioni che devono entrambe essere completate correttamente per iniziare l'azione successiva.
 - `DeployBackend`— Questo blocco rappresenta un'azione che distribuisce i componenti di backend dell'applicazione nel cloud. AWS
 - `Test`: questo blocco rappresenta un raggruppamento di due azioni di test che devono entrambe essere completate correttamente per iniziare l'azione successiva.
 - `DeployFrontend`— Questo blocco rappresenta un'azione che distribuisce i componenti frontend dell'applicazione nel cloud. AWS
4. Scegli la scheda `Definizione` (in alto). Il [file di definizione del flusso](#) di lavoro viene visualizzato sulla destra. Il file contiene le seguenti sezioni degne di nota:
 - Una `Triggers` sezione, in alto. Questa sezione indica che il flusso di lavoro deve iniziare ogni volta che il codice viene inviato al ramo del repository di `main` origine. I push ad altri rami (ad esempio `test`) non avvieranno questo flusso di lavoro. Il flusso di lavoro viene eseguito utilizzando i file presenti nel `main` ramo.
 - Una `Actions` sezione, sotto `Triggers`. Questa sezione definisce le azioni visualizzate nel diagramma del flusso di lavoro.
 5. Scegli la scheda `Stato` più recente (in alto) e scegli qualsiasi azione nel diagramma del flusso di lavoro.

6. A destra, scegli la scheda Configurazione per visualizzare le impostazioni di configurazione utilizzate dall'azione durante l'ultima esecuzione. Ogni impostazione di configurazione ha una proprietà corrispondente nel file di definizione del flusso di lavoro.
7. Lasciare aperta la console e passare alla procedura successiva.

Per esaminare i registri di compilazione e i rapporti sui test

1. Scegli la scheda Stato più recente.
2. Nel diagramma del flusso di lavoro, scegli l'DeployFrontendazione.
3. Attendi che l'azione finisca. Fai attenzione che l'icona «in corso» () si trasformi in un'icona di «successo» (.
4. Scegli l'azione build_backend.
5. Scegli la scheda Registri ed espandi un paio di sezioni per visualizzare i messaggi di registro relativi a questi passaggi. Puoi vedere i messaggi relativi alla configurazione del backend.
6. Scegli la scheda Rapporti, quindi scegli il backend-coverage.xml rapporto. CodeCatalyst visualizza il rapporto associato. Il rapporto mostra i test di copertura del codice che sono stati eseguiti e indica la percentuale di righe di codice che sono state convalidate con successo dai test, ad esempio l'80%.

Per ulteriori informazioni sui report dei test, vedere [Test con flussi di lavoro](#).

 Tip

È inoltre possibile visualizzare i report dei test selezionando Rapporti nel riquadro di navigazione.

7. Lascia la CodeCatalyst console aperta e vai alla procedura successiva.

Per confermare che l'applicazione moderna è stata implementata correttamente

1. Tornate al ApplicationDeploymentPipelineflusso di lavoro e scegliete il *string* link Run-dell'ultima esecuzione.

2. Nel diagramma del flusso di lavoro, individua l'DeployFrontendazione e scegli il link Visualizza app. Viene visualizzato il sito Web Mysfit.

Note

Se non vedi il link Visualizza app all'interno dell'DeployFrontendazione, assicurati di aver scelto il link Run ID.

3. Cerca il pegasus Mysfit chiamato Twilight Glitter. Nota il valore per l'età. Lo è 6. Effettuerai una modifica al codice per aggiornare l'età.

Passaggio 7: chiedi agli altri di rivedere le modifiche

Ora che hai apportato delle modifiche a un ramo denominato `test`, puoi chiedere ad altri di esaminarle creando una pull request. Esegui i seguenti passaggi per creare una richiesta pull per unire le modifiche dal `test` ramo al `main` ramo.

Per creare una pull request

1. Vai al tuo progetto.
2. Esegui una di queste operazioni:
 - Nel pannello di navigazione, scegli Codice, scegli Richieste pull, quindi scegli Crea richiesta pull.
 - Nella home page del repository, scegli Altro, quindi scegli Crea richiesta pull.
 - Nella pagina del progetto, scegli Crea richiesta pull.
3. Nel repository Source, assicurati che il repository di origine specificato sia quello che contiene il codice commesso. Questa opzione appare solo se non hai creato la pull request dalla pagina principale del repository.
4. Nel ramo di destinazione, scegli il ramo in cui unire il codice dopo averlo esaminato.
5. Nel ramo di origine, scegli il ramo che contiene il codice commesso.
6. In Pull request title, inserisci un titolo che aiuti gli altri utenti a capire cosa deve essere esaminato e perché.
7. (Facoltativo) Nella descrizione della richiesta Pull, fornisci informazioni come un link ai problemi o una descrizione delle modifiche.

 Tip

Puoi scegliere Write description for me per generare CodeCatalyst automaticamente una descrizione delle modifiche contenute nella pull request. Puoi apportare modifiche alla descrizione generata automaticamente dopo averla aggiunta alla pull request. Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio e non è disponibile per le richieste pull negli archivi collegati. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

8. (Facoltativo) In Problemi, scegli Problemi di collegamento, quindi scegli un problema dall'elenco o inserisci il relativo ID. Per scollegare un problema, scegli l'icona di scollegamento.
9. (Facoltativo) In Revisori obbligatori, scegli Aggiungi revisori richiesti. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori obbligatori devono approvare le modifiche prima che la pull request possa essere unita al ramo di destinazione.

 Note

Non è possibile aggiungere un revisore sia come revisore obbligatorio che come revisore opzionale. Non puoi aggiungerti come revisore.

10. (Facoltativo) In Revisori opzionali, scegli Aggiungi revisori opzionali. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori opzionali non devono approvare le modifiche come requisito prima che la pull request possa essere unita al ramo di destinazione.
11. Esamina le differenze tra le filiali. La differenza mostrata in una pull request è rappresentata dalle modifiche tra la revisione nel ramo di origine e la base di unione, che è l'head commit del ramo di destinazione al momento della creazione della pull request. Se non viene visualizzata alcuna modifica, i rami potrebbero essere identici oppure potresti aver scelto lo stesso ramo sia per l'origine che per la destinazione.
12. Quando sei sicuro che la pull request contenga il codice e le modifiche che desideri rivedere, scegli Crea.

 Note

Dopo aver creato la pull request, puoi aggiungere commenti. I commenti possono essere aggiunti alla pull request o alle singole righe dei file, nonché alla pull request

complessiva. È possibile aggiungere collegamenti a risorse, ad esempio file, utilizzando il segno @ seguito dal nome del file.

Quando create la pull request, il OnPullRequestflusso di lavoro inizia a utilizzare i file sorgente presenti nel test ramo. Mentre i revisori approvano la modifica del codice, puoi osservare i risultati scegliendo il flusso di lavoro e visualizzando l'output del test.

Dopo aver esaminato la modifica, puoi unire il codice. L'unione del codice nel ramo predefinito avvierà automaticamente il flusso di lavoro che genererà e distribuirà le modifiche.

Per unire una pull request dalla console CodeCatalyst

1. Passa al tuo progetto di applicazione moderna.
2. Nella pagina del progetto, in Open pull request, scegli la pull request che desideri unire. Se non vedi la pull request, scegli Visualizza tutto e poi selezionala dall'elenco. Selezionare Merge (Unisci).
3. Scegli tra le strategie di unione disponibili per la pull request. Facoltativamente, seleziona o deselecta l'opzione per eliminare il ramo di origine dopo aver unito la pull request, quindi scegli Merge.

Note

Se il pulsante Unisci non è attivo o viene visualizzata l'etichetta Non unibile, significa che uno o più revisori obbligatori non hanno ancora approvato la pull request oppure la pull request non può essere unita nella console. CodeCatalyst Un revisore che non ha approvato una pull request è indicato da un'icona a forma di orologio in Panoramica nell'area dei dettagli della pull request. Se tutti i revisori obbligatori hanno approvato la pull request ma il pulsante Merge non è ancora attivo, è possibile che si verifichi un conflitto di unione. È possibile risolvere i conflitti di unione per il ramo di destinazione nella CodeCatalyst console e quindi unire la richiesta pull, oppure è possibile risolvere i conflitti e unirli localmente, quindi inviare il commit che contiene l'unione a CodeCatalyst. Per ulteriori informazioni, consulta [Unire una richiesta pull \(Git\)](#) la documentazione di Git.

Dopo aver unito le modifiche dalla test filiale alla **main** filiale, la modifica avvia automaticamente il ApplicationDeploymentPipelineflusso di lavoro che crea e distribuisce la modifica.

Per vedere il commit unito eseguito attraverso il flusso di lavoro ApplicationDeploymentPipeline

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. In Flussi di lavoro, in, espandi Esecuzioni ApplicationDeploymentPipelinerrecenti. Puoi vedere il flusso di lavoro eseguito avviato dal merge commit. Facoltativamente, scegliilo per vedere l'avanzamento dell'esecuzione.
3. Al termine dell'esecuzione, ricarica l'URL che hai visitato in precedenza. Visualizza il pegaso per verificare che l'età sia cambiata.



Passaggio 8: chiudere il problema

Quando un problema viene risolto, può essere chiuso sulla CodeCatalyst console.

Per chiudere un problema relativo a un progetto

1. Vai al tuo progetto.
2. Nel riquadro di navigazione, scegli Problemi.
3. Drag-and-drop il problema nella colonna Fine.

Per ulteriori informazioni, consulta [Tieni traccia e organizza il lavoro con problemi in CodeCatalyst](#).

Pulizia delle risorse

Pulisci CodeCatalyst e AWS rimuovi le tracce di questo tutorial dal tuo ambiente.

Puoi scegliere di continuare a utilizzare il progetto che hai usato per questo tutorial oppure puoi eliminare il progetto e le risorse associate.

Note

L'eliminazione di questo progetto eliminerà tutti gli archivi, i problemi e gli artefatti del progetto per tutti i membri.

Per eliminare un progetto

1. Vai al tuo progetto, quindi scegli Impostazioni progetto.
2. Scegli la scheda Generale.
3. Sotto il nome del progetto, scegli Elimina progetto.

Per eliminare risorse in CloudFormation Amazon S3

1. Accedi Console di gestione AWS con lo stesso account che hai aggiunto al tuo CodeCatalyst spazio.
2. Vai al CloudFormationservizio.
3. Elimina lo stack mysfits **string**.
4. Elimina lo stack development-mysfits. **string**
5. Scegliete (ma non eliminate) lo stack. CDKToolkit Scegliere la scheda Resources (Risorse). Scegli il StagingBucketlink ed elimina il bucket e il contenuto del bucket in Amazon S3.

Note

Se non elimini questo bucket manualmente, potresti visualizzare un errore quando riesegui il blueprint dell'applicazione web Modern a tre livelli.

6. (Facoltativo) Eliminare lo stack. CDKToolkit

Documentazione di riferimento

Il moderno modello di applicazione web a tre livelli distribuisce le risorse nel tuo CodeCatalyst spazio e nel tuo AWS account nel cloud. AWS Queste risorse sono:

- Nel tuo CodeCatalyst spazio:
 - Un CodeCatalyst progetto che include le seguenti risorse:
 - Un [archivio sorgente: questo repository](#) contiene codice di esempio per un'applicazione web «Mysfits».
 - Un [flusso di lavoro](#): questo flusso di lavoro crea e distribuisce il codice dell'applicazione Mysfits ogni volta che viene apportata una modifica al ramo predefinito
 - Una [bacheca dei problemi](#) e un backlog: questa bacheca e il backlog possono essere utilizzati per pianificare e tenere traccia del lavoro.
 - Una [suite di report di test](#): questa suite include report automatici inclusi nel codice di esempio.
- Nella sezione associata Account AWS:
 - Uno CDKToolkitstack: questo stack distribuisce le seguenti risorse:
 - Un bucket di staging Amazon S3, una policy sui bucket e la AWS KMS chiave utilizzata per crittografare il bucket.
 - Un ruolo di implementazione IAM per l'azione di distribuzione.
 - AWS Ruoli e policy IAM a supporto delle risorse dello stack.

Note

Non CDKToolkitviene demolito e ricreato per ogni implementazione. Si tratta di uno stack che viene avviato in ogni account per supportare. AWS CDK

- Uno stack development-mysfits: questo *string* BackEnd stack distribuisce le seguenti risorse di backend:
 - Un endpoint Amazon API Gateway.
 - AWS Ruoli e policy IAM a supporto delle risorse dello stack.
 - Una AWS Lambda funzione e un livello forniscono la piattaforma di elaborazione serverless per l'applicazione moderna.
 - Una politica e un ruolo IAM per l'implementazione del bucket e la funzione Lambda.
- Uno stack mysfits: questo *string* stack distribuisce l'applicazione frontend. AWS Amplify

Consulta anche

Per ulteriori informazioni sui AWS servizi in cui vengono create le risorse nell'ambito di questo tutorial, consulta quanto segue:

- Amazon S3: un servizio per l'archiviazione delle risorse frontend su un servizio di storage di oggetti che offre scalabilità, alta disponibilità dei dati, sicurezza e prestazioni leader del settore. Per ulteriori informazioni, consulta la Guida per l'[utente di Amazon S3](#).
- Amazon API Gateway: un servizio per la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione di REST, HTTP e WebSocket APIs su qualsiasi scala. Per ulteriori informazioni, consulta [API Gateway Developer Guide](#).
- Amplify: un servizio per l'hosting dell'applicazione frontend. Per ulteriori informazioni, consulta [AWS Amplify Hosting User Guide](#).
- AWS Cloud Development Kit (AWS CDK)— Un framework per definire l'infrastruttura cloud in codice e fornirla tramite AWS CloudFormation. AWS CDK Include il AWS CDK Toolkit, uno strumento a riga di comando per interagire con AWS CDK app e stack. Per ulteriori informazioni, consulta la [Guida per sviluppatori di AWS Cloud Development Kit \(AWS CDK\)](#).
- Amazon DynamoDB: un servizio di database NoSQL completamente gestito per l'archiviazione dei dati. Per ulteriori informazioni, consulta [Amazon DynamoDB Developer Guide](#).
- AWS Lambda— Un servizio per richiamare il codice su un'infrastruttura di calcolo ad alta disponibilità senza fornire o gestire server. Per ulteriori informazioni, consulta la [Guida per sviluppatori di AWS Lambda](#).
- AWS IAM: un servizio per il controllo sicuro dell'accesso e delle relative AWS risorse. Per ulteriori informazioni, consulta [Guida per l'utente di IAM](#).

Tutorial: iniziare con un progetto vuoto e aggiungere manualmente le risorse

Puoi creare un progetto vuoto senza risorse predefinite al suo interno scegliendo il blueprint del progetto Empty quando crei il progetto. Dopo aver creato un progetto vuoto, è possibile creare e aggiungere risorse ad esso in base alle esigenze del progetto. Poiché i progetti creati senza un blueprint sono vuoti al momento della creazione, questa opzione richiede una maggiore conoscenza della creazione e della configurazione CodeCatalyst delle risorse per iniziare.

Argomenti

- [Prerequisiti](#)
- [Crea un progetto vuoto](#)
- [Crea un repository di sorgenti](#)
- [Crea un flusso di lavoro per creare, testare e implementare una modifica al codice](#)
- [Invita qualcuno al tuo progetto](#)
- [Crea problemi su cui collaborare e monitora il lavoro](#)

Prerequisiti

Per creare un progetto vuoto, è necessario che vi sia assegnato il ruolo di amministratore dello spazio o Power user. Se è la prima volta che accedi a CodeCatalyst, vedi [Configura e accedi a CodeCatalyst](#).

Crea un progetto vuoto

La creazione di un progetto è il primo passo per poter lavorare insieme. Se desideri creare risorse personalizzate, come archivi di sorgenti e flussi di lavoro, puoi iniziare con un progetto vuoto.

Per creare un progetto vuoto

1. Vai allo spazio in cui desideri creare un progetto.
2. Nella dashboard dello spazio, scegli Crea progetto.
3. Scegli Inizia da zero.
4. In Dai un nome al tuo progetto, inserisci il nome che desideri assegnare al progetto. Il nome deve essere univoco all'interno del tuo spazio.
5. Seleziona Crea progetto.

Ora che hai un progetto vuoto, il passo successivo è creare un repository di sorgenti.

Crea un repository di sorgenti

Crea un archivio di sorgenti per archiviare e collaborare al codice del tuo progetto. I membri del progetto possono clonare questo repository sui propri computer locali per lavorare sul codice. In alternativa, puoi scegliere di collegare un repository ospitato in un servizio supportato, ma questo non è trattato in questo tutorial. Per ulteriori informazioni, consulta [Collegamento di un repository di sorgenti](#).

Per creare un archivio di sorgenti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Crea repository.
5. In Nome archivio, fornisci un nome per il repository. In questa guida, lo usiamo *codecatalyst-source-repository*, ma puoi scegliere un nome diverso. I nomi dei repository devono essere univoci all'interno di un progetto. Per ulteriori informazioni sui requisiti per i nomi dei repository, vedere. [Quote per i repository di sorgenti in CodeCatalyst](#)
6. (Facoltativo) In Descrizione, aggiungi una descrizione del repository che aiuterà gli altri utenti del progetto a capire a cosa serve il repository.
7. Scegli Crea repository (impostazione predefinita). Questa opzione crea un repository che include un ramo predefinito e un file README.md. A differenza di un repository vuoto, è possibile utilizzare questo repository non appena viene creato.
8. Nel ramo predefinito, lascia il nome come principale a meno che tu non abbia un motivo per scegliere un nome diverso. Gli esempi di questa guida utilizzano tutti il nome main per il ramo predefinito.
9. (Facoltativo) Aggiungi un .gitignore file per il tipo di codice che intendi inviare.
10. Scegli Create (Crea) .

Note

CodeCatalyst aggiunge un README .md file al tuo repository quando lo crei. CodeCatalyst crea anche un commit iniziale per il repository in un ramo predefinito denominato main. È possibile modificare o eliminare il file README.md, ma non è possibile eliminare il ramo predefinito.

Puoi aggiungere rapidamente codice nel tuo repository creando un ambiente di sviluppo. Per questo tutorial, ti consigliamo di creare un ambiente di sviluppo utilizzando AWS Cloud9 e scegliere l'opzione per creare un ramo dal ramo principale durante la creazione dell'ambiente di sviluppo. Utilizziamo il nome **test** per questo ramo, ma puoi inserire un nome di ramo diverso se preferisci.

Per creare un ambiente di sviluppo con un nuovo ramo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri creare un ambiente di sviluppo.
3. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, scegli Archivi di origine e scegli il repository per il quale desideri creare un ambiente di sviluppo.
4. Nella home page del repository, scegli Crea ambiente di sviluppo.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Scegli il repository da clonare, scegli Lavora in un nuovo ramo, inserisci il nome di un ramo nel campo Nome del ramo e scegli un ramo dal quale creare il nuovo ramo dal menu a discesa Crea ramo da.
7. Facoltativamente, aggiungi un alias per Dev Environment.
8. Facoltativamente, scegli il pulsante di modifica della configurazione dell'ambiente di sviluppo per modificare la configurazione di calcolo, archiviazione o timeout dell'ambiente di sviluppo.
9. Scegli Create (Crea) . Durante la creazione dell'ambiente di sviluppo, la colonna di stato dell'ambiente di sviluppo mostrerà Avvio e la colonna di stato verrà visualizzata in esecuzione una volta creato l'ambiente di sviluppo. Si aprirà una nuova scheda con il tuo ambiente di sviluppo nell'IDE di tua scelta. Puoi modificare il codice, eseguire il commit e inviare le modifiche.

Crea un flusso di lavoro per creare, testare e implementare una modifica al codice

Nel CodeCatalyst, organizzi la creazione, il test e la distribuzione delle tue applicazioni o servizi nei flussi di lavoro. I flussi di lavoro consistono in azioni e possono essere configurati per essere eseguiti automaticamente dopo che si verificano eventi specifici del repository di origine, come l'invio di codice o l'apertura o l'aggiornamento di una pull request. Per ulteriori informazioni sui flussi di lavoro, consulta [Crea, testa e distribuisci con i flussi di lavoro](#).

Segui le istruzioni per creare il tuo [Guida introduttiva ai flussi di lavoro](#) primo flusso di lavoro.

Invita qualcuno al tuo progetto

Ora che hai impostato il tuo progetto personalizzato, invita altre persone a lavorare con te.

Per invitare qualcuno al tuo progetto

1. Vai al progetto a cui desideri invitare gli utenti.
2. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Nella scheda Membri, scegli Invita.
4. Digita gli indirizzi email delle persone che desideri invitare come utenti del progetto. Puoi digitare più indirizzi e-mail separati da uno spazio o da una virgola. Puoi anche scegliere tra membri del tuo spazio che non sono membri del progetto.
5. Scegli il ruolo dell'utente.

Quando hai finito di aggiungere utenti, scegli Invita.

Crea problemi su cui collaborare e monitora il lavoro

CodeCatalyst ti aiuta a tenere traccia delle funzionalità, delle attività, dei bug e di qualsiasi altra attività coinvolta nel progetto con problemi. Puoi creare problemi per tenere traccia del lavoro e delle idee necessari. Per impostazione predefinita, quando crei un problema, questo viene aggiunto al backlog. Puoi spostare i problemi in una bacheca in cui tenere traccia dei lavori in corso. Puoi anche assegnare un problema a un membro specifico del progetto.

Per creare un problema per un progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).

Assicurati di navigare all'interno del progetto in cui desideri creare problemi. Per visualizzare tutti i progetti, nel pannello di navigazione, scegli Amazon e CodeCatalyst, se necessario, scegli Visualizza tutti i progetti. Scegli il progetto in cui vuoi creare o risolvere i problemi.

2. Nel riquadro di navigazione, scegli Track, quindi scegli Backlog.
3. Scegli Crea problema.
4. Nel titolo del problema, fornisci un nome per il problema. Facoltativamente, fornisci una descrizione del problema. Se lo desideri, scegli lo stato, la priorità e la stima del problema. Puoi anche assegnare il problema a un membro del progetto dall'elenco dei membri del progetto.

Tip

Puoi scegliere di assegnare un problema ad Amazon Q per fare in modo che Amazon Q provi a risolverlo. Se il tentativo ha esito positivo, verrà creata una pull request e lo stato

del problema cambierà in In revisione, in modo da poter esaminare e testare il codice.

Per ulteriori informazioni, consulta [Tutorial: Utilizzo delle funzionalità di intelligenza artificiale CodeCatalyst generativa per velocizzare il lavoro di sviluppo](#).

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

5. Seleziona Salva.

Dopo aver creato i problemi, puoi assegnarli ai membri del progetto, stimarli e tenerne traccia su una bacheca Kanban. Per ulteriori informazioni, consulta [Tieni traccia e organizza il lavoro con problemi in CodeCatalyst](#).

Tutorial: Utilizzo delle funzionalità di intelligenza artificiale CodeCatalyst generativa per velocizzare il lavoro di sviluppo

Se disponi di un progetto e di un archivio di sorgenti in Amazon CodeCatalyst in uno spazio in cui sono abilitate le funzionalità di intelligenza artificiale generativa, puoi utilizzare queste funzionalità per accelerare lo sviluppo del software. Gli sviluppatori hanno spesso più attività da svolgere che tempo per eseguirle. Spesso non si prendono il tempo di spiegare le modifiche apportate al codice ai colleghi quando creano richieste pull per la revisione di tali modifiche, aspettandosi che gli altri utenti trovino le modifiche da sole. Inoltre, i creatori e i revisori delle pull request non hanno il tempo di trovare e leggere attentamente tutti i commenti su una pull request, in particolare se la pull request ha più revisioni. CodeCatalyst si integra con Amazon Q Developer Agent per lo sviluppo di software per fornire funzionalità di intelligenza artificiale generativa che possono aiutare i membri del team a svolgere le proprie attività più rapidamente e aumentare il tempo a disposizione per concentrarsi sulle parti più importanti del loro lavoro.

Amazon Q Developer è un assistente conversazionale generativo basato sull'intelligenza artificiale che può aiutarti a comprendere, creare, estendere e utilizzare le applicazioni. AWS Per accelerare la tua crescita AWS, il modello alla base di Amazon Q è arricchito con AWS contenuti di alta qualità per produrre risposte più complete, utilizzabili e referenziate. Per ulteriori informazioni, consulta [Cos'è Amazon Q Developer?](#) nella Guida per l'utente di Amazon Q Developer.

Note

Realizzato da Amazon Bedrock: AWS implementa il rilevamento [automatico degli abusi](#). Poiché le funzioni *Scrivi descrizione per me*, *Crea un riepilogo dei contenuti*, *Consiglia attività*, *Usa Amazon Q per creare o aggiungere funzionalità a un progetto* e *Assegna problemi ad Amazon Q* con *Amazon Q Developer Agent* per le funzionalità di sviluppo software sono basate su Amazon Bedrock, gli utenti possono sfruttare appieno i controlli implementati in Amazon Bedrock per rafforzare la sicurezza e l'uso responsabile dell'intelligenza artificiale (AI).

In questo tutorial, imparerai come utilizzare le funzionalità di intelligenza artificiale generativa CodeCatalyst per aiutarti a creare progetti con blueprint, oltre ad aggiungere blueprint a progetti esistenti. Inoltre, imparerai come riepilogare le modifiche tra i rami durante la creazione di richieste pull e riepilogare i commenti lasciati su una pull request. Imparerai anche a creare problemi con le tue idee per modifiche o miglioramenti al codice e ad assegnarle ad Amazon Q. Nell'ambito della gestione dei problemi assegnati ad Amazon Q, imparerai come consentire ad Amazon Q di suggerire attività e come assegnare e lavorare su qualsiasi attività creata nell'ambito della risoluzione di un problema.

Argomenti

- [Prerequisiti](#)
- [Utilizzo di Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di funzionalità](#)
- [Crea un riepilogo delle modifiche al codice tra i rami durante la creazione di una pull request](#)
- [Crea un riepilogo dei commenti lasciati sulle modifiche al codice in una pull request](#)
- [Crea un problema e assegnalo ad Amazon Q](#)
- [Crea un problema e chiedi ad Amazon Q di eseguire delle attività consigliate per risolverlo](#)
- [Eseguire la pulizia delle risorse](#)

Prerequisiti

Per utilizzare le CodeCatalyst funzionalità di questo tutorial, devi prima aver completato e avere accesso alle seguenti risorse:

- Hai un AWS Builder ID o un'identità Single Sign-On (SSO) a cui accedere. CodeCatalyst

- Ti trovi in uno spazio in cui sono abilitate le funzionalità di intelligenza artificiale generativa. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).
- Hai il ruolo di collaboratore o amministratore di progetto in un progetto in quello spazio.
- A meno che tu non stia creando un progetto con intelligenza artificiale generativa, per il tuo progetto esistente è configurato almeno un repository di sorgenti. Gli archivi collegati non sono supportati.
- Quando si assegnano problemi a una soluzione iniziale creata dall'intelligenza artificiale generativa, il progetto non può essere configurato con l'estensione Jira Software. L'estensione non è supportata per questa funzionalità.

Per ulteriori informazioni, consulta [Creare uno spazio](#), [Tieni traccia e organizza il lavoro con problemi in CodeCatalyst](#), [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#) e [Concessione dell'accesso con ruoli utente](#).

Questo tutorial si basa su un progetto creato utilizzando il modello di applicazione web Modern a tre livelli con Python. Se utilizzi un progetto creato con un progetto diverso, puoi comunque seguire i passaggi, ma alcune specifiche possono variare, come il codice e il linguaggio di esempio.

Utilizzo di Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di funzionalità

In qualità di sviluppatore di progetti, puoi collaborare con Amazon Q, un assistente AI generativo, durante la creazione di nuovi progetti o l'aggiunta di componenti a progetti esistenti. Puoi fornire ad Amazon Q i requisiti per il tuo progetto interagendo con esso in un'interfaccia simile a una chat. In base ai tuoi requisiti, Amazon Q suggerisce un modello e delinea anche i requisiti che non possono essere soddisfatti. Se il tuo spazio dispone di progetti personalizzati, Amazon Q apprende e include tali progetti anche nei consigli. Puoi quindi procedere con il suggerimento di Amazon Q se sei soddisfatto e creerà le risorse necessarie, come un repository di sorgenti con codice per le tue esigenze. Amazon Q crea anche problemi per requisiti che non possono essere soddisfatti con un blueprint. Per ulteriori informazioni sui CodeCatalyst blueprint disponibili, consulta [Creazione di un progetto completo con CodeCatalyst progetti](#). Per ulteriori informazioni sull'utilizzo di Amazon Q con i blueprint, consulta [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#).

Per creare un progetto con Amazon Q

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Nella CodeCatalyst console, accedi allo spazio in cui desideri creare un blueprint.
3. Nella dashboard dello spazio, scegli Crea con Amazon Q.
4. Nel campo di immissione del testo del prompt di Amazon Q, fornisci istruzioni scrivendo una breve descrizione del progetto che desideri creare. Ad esempio, "I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic and functionality of the application, and a data layer that manages the storage and retrieval of the data."

(Facoltativo) In Prova esempi, puoi utilizzare un prompt già scritto scegliendo un blueprint. Ad esempio, se scegli l'app React, viene fornito il seguente prompt: "I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic and functionality of the application, and a data layer that manages the storage and retrieval of the data. I also want to add authentication and authorization mechanisms for security and allowable actions."

5. Scegli Invia per inviare le istruzioni ad Amazon Q. L'assistente generativo AI fornisce un suggerimento e delinea i requisiti che non possono essere soddisfatti dal modello. Ad esempio, Amazon Q potrebbe suggerire quanto segue in base ai tuoi criteri:

I recommend using the Modern three-tier web application blueprint based on your requirements. Blueprints are dynamic and can always be updated and edited later.

Modern three-tier web application

By Amazon Web Services

This blueprint creates a Mythical Mysfits 3-tier web application with a modular presentation, application, and data layers.

The application leverages containers, infrastructure as code (IaC), continuous integration and continuous delivery (CI/CD), and serverless code functions.

Version: 0.1.163

[View details](#)

The following requirements could not be met so I will create issues for you.

- Add authentication and authorization mechanisms for security and allowable actions.

6. (Facoltativo) Per visualizzare i dettagli approfonditi del blueprint suggerito, scegli Visualizza dettagli.
7. Esegui una delle seguenti operazioni:
 - a. Scegli Sì, usa questo blueprint se sei soddisfatto del suggerimento.
 - b. Scegli Modifica richiesta se desideri modificare la richiesta.
 - c. Scegli Ricomincia se desideri cancellare completamente il prompt.
8. Esegui una delle seguenti operazioni:
 - a. Scegli Configura se desideri configurare il blueprint suggerito. Puoi anche configurare il blueprint in un secondo momento.
 - b. Scegli Salta se al momento non desideri modificare le configurazioni del blueprint.
9. Se hai scelto di configurare il blueprint, scegli Continua dopo aver modificato le risorse del progetto.
10. Quando richiesto, inserisci il nome che desideri assegnare al progetto e i nomi delle risorse associate. Il nome deve essere univoco all'interno del tuo spazio.
11. Scegli Crea progetto per creare un progetto con il blueprint. Amazon Q crea risorse utilizzando il blueprint. Ad esempio, se crei un progetto con il blueprint di un'applicazione a pagina singola, viene creato un repository di origine per il codice e i flussi di lavoro pertinenti per. CI/CD
12. (Facoltativo) Per impostazione predefinita, Amazon Q crea anche problemi per i requisiti che non sono soddisfatti da un blueprint. Puoi scegliere per quali articoli non vuoi creare problemi. Dopo aver scelto di lasciare che Amazon Q crei problemi, puoi assegnare un problema anche ad Amazon Q. Analizzerà il problema nel contesto degli archivi di origine indicati, fornendo un riepilogo dei file e del codice sorgente pertinenti. Per ulteriori informazioni, consultare [Individuazione e visualizzazione dei problemi](#), [Crea un problema e assegnalo ad Amazon Q](#) e [Procedure consigliate per la creazione e l'utilizzo di problemi assegnati ad Amazon Q](#).

Dopo aver creato un progetto con Amazon Q, puoi anche utilizzare Amazon Q per aggiungere nuovi componenti in quanto suggerisce CodeCatalyst progetti in base alle tue esigenze.

Per aggiungere un blueprint con Amazon Q

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.

2. Nella CodeCatalyst console, accedi al progetto a cui desideri aggiungere un blueprint.
3. Scegli Aggiungi con Amazon Q.
4. Nel campo di immissione del testo del prompt di Amazon Q, fornisci istruzioni scrivendo una breve descrizione del progetto che desideri creare. Ad esempio, "I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic and functionality of the application, and a data layer that manages the storage and retrieval of the data."

(Facoltativo) In Prova esempi, puoi utilizzare un prompt già scritto scegliendo un blueprint. Ad esempio, se scegli l'app React, viene fornito il seguente prompt: "I want to create a project in Python that has a presentation layer responsible for how the data is presented, an application layer that contains the core logic and functionality of the application, and a data layer that manages the storage and retrieval of the data. I also want to add authentication and authorization mechanisms for security and allowable actions."

5. Scegli Invia per inviare le istruzioni ad Amazon Q. L'assistente generativo AI fornisce un suggerimento e delinea i requisiti che non possono essere soddisfatti dal modello. Ad esempio, Amazon Q potrebbe suggerire quanto segue in base ai tuoi criteri:

I recommend using the Single-page application blueprint based on your requirements. Blueprints are dynamic and can always be updated and edited later.

Single-page application

By Amazon Web Services

This blueprint creates a SPA (single-page application) using React, Vue, or Angular frameworks and deploys to AWS Amplify Hosting.

Version: 0.2.15

[View details](#)

The following requirements could not be met so I will create issues for you.

- The application should have reusable UI components
- The application should support for client-side routing
- The application may require server-side rendering for improved performance and SEO

6. (Facoltativo) Per visualizzare i dettagli approfonditi del blueprint suggerito, scegli Visualizza dettagli.
7. Esegui una delle seguenti operazioni:
 - a. Scegli Sì, usa questo blueprint se sei soddisfatto del suggerimento.
 - b. Scegli Modifica richiesta se desideri modificare la richiesta.
 - c. Scegli Ricomincia se desideri cancellare completamente il prompt.
8. Esegui una delle seguenti operazioni:
 - a. Scegli Configura se desideri configurare il blueprint suggerito. Puoi anche configurare il blueprint in un secondo momento.
 - b. Scegli Salta se al momento non desideri modificare le configurazioni del blueprint.
9. Se hai scelto di configurare il blueprint, scegli Continua dopo aver modificato le risorse del progetto.
10. Scegli Aggiungi al progetto per aggiungere risorse a un progetto con il blueprint. Amazon Q crea risorse utilizzando il blueprint. Ad esempio, se si aggiunge un progetto con il blueprint di un'applicazione a pagina singola, viene creato un repository di origine per il codice e i flussi di lavoro pertinenti per CI/CD.
11. (Facoltativo) Per impostazione predefinita, Amazon Q crea anche problemi per i requisiti che non sono soddisfatti da un blueprint. Puoi scegliere per quali articoli non vuoi creare problemi. Dopo aver scelto di lasciare che Amazon Q crei problemi, puoi assegnare un problema anche ad Amazon Q. Analizzerà il problema nel contesto degli archivi di origine indicati, fornendo un riepilogo dei file e del codice sorgente pertinenti. Per ulteriori informazioni, consultare [Crea un problema e assegnalo ad Amazon Q](#) e [Procedure consigliate per la creazione e l'utilizzo di problemi assegnati ad Amazon Q](#).

Crea un riepilogo delle modifiche al codice tra i rami durante la creazione di una pull request

Una pull request è il modo principale con cui tu e gli altri membri del progetto potete rivedere, commentare e unire le modifiche al codice da una filiale all'altra. Puoi utilizzare le pull request per esaminare le modifiche al codice in modo collaborativo per apportare modifiche o correzioni minori, aggiunte di funzionalità importanti o nuove versioni del software rilasciato. Riassumere le modifiche al codice e l'intento alla base delle modifiche come parte della descrizione della pull request è utile per gli altri utenti che esamineranno il codice e aiuta anche a comprendere lo storico delle modifiche.

apportate al codice nel tempo. Tuttavia, gli sviluppatori spesso si affidano al codice per spiegare se stesso o fornire dettagli ambigui anziché descrivere le modifiche con dettagli sufficienti per consentire ai revisori di capire cosa stanno esaminando o quale fosse l'intento alla base delle modifiche al codice.

Puoi utilizzare la funzione Write description for me durante la creazione di richieste pull per fare in modo che Amazon Q crei una descrizione delle modifiche contenute in una richiesta pull. Quando scegli questa opzione, Amazon Q analizza le differenze tra il ramo di origine che contiene le modifiche al codice e il ramo di destinazione in cui desideri unire queste modifiche. Quindi crea un riepilogo di quali sono tali modifiche, nonché la migliore interpretazione dell'intento e dell'effetto di tali modifiche.

Note

Questa funzionalità non funziona con i sottomoduli Git. Non riassumerà alcuna modifica in un sottomodulo Git che fa parte della pull request.

Questa funzionalità non è disponibile per le richieste pull nei repository collegati.

Puoi provare questa funzionalità con qualsiasi pull request che crei, ma in questo tutorial la testeremo apportando alcune semplici modifiche al codice contenuto in un progetto creato in un modello di applicazione web Modern a tre livelli basato su Python.

Tip

Se stai usando un progetto creato con un progetto diverso o con il tuo codice, puoi comunque seguire questo tutorial, ma gli esempi in questo tutorial non corrisponderanno al codice del tuo progetto. Invece dell'esempio suggerito di seguito, apporta semplici modifiche al codice del progetto in un ramo, quindi crea una pull request per testare la funzionalità, come mostrato nei passaggi seguenti.

Innanzitutto, creerai un ramo nel repository dei sorgenti. Quindi apporterai una rapida modifica al codice di un file in quel ramo utilizzando l'editor di testo nella console. Creerai quindi una pull request e utilizzerai la funzione Scrivi una descrizione per me per riepilogare le modifiche apportate.

Per creare un ramo (console)

1. Nella CodeCatalyst console, accedi al progetto in cui risiede il tuo repository di origine.

2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Scegli il repository in cui desideri creare un ramo.
4. Nella pagina di panoramica del repository, scegli Altro, quindi scegli Crea ramo.
5. Inserisci un nome per il ramo.
6. Scegliete un ramo da cui creare il ramo, quindi scegliete Crea.

Una volta creato un ramo, modifica un file in quel ramo con una semplice modifica. In questo esempio, modificherai il `test_endpoint.py` file per cambiare il numero di tentativi per i test da 0 3 a 5.

 Tip

Puoi anche scegliere di creare o utilizzare un ambiente di sviluppo per apportare questa modifica al codice. Per ulteriori informazioni, consulta [Creazione di un ambiente di sviluppo](#).

Per modificare il **`test_endpoint.py`** file nella console

1. Nella pagina di panoramica del repository di **`mysfits`** origine, scegli il menu a discesa del ramo e scegli il ramo creato nella procedura precedente.
2. In File, accedi al file che desideri modificare. Ad esempio, per modificare il `test_endpoint.py` file, espandi test, espandi integ e quindi scegli `test_endpoint.py`.
3. Scegli Modifica.
4. Alla riga 7, modifica il numero di volte in cui verranno ripetuti tutti i test:

```
def test_list_all(retry=3):
```

to:

```
def test_list_all(retry=5):
```

5. Scegli Commit e invia le modifiche alla tua filiale.

Ora che hai una filiale con una modifica, puoi creare una pull request.

Crea una pull request con un riepilogo delle modifiche

1. Nella pagina di panoramica del repository, scegli Altro, quindi scegli Crea richiesta pull.
2. Nel ramo di destinazione, scegli il ramo in cui unire il codice dopo averlo esaminato.

Tip

Scegli il ramo da cui hai creato il ramo nella procedura precedente per una dimostrazione più semplice di questa funzionalità. Ad esempio, se hai creato la filiale dal ramo predefinito del repository, scegli quel ramo come ramo di destinazione per la tua pull request.

3. Nel ramo Source, scegli il ramo che contiene le modifiche che hai appena eseguito nel `test_endpoint.py` file.
4. In Pull request title, inserisci un titolo che aiuti gli altri utenti a capire cosa deve essere esaminato e perché.
5. In Pull request description, scegli Write description for me per fare in modo che Amazon Q crei una descrizione delle modifiche contenute nella pull request.
6. Viene visualizzato un riepilogo delle modifiche. Controlla il testo suggerito, quindi scegli Accetta e aggiungi alla descrizione.
7. Facoltativamente, modifica il riepilogo per riflettere meglio le modifiche apportate al codice. Puoi anche scegliere di aggiungere revisori o collegare i problemi a questa pull request. Quando hai finito di apportare le modifiche aggiuntive che desideri, scegli Crea.

Crea un riepilogo dei commenti lasciati sulle modifiche al codice in una pull request

Quando gli utenti esaminano una pull request, spesso lasciano più commenti sulle modifiche apportate a quella pull request. Se ci sono molti commenti da parte di molti revisori, può essere difficile individuare temi comuni nel feedback o addirittura assicurarsi di aver esaminato tutti i commenti in tutte le revisioni. Puoi utilizzare la funzione Crea riepilogo commenti per consentire ad Amazon Q di analizzare tutti i commenti lasciati sulle modifiche al codice in una pull request e creare un riepilogo di tali commenti.

Note

I riepiloghi dei commenti sono temporanei. Se aggiorni una pull request, il riepilogo scomparirà. I riepiloghi dei contenuti non includono commenti sulla pull request complessiva, ma solo commenti lasciati sulle differenze di codice nelle revisioni della pull request. Questa funzionalità non funziona con i commenti lasciati sulle modifiche al codice nei sottomoduli Git. Questa funzionalità non è disponibile per le richieste pull nei repository collegati.

Per creare un riepilogo dei commenti in una pull request

1. Passa alla pull request creata nella procedura precedente.

Tip

Se preferisci, puoi utilizzare qualsiasi pull request aperta nel tuo progetto. Nella barra di navigazione, scegli Code, scegli Pull request e scegli una pull request aperta.

2. Aggiungi alcuni commenti alla pull request in Modifiche se la pull request non contiene già commenti.
3. In Panoramica, scegli Crea riepilogo dei commenti. Al termine, la sezione di riepilogo dei commenti si espande.
4. Esamina il riepilogo dei commenti lasciati sulle modifiche al codice nelle revisioni della pull request e confrontalo con i commenti nella pull request.

Crea un problema e assegnalo ad Amazon Q

I team di sviluppo creano problemi per tracciare e gestire il proprio lavoro, ma a volte un problema persiste perché non è chiaro chi debba occuparsene, oppure il problema richiede una ricerca su una parte particolare della base di codice o è necessario prima occuparsi di altri lavori urgenti. CodeCatalyst include l'integrazione con Amazon Q Developer Agent per lo sviluppo di software. Puoi assegnare i problemi a un assistente AI generativo chiamato Amazon Q che può analizzare un problema in base al titolo e alla descrizione. Se assegni il problema ad Amazon Q, quest'ultima tenterà di creare una bozza di soluzione da valutare. Questo può aiutare te e il tuo team a concentrarti e ottimizzare il lavoro su questioni che richiedono la tua attenzione, mentre Amazon Q lavora a una soluzione per problemi che non hai risorse per risolvere immediatamente.

 Tip

Amazon Q offre le migliori prestazioni su problemi semplici e diretti. Per ottenere i migliori risultati, usa un linguaggio semplice per spiegare chiaramente cosa vuoi fare.

Quando assegni un problema ad Amazon Q, CodeCatalyst lo contrassegnerà come bloccato finché non confermerai come desideri che Amazon Q risolva il problema. È necessario rispondere a tre domande prima di continuare:

- Sia che tu voglia confermare ogni passaggio necessario o che proceda senza feedback. Se scegli di confermare ogni passaggio, puoi rispondere ad Amazon Q con un feedback sull'approccio che ha creato, in modo che possa ripetere il suo approccio, se necessario. Amazon Q può anche esaminare il feedback che gli utenti lasciano su qualsiasi pull request creata se scegli questa opzione. Se scegli di non confermare ogni passaggio, Amazon Q potrebbe completare il lavoro più rapidamente, ma non esaminerà alcun feedback fornito nel problema o nelle pull request che crea.
- Sia che tu voglia consentirgli di aggiornare i file del flusso di lavoro come parte del suo lavoro. Il tuo progetto potrebbe avere flussi di lavoro configurati per avviare esecuzioni in base a eventi di pull request. In tal caso, qualsiasi richiesta pull creata da Amazon Q che includa la creazione o l'aggiornamento del flusso di lavoro YAML potrebbe avviare un'esecuzione dei flussi di lavoro inclusi nella richiesta pull. Come best practice, non scegliere di consentire ad Amazon Q di lavorare su file di workflow a meno che tu non sia sicuro che nel tuo progetto non ci siano flussi di lavoro che eseguano automaticamente tali flussi di lavoro prima di rivedere e approvare la pull request creata.
- Se vuoi consentirgli di suggerire la creazione di attività per suddividere il lavoro sul problema in incrementi più piccoli che possono essere assegnati individualmente agli utenti, incluso Amazon Q stesso. Consentire ad Amazon Q di suggerire e creare attività può aiutare ad accelerare lo sviluppo di problemi complessi, permettendo a più persone di lavorare su parti distinte del problema. Può anche aiutare a ridurre la complessità della comprensione dell'intero lavoro, poiché il lavoro necessario per completare ogni attività è idealmente più semplice del problema a cui appartiene.
- In quale archivio sorgente vuoi che funzioni. Anche se il tuo progetto ha più repository di sorgenti, Amazon Q può lavorare solo sul codice in un unico repository di sorgenti. Gli archivi collegati non sono supportati.

Dopo aver effettuato e confermato le tue scelte, Amazon Q sposterà il problema nello stato In corso mentre tenta di determinare la richiesta in base al titolo del problema e alla sua descrizione, nonché

al codice nel repository specificato. Creerà un commento aggiunto in cui fornirà aggiornamenti sullo stato del suo lavoro. Dopo aver esaminato i dati, Amazon Q formulerà un potenziale approccio a una soluzione. Amazon Q registra le sue azioni aggiornando il commento aggiunto e commentando lo stato di avanzamento del problema in ogni fase. A differenza dei commenti e delle risposte bloccati, non conserva una registrazione strettamente cronologica del suo lavoro. Piuttosto, inserisce le informazioni più rilevanti sul suo lavoro al primo livello del commento aggiunto. Tenterà di creare codice in base al suo approccio e alla sua analisi del codice già presente nel repository. Se genera con successo una potenziale soluzione, creerà un ramo e inserirà il codice in quel ramo. Quindi crea una pull request che unirà quel ramo con il ramo predefinito. Quando Amazon Q completa il suo lavoro, sposta il problema su In review in modo che tu e il tuo team sappiate che c'è un codice pronto per la valutazione.

Note

Questa funzionalità è disponibile solo tramite Issues in the US West (Oregon) Region. Non è disponibile se hai configurato il progetto per utilizzare Jira con l'estensione Jira Software. Inoltre, se hai personalizzato il layout della scheda, il problema potrebbe non modificare lo stato. Per ottenere i migliori risultati, usa questa funzionalità solo con progetti con un layout di scheda standard.

Questa funzionalità non funziona con i sottomoduli Git. Non può apportare modifiche a nessun sottomodulo Git incluso nel repository.

Dopo aver assegnato un problema ad Amazon Q, non puoi modificare il titolo o la descrizione del problema o assegnarlo a nessun altro. Se annulli l'assegnazione di Amazon Q al problema, completerà la fase attuale e poi smetterà di funzionare. Una volta annullata l'assegnazione, non può riprendere a funzionare o essere riassegnato al problema.

Un problema può essere spostato automaticamente nella colonna In revisione se assegnato ad Amazon Q se un utente sceglie di consentirgli di creare attività. Tuttavia, il problema in In revisione potrebbe avere ancora attività che si trovano in uno stato diverso, ad esempio nello stato In corso.

In questa parte del tutorial, creerai tre problemi basati sulle potenziali funzionalità del codice incluse nei progetti creati con il blueprint di un'applicazione web moderna a tre livelli: uno per aggiungere uno per creare una nuova creatura mysfit, uno per aggiungere una funzionalità di ordinamento e uno per aggiornare un flusso di lavoro per includere un ramo denominato. **test**

 Note

Se stai lavorando a un progetto con codice diverso, crea problemi con titoli e descrizioni relativi a quel codice base.

Per creare un problema e far sì che venga generata una soluzione da valutare

1. Nel riquadro di navigazione, scegli Problemi e assicurati di essere nella visualizzazione Bacheca.
2. Scegli Crea problema.
3. Assegna al numero un titolo che spieghi cosa vuoi fare in un linguaggio semplice. Ad esempio, per questo problema, inserisci un titolo di **Create another mysfit named Quokkapus**. In Descrizione, fornisci i seguenti dettagli:

Expand the table of mysfits to 13, and give the new mysfit the following characteristics:

Name: Quokkapus

Species: Quokka-Octopus hybrid

Good/Evil: Good

Lawful/Chaotic: Chaotic

Age: 216

Description: Australia is full of amazing marsupials, but there's nothing there quite like the Quokkapus.

She's always got a friendly smile on her face, especially when she's using her eight limbs to wrap you up in a great big hug. She exists on a diet of code bugs and caffeine. If you've got some gnarly code that needs assistance, adopt Quokkapus and put her to work - she'll love it! Just make sure you leave enough room for her to grow, and keep that coffee coming.

4. (Facoltativo) Allega al problema un'immagine da utilizzare come miniatura e immagine del profilo per mysfit. Se lo fai, aggiorna la descrizione per includere dettagli su quali immagini vuoi usare e perché. Ad esempio, potresti aggiungere quanto segue alla descrizione: «Il mysfit richiede che

i file di immagine vengano distribuiti sul sito web. Aggiungi queste immagini allegate a questo numero all'archivio di origine come parte del lavoro e distribuiscile sul sito web».

Note

Le immagini allegate potrebbero o meno essere distribuite sul sito Web durante le interazioni di questo tutorial. Puoi aggiungere tu stesso le immagini al sito Web e quindi lasciare commenti affinché Amazon Q aggiorni il codice in modo che punti alle immagini che desideri che utilizzi dopo aver creato una pull request.

Controlla la descrizione e assicurati che contenga tutti i dettagli che potrebbero essere necessari prima di procedere al passaggio successivo.

5. In Assegnatari, scegli Assegna ad Amazon Q.
6. Nel repository Source, scegli il repository di origine che contiene il codice del progetto.
7. Fai scorrere il pulsante Require Amazon Q per interromperlo dopo ogni passaggio e, se necessario, attendi che il selettore di lavoro ritorni attivo.

Note

Scegliendo l'opzione di interruzione di Amazon Q dopo ogni passaggio, puoi commentare il problema o qualsiasi attività creata e avere la possibilità che Amazon Q cambi il suo approccio fino a tre volte in base ai tuoi commenti. Se scegli di non lasciare che Amazon Q si fermi dopo ogni fase in modo da poterne controllare il funzionamento, il lavoro potrebbe procedere più rapidamente perché Amazon Q non aspetta il tuo feedback, ma non sarai in grado di influenzare la direzione intrapresa da Amazon Q lasciando commenti. Inoltre, Amazon Q non risponderà ai commenti lasciati in una pull request se scegli questa opzione.

8. Lascia il selettore Consenti ad Amazon Q di modificare i file del flusso di lavoro nello stato inattivo.
9. Fai scorrere il selettore Consenti ad Amazon Q di suggerire la creazione di attività sullo stato attivo.
10. Scegli Crea problema. La tua visualizzazione viene modificata nella scheda Problemi.

11. Scegli Crea problema per creare un altro numero, questa volta con lo stesso titolo **Change the get_all_mysfits() API to return mysfits sorted by the Age attribute**. Assegna questo problema ad Amazon Q e crea il problema.
12. Scegli Crea problema per creare un altro numero, questa volta con lo stesso titolo **Update the OnPullRequest workflow to include a branch named test in its triggers**. Se lo desideri, inserisci un link al flusso di lavoro nella descrizione. Assegna questo problema ad Amazon Q ma questa volta assicurati che il selettore Consenti ad Amazon Q di modificare i file del flusso di lavoro sia impostato sullo stato attivo. Crea il problema per tornare alla scheda Problemi.

 Tip

È possibile cercare file, inclusi i file del flusso di lavoro, inserendo il simbolo at (@) e il nome del file.

Dopo aver creato e assegnato i problemi, i problemi passeranno alla sezione In corso. Amazon Q aggiungerà commenti che ne tracciano l'avanzamento all'interno del problema in un commento aggiunto. Se è in grado di definire un approccio a una soluzione, aggiornerà la descrizione del problema con una sezione Background che contiene l'analisi del codice base e una sezione Approccio che descrive in dettaglio l'approccio proposto per la creazione di una soluzione. Se Amazon Q riesce a trovare una soluzione al problema descritto nel problema, creerà una filiale e apporterà modifiche al codice in quella filiale che implementeranno la soluzione proposta. Se il codice proposto contiene somiglianze con il codice open source di cui Amazon Q è a conoscenza, fornirà un file che include collegamenti a quel codice in modo che tu possa esaminarlo. Una volta che il codice è pronto, crea una pull request in modo da poter esaminare le modifiche al codice suggerite, aggiunge un link a quella pull request al problema e sposta il problema in In review.

 Important

Dovresti sempre rivedere tutte le modifiche al codice in una pull request prima di unirla. L'unione delle modifiche al codice apportate da Amazon Q, come qualsiasi altra modifica al codice, può avere un impatto negativo sulla base di codice e sul codice dell'infrastruttura se il codice unito non viene esaminato correttamente e contiene errori durante l'unione.

Per esaminare un problema e una pull request collegata che contiene le modifiche apportate da Amazon Q

1. In Problemi, scegli un problema assegnato ad Amazon Q che è in corso. Esamina i commenti per monitorare lo stato di avanzamento di Amazon Q. Se presenti, esamina lo sfondo e consulta i record nella descrizione del problema. Se hai scelto di consentire ad Amazon Q di suggerire attività, esamina le attività proposte e intraprendi le azioni necessarie. Ad esempio, se Amazon Q ha suggerito delle attività e desideri modificare l'ordine o assegnare attività a utenti specifici, scegli Modifica, aggiungi o riordina le attività ed esegui gli aggiornamenti necessari. Quando hai finito di visualizzare il problema, scegli X per chiudere il riquadro del problema.

 Tip

Per visualizzare lo stato di avanzamento delle attività, scegli l'attività dall'elenco delle attività del numero. Le attività non vengono visualizzate come elementi separati sulla lavagna e sono accessibili solo tramite un numero. Se un'attività è assegnata ad Amazon Q, devi aprirla per approvare le azioni che desidera eseguire. È inoltre necessario aprire un'attività per visualizzare le richieste pull collegate, in quanto non verranno visualizzate come collegamenti nel problema, ma solo nell'attività. Per tornare a un problema da un'attività, scegli il link al problema.

2. Ora scegli un problema assegnato ad Amazon Q che si trova in In revisione. Esamina lo sfondo e consulta i dati riportati nella descrizione del problema. Esamina i commenti per comprendere le azioni che ha eseguito. Esamina tutte le attività create per il lavoro correlato a questo problema, inclusi lo stato di avanzamento, le azioni che potresti dover intraprendere e gli eventuali commenti. Nelle richieste Pull, scegli il link alla pull request accanto all'etichetta Open per rivedere il codice.

 Tip

Le richieste pull generate per le attività vengono visualizzate solo come richieste pull collegate nella visualizzazione delle attività. Non vengono visualizzate come richieste pull collegate al problema.

3. Nella pull request, esamina le modifiche al codice. Per ulteriori informazioni, consulta [Revisione di una richiesta di pull](#). Lascia commenti sulla pull request se desideri che Amazon Q modifichi uno dei suoi codici suggeriti. Sii specifico quando lasci commenti per Amazon Q per ottenere i migliori risultati.

Ad esempio, quando esamini la pull request creata per **Create another mysfit named Quokkapus**, potresti notare che c'è un errore di battitura nella descrizione. Puoi lasciare un commento per Amazon Q che indichi «Modifica la descrizione per correggere l'errore di battitura «needsa» aggiungendo uno spazio tra «needs» e «a».» In alternativa, puoi lasciare un commento che indichi ad Amazon Q di aggiornare la descrizione e fornire l'intera descrizione rivista da incorporare.

Se hai caricato immagini per il nuovo mysfit sul sito Web, puoi lasciare un commento ad Amazon Q per aggiornare mysfit con puntatori all'immagine e miniature da utilizzare per il nuovo mysfit.

Note

Amazon Q non risponderà ai commenti individuali. Amazon Q includerà il feedback lasciato nei commenti nelle pull request solo se hai scelto l'opzione predefinita di fermarti dopo ogni passaggio per l'approvazione quando hai creato il problema.

4. (Facoltativo) Dopo che tu e gli altri utenti del progetto avete lasciato tutti i commenti desiderati per le modifiche al codice, scegli Crea revisione per fare in modo che Amazon Q crei una revisione della pull request che incorpori le modifiche richieste nei commenti. Lo stato di avanzamento della creazione della revisione verrà segnalato da Amazon Q in Panoramica, non in Modifiche. Assicurati di aggiornare il browser per visualizzare gli ultimi aggiornamenti di Amazon Q sulla creazione della revisione.

Note

Solo l'utente che ha creato il problema può creare una revisione della pull request. Puoi richiedere una sola revisione di una pull request. Assicurati di aver risolto tutti i problemi relativi ai commenti e di essere soddisfatto del contenuto dei commenti prima di scegliere Crea revisione.

5. In questo progetto di esempio viene eseguito un flusso di lavoro per ogni pull request. Assicurati di vedere un flusso di lavoro eseguito correttamente prima di unire la pull request. Puoi anche scegliere di creare flussi di lavoro e ambienti aggiuntivi per testare il codice prima di unirli. Per ulteriori informazioni, consulta [Guida introduttiva ai flussi di lavoro](#).
6. Quando sei soddisfatto dell'ultima revisione della pull request, scegli Merge.

Crea un problema e chiedi ad Amazon Q di eseguire delle attività consigliate per risolverlo

A volte un problema può comportare una mole di lavoro lunga o complessa. CodeCatalyst include l'integrazione con Amazon Q Developer Agent per lo sviluppo di software. Puoi chiedere ad Amazon Q di analizzare un problema in base al titolo e alla descrizione e consigliare una suddivisione logica del lavoro in attività separate. Tenterà di creare un elenco di attività consigliate da rivedere, modificare e scegliere se creare o meno. Questo può aiutare te e il tuo team ad assegnare singole parti del lavoro agli utenti in modi più gestibili e più rapidamente.

Per creare ed esaminare un elenco di attività consigliate per un problema

1. Nel riquadro di navigazione, scegli Problemi e assicurati di essere nella visualizzazione Bacheca.
2. Scegli Crea problema.
3. Assegna al numero un titolo che spieghi cosa vuoi fare in un linguaggio semplice. Ad esempio, per questo problema, inserisci un titolo di **Change the `get_all_mysfits()` API to return mysfits sorted by the Good/Evil attribute**. In Descrizione, fornisci i seguenti dettagli:

Update the API to allow sorting of mysfits by whether they are Good, Neutral, or Evil. Add a button on the website that allows users to quickly choose this sort and to exclude alignments that they don't want to see.

4. Controlla la descrizione e assicurati che contenga tutti i dettagli che potrebbero essere necessari prima di procedere al passaggio successivo.
5. In Assegnatari, scegli di assegnare il problema a te stesso.
6. Scegli Crea problema. La tua visualizzazione viene modificata nella scheda Problemi.
7. Scegli il problema che hai appena creato per aprirlo. Scegli Consiglia attività.
8. Scegli il repository di origine che contiene il codice per i problemi. Scegli Inizia a consigliare attività.

La finestra di dialogo si chiuderà e Amazon Q inizierà ad analizzare la complessità del problema. Se il problema è complesso, suggerirà una suddivisione del lavoro in attività sequenziali separate. Quando l'elenco è pronto, scegli Visualizza attività consigliate. Puoi aggiungere attività aggiuntive, modificare le attività consigliate e riordinare le attività. Se sei d'accordo con i consigli, selezionando Crea attività

verranno create le attività. Puoi quindi assegnare tali attività agli utenti affinché ci lavorino o anche ad Amazon Q stesso.

Eseguire la pulizia delle risorse

Una volta completato questo tutorial, prendi in considerazione l'idea di intraprendere le seguenti azioni per ripulire tutte le risorse che hai creato durante questo tutorial che non ti servono più.

- Annulla Amazon Q da eventuali problemi su cui non stai più lavorando. Se Amazon Q ha terminato il lavoro su un problema o non è riuscito a trovare una soluzione, assicurati di annullare l'assegnazione di Amazon Q per evitare di raggiungere la quota massima per le funzionalità di intelligenza artificiale generativa. [Per ulteriori informazioni, consulta Gestione delle funzionalità e prezzi dell'IA generativa.](#)
- Sposta tutti i problemi in cui il lavoro è completo su Fine.
- Se il progetto non è più necessario, eliminalo.

Organizza le risorse con spazi in CodeCatalyst

Crei uno spazio che rappresenta te, la tua azienda, il tuo reparto o gruppo e fornisce un luogo in cui i team di sviluppo possono gestire i progetti. Devi creare uno spazio per aggiungere progetti, membri e le risorse cloud associate che crei in Amazon CodeCatalyst.

Note

I nomi degli spazi devono essere univoci in tutto CodeCatalyst. Non è possibile riutilizzare i nomi degli spazi eliminati.

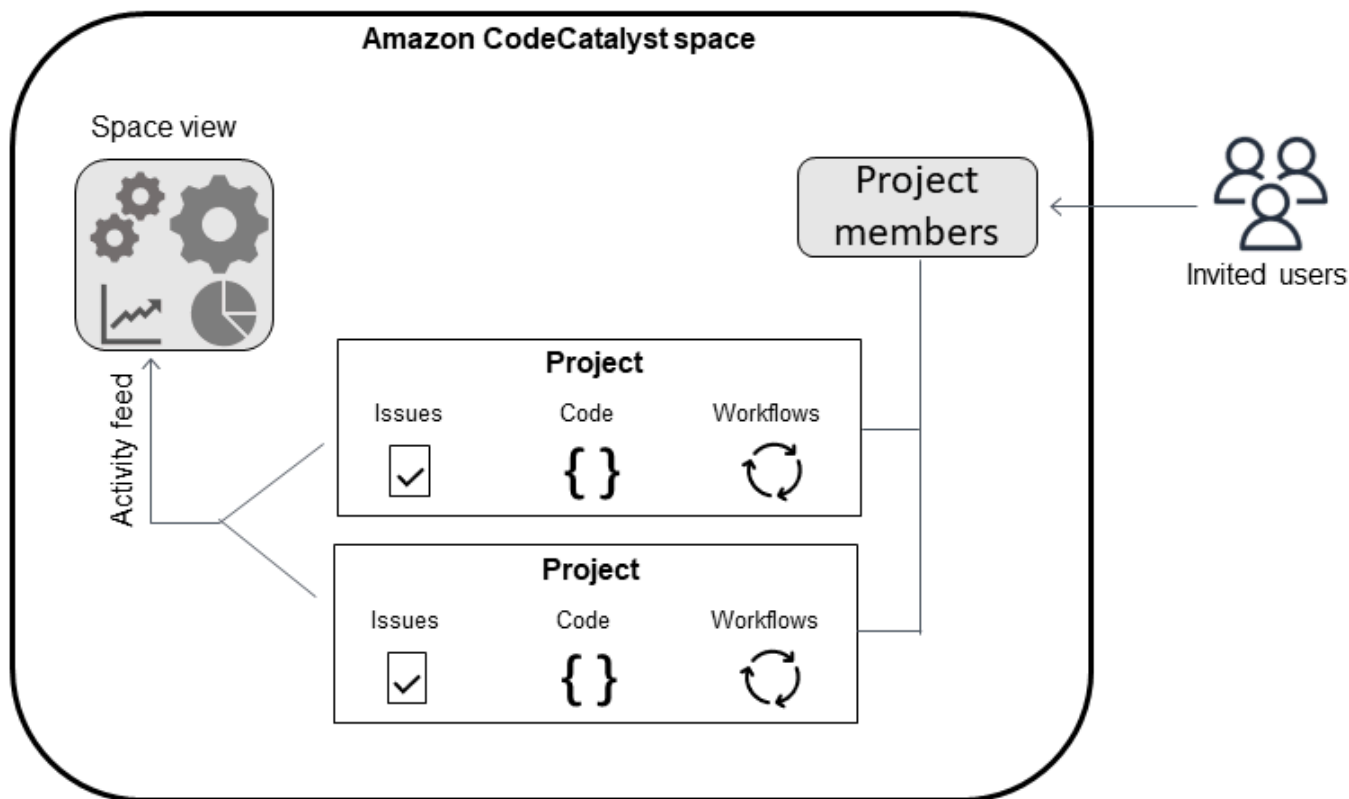
Quando crei uno spazio, ti viene assegnato automaticamente il ruolo di amministratore dello spazio. È possibile aggiungere questo ruolo ad altri utenti nello spazio.

Con il ruolo di amministratore dello spazio, puoi gestire lo spazio come segue:

- Aggiungere altri amministratori dello spazio allo spazio
- Modifica i ruoli e le autorizzazioni dei membri
- Modifica o elimina lo spazio
- Crea progetti e invita membri al progetto
- Visualizza un elenco di tutti i progetti presenti nello spazio
- Visualizza il feed delle attività per tutti i progetti nello spazio

Quando si crea uno spazio, si viene automaticamente aggiunti allo spazio con due ruoli: il ruolo di amministratore dello spazio e il ruolo di amministratore del progetto per il progetto creato durante la creazione dello spazio. Gli utenti aggiuntivi vengono aggiunti automaticamente come membri allo spazio quando accettano inviti a partecipare a progetti. Questa appartenenza allo spazio non concede alcuna autorizzazione nello spazio. Ciò che gli utenti possono fare in uno spazio è determinato dal ruolo che l'utente ricopre in un progetto specifico.

Per ulteriori informazioni sui ruoli, consulta [Concessione dell'accesso con ruoli utente](#).



Di seguito sono riportate ulteriori considerazioni relative agli account aggiunti:

- Account AWS aggiunto a uno CodeCatalyst spazio può essere utilizzato in qualsiasi progetto in quello spazio.
- Sebbene ogni ambiente possa supportarne più di uno Account AWS, è possibile utilizzare un solo account per ambiente in un'azione.
- La fatturazione è configurata a livello di spazio. È possibile configurare più account per la fatturazione, ma solo uno può essere attivo in uno CodeCatalyst spazio. An Account AWS può essere utilizzato come account di fatturazione per più di uno spazio. CodeCatalyst L'account Account AWS di fatturazione specificato per il tuo CodeCatalyst spazio ha quote diverse rispetto agli altri account connessi a uno spazio. Per ulteriori informazioni, consulta [Quote per CodeCatalyst](#).
- Dopo aver creato una connessione, è necessario aggiungere i ruoli AWS IAM alla connessione se il flusso di lavoro deve accedere a tali ruoli IAM con il proprio CodeCatalyst ambiente. Per ulteriori informazioni su come vengono utilizzati gli ambienti, consulta [Implementazione in e Account AWS VPCs](#).

Argomenti

- [Creare uno spazio](#)
- [Modificare uno spazio](#)
- [Eliminazione di uno spazio](#)
- [Monitoraggio dell'attività per utenti e risorse in uno spazio](#)
- [Consentire l'accesso alle AWS risorse con connessione Account AWS](#)
- [Configurazione dei ruoli IAM per gli account connessi](#)
- [Concessione delle autorizzazioni di spazio agli utenti](#)
- [Consentire l'accesso allo spazio tramite team](#)
- [Consentire l'accesso allo spazio per le risorse della macchina](#)
- [Amministrazione degli ambienti di sviluppo per uno spazio](#)
- [Quote per gli spazi](#)

Creare uno spazio

Quando ti registri per la prima volta su Amazon CodeCatalyst con il tuo ID AWS Builder, devi creare uno spazio. Per ulteriori informazioni, consulta [Configura e accedi a CodeCatalyst](#). Puoi scegliere di creare spazi aggiuntivi per soddisfare le tue esigenze aziendali.

Note

I nomi degli spazi devono essere univoci CodeCatalyst. Non è possibile riutilizzare i nomi degli spazi eliminati.

Le informazioni contenute in questa guida vengono fornite per creare spazi CodeCatalyst che supportano gli utenti AWS Builder ID. I passaggi per configurare e amministrare uno spazio che supporti la federazione delle identità sono disponibili nella Guida per l'CodeCatalyst amministratore. Per lavorare con spazi configurati per la federazione delle identità, consulta [Configurazione e amministrazione CodeCatalyst degli spazi](#) nella Amazon CodeCatalyst Administrator Guide.

Per creare spazi aggiuntivi che supportino gli utenti AWS Builder ID, devi avere il ruolo di amministratore di Space.

 Note

Quando crei uno spazio aggiuntivo, non ti viene richiesto di creare un progetto. Per informazioni su come creare progetti in uno spazio, consulta [Creare un progetto](#).

Per creare un altro spazio

1. Nel Console di gestione AWS, assicurati di aver effettuato l'accesso con lo stesso Account AWS che desideri associare al tuo CodeCatalyst spazio.
2. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Vai al tuo spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

4. Scegli Crea spazio.
5. Nella pagina Crea uno spazio, in Nome spazio, inserisci un nome per lo spazio. Non è possibile modificarlo in un secondo momento.

 Note

I nomi degli spazi devono essere univoci CodeCatalyst. Non è possibile riutilizzare i nomi degli spazi eliminati.

6. In Regione AWS, scegli la regione in cui desideri archiviare lo spazio e i dati del progetto. Non puoi modificarlo in un secondo momento.
7. In Account AWS ID, inserisci l'ID a dodici cifre dell'account che desideri connettere al tuo spazio.

Nel token di verifica dell'AWS account, copia l'ID del token generato. Il token viene copiato automaticamente per te, ma potresti volerlo archiviare mentre approvi la richiesta di AWS connessione.

8. Scegli Verifica in. AWS
9. La pagina Verify Amazon CodeCatalyst Space si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

Nel Console di gestione AWS, assicurati di scegliere lo stesso Regione AWS luogo in cui desideri creare il tuo spazio.

Per accedere direttamente alla pagina, accedi ad Amazon CodeCatalyst Spaces nella Console di gestione AWS pagina <https://console.aws.amazon.com/codecatalyst/> iniziale/.

Il token di verifica viene inserito automaticamente nel token di verifica. Un banner di successo mostra un messaggio che indica che il token è un token valido.

10. Scegli Verifica lo spazio.

Viene visualizzato un messaggio di verifica dell'account che indica che l'account è stato aggiunto allo spazio.

11. Rimani sulla pagina Verify Amazon CodeCatalyst Space. Scegli il seguente link: Per aggiungere ruoli IAM per questo spazio, visualizza i dettagli dello spazio.

La pagina dei dettagli CodeCatalyst dello spazio si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

12. In Ruoli IAM disponibili per CodeCatalyst, scegli Aggiungi ruolo IAM.

Viene visualizzata la pagina Aggiungi ruoli IAM disponibili alla CodeCatalyst pagina.

13. Scegli Crea il ruolo di amministratore CodeCatalyst dello sviluppo in IAM. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di fiducia per il ruolo di sviluppo.

Il ruolo di sviluppatore è un ruolo AWS IAM che consente ai CodeCatalyst flussi di lavoro di accedere a AWS risorse come Amazon S3, Lambda e. CloudFormation Il ruolo avrà un nome CodeCatalystWorkflowDevelopmentRole-*spaceName* con un identificatore univoco aggiunto. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere.

[Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName*del servizio](#)

14. Scegli Crea ruolo di sviluppo.

15. Nella pagina di connessione, in Ruoli IAM disponibili per CodeCatalyst, visualizza il ruolo di sviluppatore nell'elenco dei ruoli IAM aggiunti al tuo account.

16. Scegli Vai ad Amazon CodeCatalyst.

17. Nella pagina di creazione CodeCatalyst, scegli Crea spazio.

Modificare uno spazio

Puoi modificare la descrizione di uno spazio per aiutare gli utenti a capire meglio a cosa serve.

È necessario disporre del ruolo di amministratore dello spazio per modificare i dettagli dello spazio.

Le informazioni contenute in questa guida vengono fornite per modificare gli spazi CodeCatalyst che supportano gli utenti di AWS Builder ID. Per ulteriori informazioni sui passaggi per configurare e amministrare uno spazio che supporti la federazione delle identità, consulta [Setup and administration for CodeCatalyst spaces](#) nella Amazon CodeCatalyst Administrator Guide.

Per modificare la descrizione di uno spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio.

Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Nella scheda Impostazioni dello spazio, scegli Modifica. Apporta le modifiche desiderate alla descrizione dello spazio, quindi scegli Salva.

Eliminazione di uno spazio

Puoi eliminare uno spazio per rimuovere l'accesso a tutte le risorse dello spazio. È necessario disporre del ruolo di amministratore dello spazio per eliminare uno spazio.

Note

Non è possibile annullare l'eliminazione di uno spazio.

Dopo aver eliminato uno spazio, tutti i membri dello spazio non saranno in grado di accedere alle risorse dello spazio. Inoltre, la fatturazione delle risorse spaziali verrà interrotta e tutti i flussi di lavoro richiesti dagli archivi di sorgenti di terze parti verranno interrotti.

 Note

I nomi degli spazi devono essere univoci per tutti. CodeCatalyst Non è possibile riutilizzare i nomi degli spazi eliminati.

Le informazioni contenute in questa guida vengono fornite per l'eliminazione di spazi CodeCatalyst che supportano gli utenti AWS Builder ID. Per ulteriori informazioni sui passaggi per configurare e amministrare uno spazio che supporti la federazione delle identità, consulta [Setup and administration for CodeCatalyst spaces](#) nella Amazon CodeCatalyst Administrator Guide.

Per eliminare uno spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Elimina.
4. Digita **delete** per confermare l'eliminazione.
5. Scegli Elimina.

 Note

Se appartieni a più di uno spazio, verrai reindirizzato alla pagina di panoramica dello spazio. Se appartieni a uno spazio, verrai reindirizzato alla pagina di creazione dello spazio.

Monitoraggio dell'attività per utenti e risorse in uno spazio

Per visualizzare i progetti creati di recente e gli aggiornamenti sullo stato, puoi utilizzare la CodeCatalyst console per visualizzare un feed di attività che mostra gli aggiornamenti per le risorse spaziali.

Nel feed delle attività, puoi visualizzare metriche come le esecuzioni non riuscite del flusso di lavoro e i progetti creati.

Per visualizzare le attività nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegliere Activity (Attività).
4. Visualizza le informazioni in Attività.
5. Per filtrare per attività, scegli il selettore in alto a destra.
6. Per visualizzare tutte le attività nel tuo spazio, scegli Qualsiasi tipo di attività.

Consentire l'accesso alle AWS risorse con connessione Account AWS

Puoi utilizzare le risorse dei tuoi Account AWS CodeCatalyst spazi Amazon. A tale scopo, devi configurare una connessione tra il Account AWS e il tuo spazio in CodeCatalyst. La creazione di una connessione come questa significa che i progetti e i flussi di lavoro all'interno del tuo CodeCatalyst spazio possono interagire con le risorse presenti nel tuo Account AWS. Devi creare una connessione per ogni connessione Account AWS che desideri utilizzare con il tuo CodeCatalyst spazio.

Dopo aver creato una connessione, puoi scegliere di associarvi i ruoli AWS IAM.

Argomenti

- [Aggiungere un Account AWS a uno spazio](#)
- [Aggiungere ruoli IAM alle connessioni degli account](#)
- [Aggiungere la connessione all'account e i ruoli IAM all'ambiente di distribuzione](#)
- [Visualizzazione delle connessioni degli account](#)
- [Eliminazione delle connessioni all'account \(in CodeCatalyst\)](#)
- [Configurazione di un account di fatturazione per uno spazio](#)

Puoi configurarlo CodeCatalyst per l'utilizzo autorizzato Account AWS aggiungendo gli account al tuo spazio. Aggiungendo spazio Account AWS al tuo CodeCatalyst spazio, puoi consentire ai flussi di lavoro del progetto di accedere alle Account AWS risorse e alla configurazione di fatturazione.

L'aggiunta di un Account AWS crea una connessione che CodeCatalyst autorizza l'utilizzo di questo account. Puoi usare added Account AWS per fare quanto segue:

- Configura la fatturazione per uno CodeCatalyst spazio. Consulta [la sezione Gestione della fatturazione](#) nella Amazon CodeCatalyst Administrator Guide. L' Account AWS account specificato come account di fatturazione per il tuo CodeCatalyst spazio ha quote diverse rispetto agli altri account connessi allo stesso spazio. Per ulteriori informazioni, consulta [Quote per CodeCatalyst](#).
- Consenti CodeCatalyst di assumere ruoli IAM per accedere alle AWS risorse e distribuirle Servizi AWS nell'account. Per informazioni, consulta [Configurazione dei ruoli IAM per gli account connessi](#).

Le connessioni all'account vengono create completando l'autorizzazione con. Account AWS Dopo aver creato la connessione, configuri ulteriormente la connessione per i flussi di lavoro e i progetti da utilizzare aggiungendo ruoli IAM.

Per i passaggi per configurare le connessioni degli account nella Console di gestione AWS pagina relativa alla funzione di CodeCatalyst amministratore dello spazio Account AWS e dello spazio, consulta la sezione [Gestione degli account connessi](#) nella Guida per l'CodeCatalyst amministratore. Le connessioni degli account possono essere configurate in modo da limitarle a progetti specifici. Puoi associare flussi di lavoro o connessioni VPC solo a qualcuno che Account AWS ha accesso al tuo progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#).

Aggiungere un Account AWS a uno spazio

Utilizzi la CodeCatalyst console e il Console di gestione AWS per connettere il tuo spazio a un Account AWS.

Prima di aggiungere uno Account AWS a uno spazio in CodeCatalyst, completa i seguenti prerequisiti:

- Crea un account Account AWS e acquisisci le autorizzazioni per creare ruoli AWS IAM nell'account che desideri connettere.
- Crea il ruolo o i ruoli IAM che desideri associare alla connessione del tuo account, incluse le politiche IAM con le autorizzazioni per i ruoli.

- Acquisisci il ruolo di amministratore dello CodeCatalyst spazio nello spazio in cui desideri creare la connessione.

Argomenti

- [Passaggio 1: creazione di una richiesta di connessione](#)
- [Fase 2: Accettazione di una richiesta di connessione all'account](#)
- [Passaggio 3: verifica di una connessione approvata](#)
- [Passaggio 4: aggiungi i ruoli IAM alla tua connessione](#)
- [Prossimi passaggi: crea ruoli IAM aggiuntivi per la connessione del tuo account](#)

Passaggio 1: creazione di una richiesta di connessione

La creazione di una richiesta di connessione nella CodeCatalyst console genera un token di connessione che è possibile utilizzare per completare l'autorizzazione.

È necessario disporre del ruolo di amministratore dello spazio o Power user nello CodeCatalyst spazio in cui si desidera creare la connessione. È inoltre necessario disporre delle autorizzazioni amministrative per il file Account AWS che si desidera aggiungere.

Come creare una connessione

1. Nel Console di gestione AWS, assicurati di aver effettuato l'accesso con lo stesso account con cui desideri creare una connessione.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
4. Scegli Aggiungi un Account AWS.
5. Nella CodeCatalyst pagina Associa Account AWS ad Amazon, in Account AWS ID, inserisci l'ID a dodici cifre per l'account che desideri connettere al tuo spazio. Per informazioni su come trovare il tuo Account AWS ID, consulta Il tuo ID e [il suo alias Account AWS](#).
6. In Amazon CodeCatalyst display name, inserisci un nome di riferimento per l'account.
7. (Facoltativo) In Descrizione della connessione, inserisci una descrizione dell'account che ti aiuterà a scegliere i progetti a cui applicare l'account e il ruolo o i ruoli.
8. Selezionare Associate (Associa) Account AWS.

9. La pagina torna alla pagina dei Account AWS dettagli in cui viene visualizzato un banner di successo.

Fase 2: Accettazione di una richiesta di connessione all'account

Dopo aver inviato una richiesta di connessione nella CodeCatalyst console Account AWS, collabora con l' AWS amministratore per accettare la richiesta di connessione inviandola con il token di connessione fornito.

Assicurati di disporre delle autorizzazioni di amministratore per il tuo account e di aver effettuato l'accesso Console di gestione AWS con le stesse autorizzazioni Account AWS per cui stai creando la connessione.

Per approvare una richiesta di connessione (console)

1. Nel Console di gestione AWS, assicurati di aver effettuato l'accesso con lo stesso account con cui desideri creare una connessione.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
4. Nella pagina dei Account AWS dettagli, scegli Configurazione completa in Console di gestione AWS.
5. La pagina Verify Amazon CodeCatalyst Space si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

Per accedere direttamente alla pagina, accedi ad Amazon CodeCatalyst Spaces nella Console di gestione AWS pagina <https://console.aws.amazon.com/codecatalyst/> iniziale/.

Il token di verifica viene inserito automaticamente nel token di verifica. Un messaggio di successo mostra un messaggio che indica che il token è un token valido.

6. (Facoltativo) In Livelli a pagamento autorizzati, scegli Autorizza livelli a pagamento (Standard, Enterprise) per attivare i livelli a pagamento per il tuo account di fatturazione.

Note

Ciò non comporta l'upgrade del livello di fatturazione a un livello a pagamento.

Tuttavia, questo lo configura Account AWS in modo che tu possa modificare il livello di

fatturazione per il tuo spazio in qualsiasi momento. CodeCatalyst Puoi attivare i livelli a pagamento in qualsiasi momento. Senza apportare questa modifica, lo spazio è in grado di utilizzare solo il livello gratuito.

7. Scegli Verifica spazio.

Viene visualizzato un messaggio di verifica dell'account che indica che l'account è stato aggiunto allo spazio.

Passaggio 3: verifica di una connessione approvata

Dopo aver ottenuto l'approvazione di una connessione, puoi visualizzare la connessione nella console, insieme ai ruoli IAM che le hai aggiunto.

Per esaminare una connessione approvata

1. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
2. La connessione all'account è elencata con la data di creazione.
3. Scegli il nome visualizzato dell'account. Viene visualizzata la pagina dei Account AWS dettagli.

Passaggio 4: aggiungi i ruoli IAM alla tua connessione

Se utilizzi un ruolo IAM configurato per un'azione di CodeCatalyst distribuzione, aggiungi il ruolo al tuo ambiente di distribuzione. Per ulteriori informazioni, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

Prossimi passaggi: crea ruoli IAM aggiuntivi per la connessione del tuo account

Dopo aver creato una connessione, puoi creare ruoli IAM aggiuntivi da aggiungere. I ruoli IAM che aggiungi dipendono dai tuoi flussi di lavoro. Ad esempio, un'azione di CodeCatalyst compilazione richiede il ruolo CodeCatalyst build.

Per connettere il tuo account, avrai bisogno dell'Amazon Resource Name (ARN) per i ruoli che hai creato. Copia l'ARN per il tuo ruolo o i tuoi ruoli come descritto qui. Per ulteriori informazioni su come lavorare con ARNs i ruoli IAM, consulta [Amazon Resource Name \(ARN\)](#).

Per accedere al tuo ruolo IAM (ARN)

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Nella casella di ricerca, inserisci il nome del ruolo che desideri aggiungere.
4. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

5. In alto, copia il valore Role ARN.

Aggiungere ruoli IAM alle connessioni degli account

Parte della creazione della connessione all'account include l'aggiunta del ruolo o dei ruoli IAM che desideri utilizzare con i progetti del tuo CodeCatalyst spazio.

Note

Per utilizzare i ruoli IAM con una connessione all'account, assicurati che la policy di fiducia sia aggiornata per utilizzare il CodeCatalyst service principal.

Aggiungi i ruoli IAM alla connessione di un account (console)

1. Nel Console di gestione AWS, assicurati di aver effettuato l'accesso con lo stesso account che desideri gestire.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
4. Scegli il nome CodeCatalyst visualizzato su Amazon della connessione del tuo account, quindi scegli Gestisci ruoli da Console di gestione AWS.

Viene visualizzata la pagina Aggiungi ruolo IAM CodeCatalyst allo spazio Amazon.

5. Esegui una delle seguenti operazioni:
 - Per creare un ruolo di servizio che contenga la politica di autorizzazione e la politica di fiducia per il ruolo di sviluppatore, scegli Crea il ruolo di amministratore CodeCatalyst dello sviluppo in IAM. Il ruolo avrà un nome `CodeCatalystWorkflowDevelopmentRole-spaceName` con un identificatore univoco

aggiunto. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere.

[Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName*del servizio](#)

Scegli Crea ruolo di sviluppo.

- Per aggiungere un ruolo che hai già creato in IAM, scegli Aggiungi un ruolo IAM esistente. In Seleziona il ruolo IAM esistente, scegli il ruolo dall'elenco a discesa.

Scegliere Add role (Aggiungi ruolo).

La pagina si apre in. Console di gestione AWS Potrebbe essere necessario effettuare il login per accedere alla pagina.

6. Nel riquadro di navigazione della pagina di Amazon CodeCatalyst Spaces, scegli Spaces.

Per accedere direttamente alla pagina, accedi ad Amazon CodeCatalyst Spaces nella Console di gestione AWS pagina <https://console.aws.amazon.com/codecatalyst/> iniziale/.

7. Scegli l'account aggiunto per il tuo CodeCatalyst spazio. Viene visualizzata la pagina di connessione.
8. Nella pagina di connessione, in Ruoli IAM disponibili per CodeCatalyst, visualizza l'elenco dei ruoli IAM aggiunti al tuo account. Scegli Associa il ruolo IAM a CodeCatalyst.
9. Nel pop-up Associa un ruolo IAM, in Role ARN, inserisci l'Amazon Resource Name (ARN) del ruolo IAM che desideri associare al tuo spazio. CodeCatalyst

In Scopo, scegli uno scopo del ruolo che descriva come desideri utilizzare il ruolo nella connessione del tuo account. Specificate RUNNER i ruoli da utilizzare per eseguire azioni nei flussi di lavoro. Specificate SERVICE i ruoli da utilizzare per accedere a un altro servizio.

È possibile specificare più di uno scopo.

Note

È necessario scegliere uno scopo per il ruolo ARN.

10. Scegli Associa un ruolo IAM. Ripeti questi passaggi per ruoli IAM aggiuntivi.

Aggiungere la connessione all'account e i ruoli IAM all'ambiente di distribuzione

Per accedere a AWS risorse, come Amazon ECS o AWS Lambda risorse per le distribuzioni, le azioni di CodeCatalyst compilazione e distribuzione richiedono ruoli IAM con autorizzazioni per accedere a tali risorse. Con il ruolo di amministratore di Space o Power user, puoi connettere il tuo CodeCatalyst account al Account AWS luogo in cui vengono create le tue risorse. Quindi aggiungi il ruolo IAM alla connessione del tuo account. Per le azioni di implementazione, devi quindi aggiungere il ruolo IAM a un CodeCatalyst ambiente.

Devi aggiungere i ruoli IAM che desideri utilizzare con gli ambienti di distribuzione nei tuoi progetti. L'aggiunta dei ruoli alla connessione dell'account non aggiunge i ruoli e la connessione agli ambienti di distribuzione del progetto. Per aggiungere la connessione all'account e i ruoli IAM all'ambiente di distribuzione, assicurati che la connessione all'account e i ruoli siano creati come descritto in.

[Passaggio 4: aggiungi i ruoli IAM alla tua connessione](#)

Quindi, utilizza la pagina Ambienti nella CodeCatalyst console per aggiungere la connessione all'account e il ruolo IAM a un ambiente di distribuzione in un progetto.

Note

Aggiungi un ruolo IAM a un ambiente solo se il ruolo IAM viene utilizzato per un' CodeCatalyst azione che richiede un ruolo IAM. Tutte le azioni del flusso di lavoro che richiedono ruoli IAM, incluse le azioni di compilazione, devono utilizzare un CodeCatalyst ambiente.

Per aggiungere la connessione all'account e i ruoli IAM all'ambiente di distribuzione

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa al progetto con l'ambiente di distribuzione in cui desideri aggiungere la connessione all'account e i ruoli IAM.
3. Espandi CI/CD, quindi scegli Ambienti.
4. Scegli il tuo ambiente, quindi verranno visualizzate le schede aggiuntive.
5. Scegli la scheda delle Account AWS connessioni. In Nome della connessione, vengono elencati gli account che sono stati aggiunti all'ambiente, se presenti.

6. Selezionare Associate (Associa) Account AWS. <environment_name>Viene visualizzata la pagina Account AWS Associa a.
7. In Connessione, scegli il nome della connessione all'account con i ruoli IAM che desideri aggiungere. Selezionare Associate (Associa).

Visualizzazione delle connessioni degli account

È possibile visualizzare un elenco delle connessioni e visualizzare i dettagli relativi a ciascuna connessione.

È necessario disporre del ruolo di amministratore dello spazio o Power user per gestire le connessioni per il proprio spazio.

Per visualizzare tutte le connessioni relative a uno CodeCatalyst spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai allo spazio con la connessione all'account che desideri visualizzare.
3. Scegli la scheda AWS Account.
4. In AWS Account, visualizza l'elenco delle connessioni degli account per lo spazio, inclusi l'ID dell'account e lo stato di ogni connessione.

Per visualizzare i dettagli della connessione all'account

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
3. In Amazon CodeCatalyst Display name, scegli il nome della connessione. Nella pagina Dettagli, visualizza l'elenco dei ruoli IAM associati alla connessione insieme ad altri dettagli.

Eliminazione delle connessioni all'account (in CodeCatalyst)

Puoi eliminare una connessione a un account che non ti serve più. Per questa procedura, utilizzerai CodeCatalyst per eliminare una connessione all'account che hai precedentemente aggiunto al tuo spazio. Ciò elimina la connessione all'account dal tuo spazio, a condizione che l'account non sia l'account di fatturazione dello spazio.

 Important

Dopo l'eliminazione di una connessione all'account, non è possibile ricollegarla. È necessario creare una nuova connessione all'account e quindi associare i ruoli e gli ambienti IAM o impostare la fatturazione, in base alle esigenze.

È necessario designare un account di fatturazione per il tuo CodeCatalyst spazio, anche se l'utilizzo dello spazio non supererà il livello gratuito. Prima di poter rimuovere uno spazio per un account che è un account di fatturazione designato, dovrai aggiungere un altro account per lo spazio. Consulta [la sezione Gestione della fatturazione](#) nella Amazon CodeCatalyst Administrator Guide.

 Important

Sebbene sia possibile utilizzare questi passaggi per rimuovere un account, questa operazione non è consigliata. L'account potrebbe anche essere configurato per supportare i flussi di lavoro in CodeCatalyst.

Per gestire le connessioni degli account per il tuo spazio, devi avere il ruolo di amministratore dello spazio o Power user.

Un account che è stato rimosso può essere aggiunto nuovamente in un secondo momento, ma è necessario creare una nuova connessione tra l'account e lo spazio. Dovrai riassociare tutti i ruoli IAM all'account aggiunto.

Per eliminare una connessione all'account

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
3. In Amazon CodeCatalyst Display name, scegli il selettore accanto alla connessione dell'account che desideri rimuovere.
4. Scegli Rimuovi Account AWS. Conferma l'eliminazione inserendo il nome nel campo, quindi scegli Rimuovi.

Viene visualizzato un banner di successo e la connessione all'account viene rimossa dall'elenco delle connessioni.

Configurazione di un account di fatturazione per uno spazio

È necessario designare un account di fatturazione per CodeCatalyst lo spazio, anche se l'utilizzo dello spazio non supererà il livello gratuito.

Per configurare un account di fatturazione, consulta [Fatturazione nella Guida](#) per l'CodeCatalyst amministratore. L' Account AWS account specificato come account di fatturazione per il tuo CodeCatalyst spazio ha quote diverse rispetto alle connessioni di altri account per lo spazio. Per ulteriori informazioni, consulta [Quote per CodeCatalyst](#).

Per rimuovere un account che è un account di fatturazione designato per il tuo CodeCatalyst spazio, assicurati di specificare prima un nuovo account di fatturazione.

Configurazione dei ruoli IAM per gli account connessi

Crei ruoli in AWS Identity and Access Management (IAM) per l'account a cui desideri aggiungerli CodeCatalyst. Se stai aggiungendo un account di fatturazione, non è necessario creare ruoli.

Nel tuo Account AWS, devi disporre delle autorizzazioni per creare i ruoli Account AWS che desideri aggiungere al tuo spazio. Per ulteriori informazioni sui ruoli e le policy IAM, inclusi riferimenti IAM e policy di esempio, consulta [Identity and Access Management e Amazon CodeCatalyst](#). Per ulteriori informazioni sulla politica di fiducia e sui principali di servizio utilizzati in CodeCatalyst, consulta [Comprendere il modello CodeCatalyst di fiducia](#).

In CodeCatalyst, devi aver effettuato l'accesso con il ruolo di amministratore dello spazio per completare la procedura di aggiunta degli account (e dei ruoli, se applicabile) al tuo spazio.

Puoi aggiungere ruoli alle connessioni del tuo account utilizzando uno dei seguenti metodi.

- Per creare un ruolo di servizio che contenga la politica di autorizzazione e la politica di attendibilità per il CodeCatalystWorkflowDevelopmentRole-**spaceName**ruolo, consulta [Ruolo CodeCatalystWorkflowDevelopmentRole-**spaceName**](#).
- Per un esempio di creazione di un ruolo e aggiunta di una politica per creare un progetto a partire da un blueprint, vedere. [Creazione di un ruolo IAM e utilizzo della policy CodeCatalyst di fiducia](#)
- Per un elenco di esempi di policy relative ai ruoli da utilizzare durante la creazione dei ruoli IAM, consulta [Concedi l'accesso alle AWS risorse del progetto con i ruoli IAM](#).
- Per i passaggi dettagliati per creare ruoli per le azioni del flusso di lavoro, consulta il tutorial sul flusso di lavoro relativo a tale azione come segue:

- [Tutorial: caricare artefatti su Amazon S3](#)
- [Tutorial: Implementazione di un'applicazione serverless](#)
- [Tutorial: distribuire un'applicazione su Amazon ECS](#)
- [Tutorial: codice Lint usando un' GitHubazione](#)

Argomenti

- [Ruolo CodeCatalystWorkflowDevelopmentRole-spaceName](#)
- [Ruolo AWSRoleForCodeCatalystSupport](#)
- [Creazione di un ruolo IAM e utilizzo della policy CodeCatalyst di fiducia](#)

Ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName*

Il ruolo di sviluppatore viene creato con un solo clic in IAM. Devi avere il ruolo di amministratore dello spazio o Power user nello spazio in cui desideri aggiungere l'account. È inoltre necessario disporre delle autorizzazioni amministrative per l' Account AWS utente che si desidera aggiungere.

Prima di iniziare la procedura seguente, devi accedere Console di gestione AWS con lo stesso account che desideri aggiungere al tuo CodeCatalyst spazio. In caso contrario, la console restituirà un errore sconosciuto dell'account.

Per creare e aggiungere CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Prima di iniziare a utilizzare la CodeCatalyst console, apri il Console di gestione AWS, quindi assicurati di aver effettuato l'accesso con lo stesso Account AWS nome del tuo spazio.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
4. Scegli il link relativo alla Account AWS posizione in cui desideri creare il ruolo. Viene visualizzata la pagina dei Account AWS dettagli.
5. Scegli Gestisci ruoli da Console di gestione AWS.

La pagina Aggiungi il ruolo IAM CodeCatalyst allo spazio Amazon si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

- Scegli Crea il ruolo di amministratore CodeCatalyst dello sviluppo in IAM. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di fiducia per il ruolo di sviluppo. Il ruolo avrà un nome `CodeCatalystWorkflowDevelopmentRole-spaceName`. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere [Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName* del servizio](#).

Note

Questo ruolo è consigliato solo per gli account sviluppatore e utilizza la politica `AdministratorAccess` AWS gestita, che gli consente l'accesso completo alla creazione di nuove politiche e risorse Account AWS.

- Scegli Crea ruolo di sviluppo.
- Nella pagina delle connessioni, in Ruoli IAM disponibili per CodeCatalyst, visualizza il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo nell'elenco dei ruoli IAM aggiunti al tuo account.
- Per tornare al tuo spazio, scegli Vai su Amazon CodeCatalyst.

Ruolo `AWSRoleForCodeCatalystSupport`

Crei il ruolo di supporto come ruolo con un clic in IAM. Devi avere il ruolo di amministratore dello spazio o Power user nello spazio in cui desideri aggiungere l'account. È inoltre necessario disporre delle autorizzazioni amministrative per l'Account AWS utente che si desidera aggiungere.

Prima di iniziare la procedura seguente, devi accedere Console di gestione AWS con lo stesso account che desideri aggiungere al tuo CodeCatalyst spazio. In caso contrario, la console restituirà un errore sconosciuto dell'account.

Per creare e aggiungere CodeCatalyst `AWSRoleForCodeCatalystSupport`

- Prima di iniziare a utilizzare la CodeCatalyst console, apri il Console di gestione AWS, quindi assicurati di aver effettuato l'accesso con lo stesso Account AWS nome del tuo spazio.
- Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
- Scegli il link relativo alla Account AWS posizione in cui desideri creare il ruolo. Viene visualizzata la pagina dei Account AWS dettagli.
- Scegli Gestisci ruoli da Console di gestione AWS.

La pagina Aggiungi il ruolo IAM CodeCatalyst allo spazio Amazon si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

5. In Dettagli CodeCatalyst dello spazio, scegli Aggiungi ruolo CodeCatalyst Support. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di attendibilità per il ruolo di sviluppo in anteprima. Il ruolo avrà un nome `AWSRoleForCodeCatalystSupport` con un identificatore univoco aggiunto. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere. [Comprensione del ruolo AWSRoleForCodeCatalystSupport del servizio](#)
6. Nella pagina Aggiungi ruolo per CodeCatalyst Support, lascia selezionata l'impostazione predefinita, quindi scegli Crea ruolo.
7. In Ruoli IAM disponibili per CodeCatalyst, visualizza il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo nell'elenco dei ruoli IAM aggiunti al tuo account.
8. Per tornare al tuo spazio, scegli Vai su Amazon CodeCatalyst.

Creazione di un ruolo IAM e utilizzo della policy CodeCatalyst di fiducia

I ruoli IAM da utilizzare CodeCatalyst con Account AWS le connessioni devono essere configurati per utilizzare la policy di fiducia fornita qui. Utilizza questi passaggi per creare un ruolo IAM e allegare una policy che ti permetta di creare progetti a partire da blueprint in CodeCatalyst.

In alternativa, puoi creare un ruolo di servizio che contenga la politica di autorizzazione e la politica di fiducia per il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo. Per ulteriori informazioni, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Selezionare Roles (Ruoli), quindi selezionare Create role (Crea ruolo).
3. Scegli una politica di fiducia personalizzata.
4. Nel modulo Politica di fiducia personalizzata, incolla la seguente politica di fiducia.

```
"Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iam:CreateRole",  
      "Resource": "*"
```

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "codecatalyst-runner.amazonaws.com",
      "codecatalyst.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:codecatalyst::space/spaceId/project/*"
    }
  }
}
```

5. Scegli Next (Successivo).
6. In Aggiungi autorizzazioni, cerca e seleziona una policy personalizzata che hai già creato in IAM.
7. Scegli Next (Successivo).
8. Per Nome ruolo, inserisci un nome per il ruolo, ad esempio: `codecatalyst-project-role`
9. Scegli Crea ruolo.
10. Copia il ruolo Amazon Resource Name (ARN). Dovrai fornire queste informazioni quando aggiungi il ruolo alla connessione o all'ambiente del tuo account.

Concessione delle autorizzazioni di spazio agli utenti

Puoi gestire i membri di uno spazio visualizzando, aggiungendo, rimuovendo o modificando i ruoli degli utenti che si iscrivono allo spazio.

Le informazioni contenute in questa guida vengono fornite per invitare e gestire gli utenti negli spazi CodeCatalyst che supportano gli utenti AWS Builder ID. Per ulteriori informazioni sui passaggi per configurare e amministrare uno spazio che supporti la federazione delle identità, consulta [Configurazione e amministrazione CodeCatalyst degli spazi](#) nella Amazon CodeCatalyst Administrator Guide.

Visualizzazione dei membri in uno spazio

Puoi visualizzare gli utenti nel tuo spazio, comprese le informazioni sui loro nomi visualizzati, sugli alias e sul ruolo che hanno nello spazio. Esistono tre ruoli per i membri in uno spazio:

- **Amministratore dello spazio:** questo ruolo dispone di tutte le autorizzazioni CodeCatalyst, inclusa la creazione di progetti. Assegna questo ruolo solo agli utenti che devono amministrare ogni aspetto di uno spazio, ad esempio accedere a tutti i progetti nello spazio.

Non è possibile modificare questo ruolo in un secondo momento senza prima rimuovere l'utente. Per ulteriori informazioni, consulta [Ruolo di amministratore dello spazio](#).

- **Power user:** questo ruolo è il secondo ruolo più importante CodeCatalyst negli spazi Amazon, ma non ha accesso ai progetti in uno spazio. È progettato per gli utenti che devono essere in grado di creare progetti in uno spazio e aiutare a gestire gli utenti e le risorse dello spazio. Per ulteriori informazioni, consulta [Ruolo di Power User](#).
- **Accesso limitato:** questo ruolo viene assegnato per impostazione predefinita agli utenti che si iscrivono allo spazio accettando inviti ai progetti presenti nello spazio. Ai membri del progetto viene assegnato un ruolo in un progetto. Per informazioni sulla gestione dei membri del progetto, vedere [Concessione delle autorizzazioni di progetto agli utenti](#).

La tabella degli amministratori dello spazio mostra gli utenti con il ruolo di amministratore dello spazio. Questi utenti non vengono visualizzati nei membri dello spazio perché vengono assegnati automaticamente (implicitamente) a tutti i progetti nello spazio e non hanno un ruolo in un progetto.

La tabella dei membri dello spazio mostra tutti i membri dello spazio che hanno un ruolo in un progetto ma non hanno il ruolo di amministratore dello spazio.

Gli utenti vengono visualizzati in base al ruolo di amministratore dello spazio nel modo CodeCatalyst seguente:

- Un utente con il ruolo di amministratore dello spazio che successivamente accetta un invito e un ruolo al progetto non verrà visualizzato nella tabella Membri dello spazio negli spazi o nella tabella Membri del progetto sotto i progetti. Continueranno a essere visualizzati nella tabella degli amministratori dello spazio in entrambe le posizioni. In ogni progetto, tutti gli utenti con il ruolo di amministratore dello spazio vengono visualizzati nella tabella degli amministratori dello spazio del progetto relativa a quel progetto.
- Un utente che accetta un invito a partecipare al progetto con un ruolo di progetto viene aggiunto allo spazio con il ruolo di accesso limitato. Se il ruolo dell'utente passa successivamente al ruolo

di amministratore dello spazio, ma passerà anche dalla tabella Membri dello spazio alla tabella Amministratori dello spazio. Nell'ambito del progetto, l'utente passerà dalla tabella Membri del progetto alla tabella degli amministratori dello spazio.

Per visualizzare gli utenti e i ruoli nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Membri.

Gli utenti che sono membri dello spazio vengono visualizzati nella tabella Membri dello spazio.

 Tip

Se hai il ruolo di amministratore dello spazio, puoi visualizzare a quali progetti sei stato invitato direttamente. Vai alle impostazioni del progetto, quindi scegli I miei progetti.

Nella colonna Stato, i seguenti sono valori validi:

- CodeCatalyst Invitato: ha inviato l'invito ma l'utente non l'ha ancora accettato o rifiutato.
- Membro: l'utente ha accettato l'invito.

Invitare un utente direttamente in uno spazio

Puoi invitare gli utenti direttamente nel tuo CodeCatalyst spazio. Ciò è utile quando desideri invitare quell'utente ad aiutarti a gestire lo spazio assegnandogli il ruolo di amministratore dello spazio o Power user. L'assegnazione di uno di questi ruoli ad altri utenti può aiutarti a distribuire le responsabilità della gestione dello spazio tra più persone senza dover invitare questi utenti a nessun progetto.

 Note

È necessario disporre del ruolo di amministratore dello spazio o Power user per invitare i membri.

La tabella degli amministratori dello spazio mostra gli utenti con il ruolo di amministratore dello spazio. Questi utenti non vengono visualizzati nella tabella dei membri dello spazio perché vengono assegnati automaticamente (implicitamente) a tutti i progetti nello spazio e non hanno un ruolo in un progetto.

I membri che accettano un invito al progetto vengono aggiunti allo spazio per impostazione predefinita. La tabella Membri del progetto mostra tutti i membri dello spazio che hanno un ruolo in un progetto.

Per ulteriori informazioni su come accettare un invito e accedere per la prima volta, consulta [Configura e accedi a CodeCatalyst](#).

Per invitare un utente nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio.
3. Scegli Impostazioni, quindi scegli Membri.
4. Seleziona Invite (Invita).
5. Inserisci l'email della persona che desideri invitare a far parte del tuo spazio. In Ruolo, scegli il ruolo che desideri assegnare a quell'utente nello spazio.
6. Scegli Invita

Annullamento di un invito per uno spazio

Se desideri annullare un invito a entrare in uno spazio che hai inviato di recente e che non è ancora stato accettato, puoi annullarlo.

Per gestire gli inviti agli spazi, devi avere il ruolo di amministratore dello spazio o Power user.

Per annullare l'invito di un membro dello spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Vai al tuo spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Membri.
4. Verifica che il membro abbia lo stato di Invitato.

 Note

Puoi annullare solo un invito che non è stato ancora accettato.

5. Scegli l'opzione accanto alla riga con il membro invitato, quindi scegli Annulla invito.
6. Viene visualizzata una finestra di conferma. Scegli Annulla invito per confermare.

Modifica del ruolo di un membro dello spazio

Puoi modificare il ruolo assegnato a un membro del tuo spazio. È necessario disporre del ruolo di amministratore dello spazio per modificare il ruolo di un utente nello spazio.

La tabella degli amministratori dello spazio mostra gli utenti con il ruolo di amministratore dello spazio. Questi utenti non vengono visualizzati nella tabella dei membri dello spazio perché vengono assegnati automaticamente (implicitamente) a tutti i progetti nello spazio.

Per modificare il ruolo di un utente nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Membri.
4. Nella tabella Membri dello spazio, scegli l'utente di cui desideri modificare il ruolo. Scegli Cambia ruolo.

Rimuovere un membro dello spazio

Puoi rimuovere un membro del tuo spazio quando non ha bisogno di accedere a nessuna delle risorse dello spazio. È necessario disporre del ruolo di amministratore dello spazio per rimuovere un membro da uno spazio.

La tabella degli amministratori dello spazio mostra gli utenti con il ruolo di amministratore dello spazio. Questi utenti non vengono visualizzati nella tabella dei membri dello spazio perché vengono assegnati automaticamente (implicitamente) a tutti i progetti nello spazio e non hanno un ruolo in un progetto. Puoi rimuovere direttamente un membro del tuo spazio solo in questa tabella.

Per rimuovere un utente dalla tabella dei membri del progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio.

Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Membri.
4. Scegli l'utente nella tabella Membri del progetto. Scegli Rimuovi.

Note

La rimozione di un membro dallo spazio rimuoverà l'utente da tutti i progetti nello spazio, insieme alle autorizzazioni associate alle risorse di tali progetti.

Rimuovere o modificare il ruolo di un utente con il ruolo di amministratore dello spazio

Puoi rimuovere o modificare il ruolo di un utente con il ruolo di amministratore dello spazio per il tuo spazio.

È necessario disporre del ruolo di amministratore dello spazio per rimuovere un utente con il ruolo di amministratore dello spazio da uno spazio. La modifica del ruolo di un utente con il ruolo di amministratore dello spazio essenzialmente rimuove l'utente dalla tabella degli amministratori dello

spazio. Se quell'utente non ha un ruolo di progetto in nessun progetto dello spazio, la rimozione del ruolo di amministratore dello spazio dall'utente rimuoverà l'utente dallo spazio.

 Note

In qualità di utente con il ruolo di amministratore dello spazio, non puoi rimuovere te stesso. Contatta un altro utente con il ruolo di amministratore dello spazio.

Per rimuovere un utente con il ruolo di amministratore dello spazio dalla tabella dei membri dello spazio

 Note

Un utente che non è stato aggiunto esplicitamente a un progetto, non ha alcun ruolo di progetto (amministratore del progetto o collaboratore). Se il ruolo di amministratore dello spazio è l'unico ruolo dell'utente, l'utente viene rimosso completamente dallo spazio.

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Passa allo spazio in cui desideri rimuovere o modificare il ruolo di un utente con il ruolo di amministratore dello spazio.
3. Scegli Impostazioni, quindi scegli Membri.
4. Visualizza lo stato dell'invito per l'elenco dei membri e assicurati che l'elenco non contenga inviti in sospeso non autorizzati allo spazio (lo stato è Invitato).

 Important

Prima di rimuovere un utente con il ruolo di amministratore dello spazio, devi verificare che non siano stati avviati inviti in sospeso.

5. Scegli la scheda Membri. Nella tabella Amministratori dello spazio, scegli l'utente, quindi scegli Rimuovi.

Nella finestra di dialogo Rimuovi membro, effettuate una delle seguenti operazioni.

- Scegliete l'opzione per rimuovere solo il ruolo di amministratore dello spazio dell'utente. Scegli Rimuovi.

 Important

Se all'utente non è assegnato nessun altro ruolo, la modifica del ruolo da amministratore dello spazio rimuove l'utente dallo spazio.

- Scegliete l'opzione per rimuovere un utente con il ruolo di amministratore dello spazio dallo spazio e da tutti i relativi progetti. Scegli Rimuovi.
6. Aggiorna la scheda Membri. L'utente viene automaticamente aggiunto all'elenco dei membri del progetto in qualsiasi progetto a cui apparteneva tramite i ruoli del progetto. Se il ruolo di amministratore dello spazio era l'unico ruolo dell'utente, l'utente viene rimosso completamente dallo spazio.

Consentire l'accesso allo spazio tramite team

Dopo aver creato uno spazio, puoi aggiungere team. I team consentono di raggruppare gli utenti in modo che possano condividere le autorizzazioni e gestire i progetti, il monitoraggio dei problemi, i ruoli e le risorse. CodeCatalyst

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

I team vengono gestiti anche a project/space livello di CodeCatalyst. Per ulteriori informazioni sui team negli spazi/progetti, consulta. [Consentire l'accesso allo spazio tramite team](#)

Argomenti

- [Creare un team](#)
- [Visualizzazione di un team](#)
- [Assegnazione di ruoli spaziali a un team](#)
- [Assegnazione di ruoli di progetto a un team a livello di spazio](#)
- [Aggiungere direttamente un utente a un team](#)
- [Rimuovere direttamente un utente da un team](#)
- [Aggiungere un gruppo SSO a un team](#)
- [Eliminazione di un team](#)

Creare un team

Un team può disporre di autorizzazioni di ruolo, ad esempio Power user, in uno spazio. Un team può anche disporre di autorizzazioni di progetto, ad esempio Amministratore di progetto, in un progetto. I team possono essere associati a molti progetti con ruoli diversi per ogni progetto. Puoi gestire team in cui i membri del team sono singoli utenti per uno spazio AWS Builder ID o gruppi SSO per uno spazio che supporta la federazione delle identità.

Nella pagina dei membri per gli utenti dello spazio e del progetto, gli utenti possono avere più ruoli. Gli utenti con più ruoli mostreranno un indicatore quando hanno più ruoli e verranno visualizzati per primi con il ruolo con il maggior numero di autorizzazioni.

Note

Se il tuo spazio supporta la federazione delle identità, devi aver già configurato gli utenti o i gruppi SSO in IAM Identity Center.

Il modo in cui gestisci i membri del team dipende da come aggiungerai e rimuoverai utenti. Esistono due opzioni per la gestione dei membri del team:

- Aggiungere utenti direttamente: aggiungi o rimuovi gli utenti singolarmente. Ad esempio, aggiungi utenti a un team scegliendo utenti AWS Builder ID o utenti SSO già configurati in IAM Identity Center. Quando scegli di gestire i membri del team aggiungendo direttamente utenti AWS Builder ID o utenti SSO, l'opzione di utilizzare i gruppi SSO non sarà più disponibile.
- Usa i gruppi SSO: gestisci i membri del team tramite gruppi SSO già configurati in IAM Identity Center. Quando scegli di gestire i membri del team utilizzando i gruppi SSO, l'opzione per aggiungere utenti direttamente non sarà più disponibile.

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Per creare un team

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. Scegli Crea team.
4. In Nome del team, inserisci un nome descrittivo per il team.

 Note

Il nome del team deve essere univoco nel tuo spazio.

(Facoltativo) In Descrizione del team, inserisci una descrizione per il team.

5. In Ruolo spaziale, scegli un ruolo dall'elenco dei ruoli spaziali disponibili CodeCatalyst che desideri assegnare al team. Il ruolo verrà ereditato da tutti i membri del team.
 - Amministratore dello spazio: per i dettagli, vedere [Ruolo di amministratore dello spazio](#).
 - Accesso limitato: per i dettagli, vedere [Ruolo ad accesso limitato](#).
 - Power user - Per i dettagli, consulta [Ruolo di Power User](#).
6. In Appartenenza al team, scegli una delle seguenti opzioni per scegliere il metodo di aggiunta di membri al team.
 - Scegli Aggiungi membri direttamente per gestire gli utenti individualmente. Ciò include l'aggiunta di utenti AWS Builder ID per uno spazio o l'aggiunta di utenti SSO per uno spazio che supporta la federazione delle identità.
 - Scegli Usa gruppi SSO per scegliere i gruppi SSO che hai già configurato in IAM Identity Center.

In Gruppi SSO, seleziona la casella accanto ai gruppi che desideri aggiungere. Puoi aggiungere fino a cinque gruppi SSO.

 Note

Non è possibile modificarlo in un secondo momento. Quando scegli di gestire i membri del team aggiungendo direttamente utenti AWS Builder ID o utenti SSO, l'opzione per utilizzare i gruppi SSO non sarà più disponibile. Quando scegli di gestire i membri del team utilizzando i gruppi SSO, l'opzione per aggiungere utenti direttamente non sarà più disponibile.

7. Scegli Create (Crea).

 Note

Quando scegli di utilizzare i gruppi SSO, tieni presente che gli utenti del gruppo SSO non vengono richiamati al momento della creazione del team. Gli utenti dovranno aver effettuato l'accesso CodeCatalyst prima di essere visibili nell'elenco.

Visualizzazione di un team

In CodeCatalyst, puoi visualizzare i progetti e i ruoli del tuo team. Nella pagina dei membri, puoi visualizzare i ruoli del progetto e un elenco di utenti. Per i team di tipo gruppo SSO, potrai anche visualizzare un elenco di gruppi SSO associati al team.

Per visualizzare un team

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. Nel ruolo Space, visualizza il ruolo assegnato al team per questo spazio.
4. Nella scheda Ruoli del progetto, visualizza il progetto e il ruolo di progetto assegnati al team per ogni CodeCatalyst progetto nello spazio in cui il team è stato aggiunto come membro (solo per uno spazio AWS Builder ID).
5. Nella scheda Membri, visualizza l'elenco dei membri assegnati al team.
6. Nella scheda Gruppi SSO, visualizza l'elenco dei gruppi SSO assegnati al team (per uno spazio che supporta solo la federazione delle identità).

Assegnazione di ruoli spaziali a un team

I team sono un modo per raggruppare gli utenti in modo da poter concedere e gestire l'accesso del team ai progetti in CodeCatalyst. Ad esempio, puoi utilizzare i team per gestire rapidamente i ruoli e le autorizzazioni degli utenti dando a un team la possibilità di gestire uno spazio per gli utenti.

Un team può disporre di autorizzazioni di ruolo, ad esempio Power user, in uno spazio. Puoi modificare il ruolo spaziale di un team, ma tieni presente che tutti i membri del team ereditano tali autorizzazioni.

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Modifica del ruolo spaziale di un team

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. In Azioni, scegli Cambia ruolo spaziale. Puoi modificare il ruolo dello spazio in uno dei seguenti. Questo modifica il ruolo di tutti i membri del team.
 - Amministratore dello spazio: per i dettagli, vedere [Ruolo di amministratore dello spazio](#).
 - Accesso limitato: per i dettagli, vedere [Ruolo ad accesso limitato](#).
 - Power user - Per i dettagli, consulta [Ruolo di Power User](#).
4. Scegli Save (Salva).

Assegnazione di ruoli di progetto a un team a livello di spazio

Un team in CodeCatalyst è simile a un utente in quanto i membri del team possono avere autorizzazioni di ruolo, ad esempio Amministratore di progetto, in un progetto. Al team verrà applicato un cambio di ruolo e tutti i membri del team ereditano tali autorizzazioni. Puoi scegliere un ruolo per ogni progetto che verrà assegnato automaticamente al team.

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Per aggiungere o modificare un ruolo nel progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. Scegli la scheda Ruoli del progetto.
4. Per modificare un ruolo, scegli il selettore accanto al progetto in questo elenco, quindi scegli Cambia ruolo. Per aggiungere un ruolo, scegli Aggiungi ruolo di progetto. In Project, scegli il progetto che desideri aggiungere e in Ruolo scegli il ruolo. Scegli uno dei ruoli di progetto disponibili:
 - Amministratore del progetto: per i dettagli, consulta [Ruolo di amministratore del progetto](#).
 - Collaboratore - Per i dettagli, consulta [Ruolo di collaboratore](#).
 - Revisore - Per i dettagli, consulta. [Ruolo di revisore](#)
 - Sola lettura - Per i dettagli, vedi [Ruolo di sola lettura](#).
5. Scegli Save (Salva).

Per rimuovere un ruolo nel progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. Scegli la scheda Ruoli del progetto.
4. Scegli il ruolo che desideri rimuovere.

Important

La rimozione di un ruolo da un team rimuove le autorizzazioni associate per tutti gli utenti del team.

5. Scegli Save (Salva).

Aggiungere direttamente un utente a un team

Puoi aggiungere membri al tuo team. Quando aggiungi un utente, il nuovo utente erediterà le autorizzazioni da tutti i ruoli esistenti nel team.

Indipendentemente dal fatto che il tuo spazio sia configurato per il supporto agli utenti di AWS Builder ID o per la federazione delle identità, puoi configurare lo spazio per aggiungere utenti direttamente.

Note

Quando lo spazio è configurato per gestire i membri del team utilizzando i gruppi SSO, l'opzione di utilizzare Aggiungi utenti direttamente non è disponibile. Per utilizzare i gruppi SSO, vedi. [Aggiungere un gruppo SSO a un team](#)

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Per aggiungere direttamente un utente

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. Scegli la scheda Membri.
4. Scegli Aggiungi membro.

 Note

Gli utenti aggiunti a un team devono essere già membri di uno spazio. Non puoi aggiungere o invitare un membro del team che non è membro dello spazio.

5. Scegli un utente nel campo a discesa, quindi scegli Salva. Scegli utenti AWS Builder ID o utenti SSO già configurati in IAM Identity Center.

Rimuovere direttamente un utente da un team

Puoi rimuovere membri del team dal tuo team. Tutte le autorizzazioni non verranno più ereditate dall'utente. Puoi aggiungere nuovamente l'utente al team in un secondo momento.

 Note

Quando rimuovi un membro del team, le autorizzazioni associate all'utente verranno rimosse da tutti i progetti e le risorse dello spazio.

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Per rimuovere un membro del team

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. Scegli la scheda Membri.
4. Scegli il selettore accanto all'utente che desideri rimuovere, quindi scegli Rimuovi.
5. Immettete remove nel campo di immissione, quindi scegliete Rimuovi.

Aggiungere un gruppo SSO a un team

Se il tuo spazio è configurato come spazio con utenti e gruppi SSO gestiti in IAM Identity Center, puoi aggiungere un gruppo SSO che si unirà allo spazio come team separato.

 Note

Quando scegli di gestire i membri del team aggiungendo direttamente utenti AWS Builder ID o utenti SSO, l'opzione per utilizzare i gruppi SSO non è disponibile. Per aggiungere utenti direttamente, consulta. [Aggiungere direttamente un utente a un team](#)

È necessario avere il ruolo di amministratore dello spazio per gestire i team.

Per aggiungere un gruppo SSO come team

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella pagina del tuo spazio, scegli Teams. Scegli la scheda Gruppi SSO.
3. Scegli i gruppi SSO che desideri aggiungere. Puoi aggiungere fino a cinque gruppi SSO.

Eliminazione di un team

Puoi eliminare un team che non ti serve più.

 Note

Quando elimini un team, le autorizzazioni associate verranno rimosse per tutti i membri del team da tutti i progetti e le risorse dello spazio.

Devi avere il ruolo di amministratore dello spazio per gestire i team.

Eliminare un team

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo spazio. Scegli Impostazioni, quindi scegli Teams.
3. In Azioni, scegli Elimina team. Questo cambia il ruolo dell'intero team.
4. Scegli Delete (Elimina).

Consentire l'accesso allo spazio per le risorse della macchina

Le risorse della macchina sono risorse specifiche a CodeCatalyst cui vengono concesse le autorizzazioni per progetti o spazi in CodeCatalyst.

Note

Il termine risorsa macchina non si riferisce a un'infrastruttura cloud come un' EC2 istanza Amazon, ma si riferisce invece a un blueprint o a una risorsa di flusso di lavoro con autorizzazioni per uno spazio o un progetto.

Una risorsa macchina rappresenta l'identità dell'utente rispetto alla risorsa autorizzata quando si accede CodeCatalyst tramite SSO. Le risorse della macchina vengono utilizzate per concedere autorizzazioni alle risorse presenti nello spazio, come progetti e flussi di lavoro. È possibile visualizzare le risorse della macchina nel proprio spazio e scegliere di abilitare o disabilitare le risorse della macchina per il proprio spazio. Ad esempio, potresti voler disabilitare una risorsa della macchina per gestire l'accesso e riattivarla in un secondo momento.

Queste operazioni sono disponibili per le risorse della macchina nei casi in cui una risorsa della macchina debba essere revocata o disabilitata. Ad esempio, se sospetti che le credenziali possano essere state compromesse, puoi disabilitare la risorsa della macchina. In genere, non è necessario utilizzare queste operazioni.

È necessario disporre del ruolo di amministratore dello spazio per visualizzare questa pagina e gestire le risorse del computer a livello di spazio.

Le risorse della macchina vengono gestite anche a livello di progetto in CodeCatalyst. Per ulteriori informazioni sui team coinvolti nei progetti, consulta [Consentire l'accesso allo spazio per le risorse della macchina](#).

Argomenti

- [Accesso allo spazio di visualizzazione per le risorse della macchina](#)
- [Disabilitazione dell'accesso allo spazio per le risorse della macchina](#)
- [Consentire l'accesso allo spazio per le risorse della macchina](#)

Accesso allo spazio di visualizzazione per le risorse della macchina

È possibile visualizzare un elenco delle risorse del computer utilizzate nel proprio spazio.

È necessario disporre del ruolo di amministratore dello spazio per gestire le risorse della macchina.

Per visualizzare le risorse della macchina

1. Aprire la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo spazio, quindi scegli Impostazioni. Scegli Risorse della macchina.
3. Nel menu a discesa, scegli l'azione Workflow per visualizzare solo le risorse della macchina per i flussi di lavoro. Scegli Blueprint per visualizzare solo le risorse della macchina per i blueprint.

Puoi anche filtrare in base a un nome utilizzando il campo Filtro.

Disabilitazione dell'accesso allo spazio per le risorse della macchina

È possibile scegliere di disabilitare le risorse della macchina in uso nel proprio spazio.

Important

La disabilitazione delle risorse della macchina rimuoverà tutte le autorizzazioni per tutti i blueprint o i flussi di lavoro associati nello spazio.

È necessario disporre del ruolo di amministratore dello spazio per gestire le risorse della macchina.

Per disabilitare le risorse della macchina

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo spazio, quindi scegli Impostazioni. Scegli Risorse della macchina.
3. Scegliere una delle seguenti opzioni.

Important

La disabilitazione delle risorse della macchina rimuoverà tutte le autorizzazioni per tutti i blueprint o i flussi di lavoro associati nello spazio.

- Per disabilitarle singolarmente, scegli il selettore accanto a una o più risorse della macchina che desideri disabilitare. Scegli Disabilita, quindi scegli Questa risorsa.
- Per disattivare tutte le risorse, scegli Disabilita, quindi scegli Tutte le risorse.
- Per disabilitare tutte le azioni del flusso di lavoro, scegli Disabilita, quindi scegli Tutte le azioni del flusso di lavoro.
- Per disabilitare tutti i blueprint, scegli Disabilita, quindi scegli Tutti i blueprint.

Consentire l'accesso allo spazio per le risorse della macchina

È possibile scegliere di abilitare le risorse della macchina che sono in uso nel proprio spazio e che sono state disabilitate.

È necessario disporre del ruolo di amministratore dello spazio per gestire le risorse della macchina.

Per abilitare le risorse della macchina

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo spazio, quindi scegli Impostazioni. Scegli Risorse della macchina.
3. Scegliere una delle seguenti opzioni.
 - Per abilitarle singolarmente, scegli il selettore accanto a una o più risorse della macchina che desideri abilitare. Scegli Abilita, quindi scegli Questa risorsa.
 - Per abilitare tutte le risorse, scegli Abilita, quindi scegli Tutte le risorse.
 - Per abilitare tutte le azioni del flusso di lavoro, scegli Abilita, quindi scegli Tutte le azioni del flusso di lavoro.
 - Per abilitare tutti i blueprint, scegli Abilita, quindi scegli Tutti i blueprint.

Amministrazione degli ambienti di sviluppo per uno spazio

Tutti gli ambienti di sviluppo vengono creati come parte di un progetto all'interno di uno spazio. I membri dello spazio possono creare i propri ambienti di sviluppo all'interno di un progetto a livello di repository di origine. Gli amministratori dello spazio possono quindi utilizzare la CodeCatalyst console Amazon per visualizzare, modificare, eliminare e interrompere gli ambienti di sviluppo per conto dei

membri dello spazio. In breve, gli amministratori dello spazio mantengono gli ambienti di sviluppo a livello di spazio.

Considerazioni sull'amministrazione degli ambienti di sviluppo

- È necessario disporre del ruolo di amministratore dello spazio per visualizzare la pagina Ambienti di sviluppo in Impostazioni e per gestire gli ambienti di sviluppo a livello di spazio.
- I membri dello spazio gestiscono gli ambienti di sviluppo che creano nei progetti tramite i propri CodeCatalyst account. Quando amministri Dev Environments come amministratore dello spazio, gestisci queste risorse per conto dei membri dello spazio.
- Gli ambienti di sviluppo utilizzano per impostazione predefinita una configurazione di elaborazione e archiviazione specifica. Per informazioni sulla fatturazione e sulle tariffe per l'aggiornamento della configurazione, consulta la pagina dei prezzi di [Amazon CodeCatalyst](#).

Important

Gli ambienti di sviluppo non sono disponibili per gli utenti negli spazi in cui Active Directory viene utilizzato come provider di identità. Per ulteriori informazioni, consulta [Non riesco a creare un ambiente di sviluppo quando ho effettuato l'accesso CodeCatalyst utilizzando un account Single Sign-on](#).

Per altre considerazioni sugli ambienti di sviluppo, tra cui l'interruzione delle istanze in esecuzione, la configurazione di elaborazione predefinita, l'aggiornamento dell'elaborazione, i costi sostenuti e la configurazione dei timeout, consulta [Scrivi e modifica il codice con Dev Environments in CodeCatalyst](#)

Argomenti

- [Visualizzazione degli ambienti di sviluppo per il tuo spazio](#)
- [Modifica di un ambiente di sviluppo per il tuo spazio](#)
- [Interruzione di un ambiente di sviluppo per il tuo spazio](#)
- [Eliminazione di un ambiente di sviluppo per il tuo spazio](#)

Visualizzazione degli ambienti di sviluppo per il tuo spazio

Puoi visualizzare il tipo, lo stato e i dettagli di tutti gli ambienti di sviluppo nel tuo spazio. Per ulteriori informazioni sulla creazione e l'esecuzione di ambienti di sviluppo, consulta [Creazione di un ambiente di sviluppo](#).

È necessario disporre del ruolo di amministratore dello spazio per visualizzare questa pagina e gestire gli ambienti di sviluppo a livello di spazio.

Per visualizzare gli ambienti di sviluppo nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Dev Environments.

La pagina elenca tutti gli ambienti di sviluppo presenti nel tuo spazio. È possibile visualizzare il nome della risorsa, l'alias della risorsa, se applicabile, il tipo di IDE, il calcolo e lo storage predefiniti o configurati e il timeout configurato per ogni ambiente di sviluppo.

Modifica di un ambiente di sviluppo per il tuo spazio

Puoi modificare la configurazione di un ambiente di sviluppo, ad esempio l'eventuale periodo di timeout configurato per l'interruzione dell'esecuzione di un ambiente di sviluppo inattivo. Per ulteriori informazioni sulla modifica di un ambiente di sviluppo, consulta [Modifica di un ambiente di sviluppo](#).

È necessario disporre del ruolo di amministratore dello spazio per visualizzare questa pagina e gestire gli ambienti di sviluppo a livello di spazio.

Per modificare Dev Environments nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Dev Environments.
4. Scegli il selettore accanto all'ambiente di sviluppo che desideri gestire. Scegli Modifica.
5. Apporta le modifiche desiderate al timeout di calcolo o di inattività per l'ambiente di sviluppo.
6. Scegli Save (Salva).

Interruzione di un ambiente di sviluppo per il tuo spazio

Puoi interrompere un ambiente di sviluppo in esecuzione prima che diventi inattivo se l'ambiente di sviluppo è configurato per avere un timeout. Altrimenti, un ambiente di sviluppo con un timeout scaduto verrà già interrotto. Per ulteriori informazioni sull'interruzione di un ambiente di sviluppo, consulta [Interruzione di un ambiente di sviluppo](#)

È necessario disporre del ruolo di amministratore dello spazio per visualizzare questa pagina e gestire gli ambienti di sviluppo a livello di spazio.

Per fermare un ambiente di sviluppo nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Dev Environments.
4. Scegli il selettore accanto all'ambiente di sviluppo che desideri gestire. Scegli Stop (Arresta).

Eliminazione di un ambiente di sviluppo per il tuo spazio

Puoi eliminare un ambiente di sviluppo che non è più necessario o che non ha più un proprietario. Per ulteriori informazioni sulle considerazioni relative all'eliminazione di un ambiente di sviluppo, consulta [Eliminazione di un ambiente di sviluppo](#)

È necessario disporre del ruolo di amministratore dello spazio per visualizzare questa pagina e gestire gli ambienti di sviluppo a livello di spazio.

Per eliminare Dev Environments nel tuo spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

 Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Scegli Impostazioni, quindi scegli Dev Environments.
4. Scegli il selettore accanto all'ambiente di sviluppo che desideri gestire. Scegli Elimina. Per confermare, digita `delete`, quindi scegli Elimina.

Quote per gli spazi

La tabella seguente descrive le quote e i limiti per gli spazi in Amazon CodeCatalyst. Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Numero massimo di canali Slack per uno spazio	500
Numero massimo di inviti per un indirizzo email	25
Numero massimo di inviti per un utente	500
Numero massimo di spazi attivi per utente per Regione AWS	5
Numero massimo di creazioni di spazio per regione al mese per utente	5
Numero massimo di gruppi SSO per team	5
Numero massimo di team per ogni spazio	100

Numero massimo di utenti per team	1000
Descrizioni degli spazi	Le descrizioni degli spazi sono facoltative. Se specificate, devono avere una lunghezza compresa tra 0 e 200 caratteri. Possono contenere qualsiasi combinazione di lettere, numeri, spazi, punti, caratteri di sottolineatura, virgole, trattini e i seguenti caratteri speciali: ? & \$ % + = / \ ; : \n \t \r
Nomi degli spazi	I nomi degli spazi devono essere univoci in tutto CodeCatalyst. Non è possibile riutilizzare i nomi degli spazi eliminati. I nomi degli spazi devono avere una lunghezza compresa tra 3 e 63 caratteri. Devono inoltre iniziare con un carattere alfanumerico. I nomi degli spazi possono contenere qualsiasi combinazione di lettere, numeri, punti, caratteri di sottolineatura e trattini. Non possono contenere nessuno dei seguenti caratteri: ! ? @ # \$ % ^ & * () + = { } [] \ > < ~ ` ' " ; :

Organizza il lavoro con i progetti in CodeCatalyst

Utilizzi i progetti in Amazon CodeCatalyst per creare uno spazio di collaborazione in cui i team di sviluppo possano svolgere attività di sviluppo con flussi di lavoro e repository condivisi di integration/continuous distribuzione continua (CI/CD). Quando crei un progetto, puoi aggiungere, aggiornare o rimuovere risorse. Puoi anche monitorare l'avanzamento del lavoro del tuo team. Puoi avere più progetti all'interno di uno spazio.

Gli spazi in CodeCatalyst sono costituiti da progetti. Puoi vedere tutti i progetti all'interno del tuo spazio, ma puoi usare solo i progetti di cui sei membro. Quando crei un progetto, vengono generati i ruoli predefiniti per il tuo progetto, che assegni agli utenti che inviti al progetto.

- Chiunque sia assegnato al progetto con un ruolo di progetto, ad esempio il ruolo di Collaboratore, può accedere alle risorse del progetto, ad esempio un archivio di origine.
- Chiunque abbia il ruolo di amministratore dello spazio o di amministratore del progetto può inviare inviti a partecipare a un progetto.
- Gli utenti con il ruolo di amministratore del progetto possono tenere traccia dell'attività, dello stato e di altre impostazioni tra le risorse condivise.
- Gli utenti con il ruolo di accesso limitato possono gestire le assegnazioni di funzionalità, correzioni di codice e test dei progetti nell'ambito dei flussi di CI/CD lavoro.

I flussi di lavoro vengono utilizzati per creare, testare e rilasciare o aggiornare le applicazioni come CI/CD pipeline. You can assemble workflows by adding actions that transfer and work on your source artifacts. When you run actions, your project cloud resources are used to provide on-demand compute ability for your workflow actions. You might configure more CI/CD flussi di lavoro in base all'attività e all'output che si desidera configurare. Ad esempio, potresti creare un flusso di lavoro solo per le azioni di compilazione e test, in cui puoi visualizzare i risultati dei test e completare il flusso di lavoro senza una distribuzione mentre correggi i bug. Quindi, potresti creare un altro flusso di lavoro per creare e distribuire l'applicazione in un ambiente di staging.

Quando crei un progetto, puoi utilizzare un blueprint per creare un progetto che contenga codice di esempio e crei risorse oppure puoi iniziare con un progetto vuoto. Se si crea un progetto utilizzando un blueprint, il blueprint scelto determina quali risorse vengono aggiunte al progetto e gli strumenti di CodeCatalyst creazione o configurazione in modo da poter tenere traccia e utilizzare le risorse del progetto. È possibile aggiungere o rimuovere manualmente le risorse dopo aver creato un progetto.

Ogni progetto tiene traccia dell'attività del progetto sotto forma di elenco di eventi per utente, ad esempio quando un progetto viene creato o una risorsa viene modificata. L'attività del progetto viene monitorata e aggregata a livello di spazio. Per ulteriori informazioni sull'utilizzo dei dati delle attività, consulta [Visualizzazione di tutti i progetti in uno spazio](#).

Se il progetto utilizza AWS risorse, puoi collegare il tuo CodeCatalyst account a un AWS account in cui disponi delle autorizzazioni amministrative per integrare le risorse per il progetto.

Puoi aggiungere repository di sorgenti, problemi e altre risorse al progetto dopo averlo creato. È necessario disporre del ruolo di amministratore dello spazio per creare progetti.

Creare un progetto

Con CodeCatalyst i progetti, puoi svolgere attività di sviluppo con flussi di lavoro e repository di integrazione/continuous distribuzione continua condivisa (CI/CD), gestire risorse, tenere traccia dei problemi e aggiungere utenti.

Prima di creare un progetto, è necessario disporre del ruolo di amministratore dello spazio o Power user.

Argomenti

- [Creare un progetto vuoto in Amazon CodeCatalyst](#)
- [Creazione di un progetto con un repository di terze parti collegato](#)
- [Creare un progetto con un blueprint](#)
- [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#)
- [Le migliori pratiche per utilizzare i blueprint con i progetti](#)
- [Aggiungere risorse e attività ai progetti creati](#)

Creare un progetto vuoto in Amazon CodeCatalyst

Puoi creare un progetto vuoto senza risorse e aggiungere manualmente le risorse che desideri in un secondo momento.

Prima di creare un progetto, devi avere il ruolo di amministratore dello spazio o Power user.

Per creare un progetto vuoto

1. Vai allo spazio in cui desideri creare un progetto.
2. Nella dashboard dello spazio, scegli Crea progetto.
3. Scegli Inizia da zero.
4. In Dai un nome al tuo progetto, inserisci il nome che desideri assegnare al progetto. Il nome deve essere univoco all'interno del tuo spazio.
5. Seleziona Crea progetto.

Creazione di un progetto con un repository di terze parti collegato

Puoi conservare il codice sorgente del tuo progetto in un fornitore terzo preferito e continuare a utilizzare tutte le CodeCatalyst funzionalità come progetti, gestione del ciclo di vita, flussi di lavoro e altro ancora. A tale scopo, puoi creare un nuovo CodeCatalyst progetto collegato a un repository, a un GitHub repository Bitbucket o a un repository di progetto. GitLab Puoi quindi utilizzare il tuo repository di sorgenti collegato nel tuo progetto. CodeCatalyst

Prima di creare un CodeCatalyst progetto, è necessario disporre del ruolo di amministratore dello spazio o Power user. Per ulteriori informazioni, consultare [Creare uno spazio](#) e [Invitare un utente direttamente in uno spazio](#).

Per creare un progetto collegato a un archivio di sorgenti del tuo GitHub account, dovrai completare le tre attività seguenti: CodeCatalyst

1. Installa i GitHub repository, i repository Bitbucket o l'estensione dei repository. GitLab In un sito esterno ti viene richiesto di connetterti e fornire CodeCatalyst l'accesso al tuo repository, operazione che viene eseguita come parte del passaggio successivo.

Important

Per installare i GitHub repository, i repository Bitbucket o l'estensione dei GitLab repository nel tuo CodeCatalyst spazio, devi aver effettuato l'accesso con un account che abbia il ruolo di amministratore dello spazio.

2. Connect il tuo GitHub account o l'area di lavoro Bitbucket o GitLab l'utente a. CodeCatalyst

 Important

Per connettere il tuo GitHub account, Bitbucket workspace, GitLab utente al tuo CodeCatalyst spazio, devi essere sia l'amministratore della fonte di terze parti che l'amministratore dello spazio. CodeCatalyst

 Important

Dopo aver installato un'estensione del repository, il codice di tutti gli archivi a cui ti colleghi CodeCatalyst verrà indicizzato e archiviato. CodeCatalyst Ciò renderà il codice ricercabile in. CodeCatalyst Per comprendere meglio la protezione dei dati per il tuo codice quando utilizzi repository collegati in CodeCatalyst, consulta la sezione [Protezione dei dati](#) nella Amazon CodeCatalyst User Guide.

3. Crea un CodeCatalyst progetto collegato al tuo GitHub repository, al repository Bitbucket o al repository del progetto. GitLab

 Important

Sebbene tu possa collegare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto come collaboratore, puoi scollegare un repository di terze parti solo come amministratore dello spazio o amministratore del progetto. Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

 Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

È consigliabile assicurarsi sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Note

- Un GitHub repository, un repository Bitbucket o un repository di GitLab progetto possono essere collegati a un solo progetto in uno spazio. CodeCatalyst
- Non puoi utilizzare repository vuoti o archiviati, repository Bitbucket o GitHub repository di progetti con progetti. GitLab CodeCatalyst
- Non puoi collegare un GitHub repository, un repository Bitbucket o un repository di progetti con lo stesso nome di un repository in un GitLab progetto. CodeCatalyst
- L'estensione GitHub repositories non è compatibile con gli archivi di Enterprise Server. GitHub
- L'estensione Bitbucket repositories non è compatibile con i repository Bitbucket Data Center.
- L'estensione GitLab repositories non è compatibile con gli archivi di progetto autogestiti. GitLab
- Non puoi utilizzare le funzioni Scrivi descrizione per me o Riepiloga commenti con gli archivi collegati. Queste funzionalità sono disponibili solo nelle pull request in. CodeCatalyst

Per ulteriori informazioni, consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#).

Per installare l'estensione di terze parti

1. Vai allo spazio in cui desideri creare un progetto.
2. Nella dashboard dello spazio, scegli Crea progetto.
3. Scegli Porta il tuo codice.
4. In Collega repository esistente, scegli GitHub repository, repository Bitbucket, repository a seconda del provider di GitLab repository di terze parti che desideri utilizzare. Ti viene richiesto di connettere il tuo GitHub account, l'area di lavoro Bitbucket o l'account se non l'hai fatto in precedenza. GitLab Se l'estensione di terze parti di tua scelta non è già installata, viene visualizzata una richiesta di installazione.

5. Se richiesto, scegli Installa. Controlla le autorizzazioni richieste dall'estensione e, se vuoi continuare, scegli nuovamente Installa.

Dopo aver installato l'estensione di terze parti, il passaggio successivo consiste nel connettere il tuo GitHub account, l'area di lavoro Bitbucket o GitLab l'utente al tuo spazio. CodeCatalyst

Per connettere il tuo GitHub account, l'area di lavoro Bitbucket o l'utente a GitLab CodeCatalyst

Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai scelto di configurare:

- GitHub repository: Connect a un GitHub account.
 1. Scegli Connect GitHub account per accedere al sito esterno per GitHub.
 2. Accedi al tuo GitHub account utilizzando GitHub le tue credenziali, quindi scegli l'account in cui desideri installare Amazon CodeCatalyst.

 Tip

Se in precedenza hai collegato un GitHub account allo spazio, non ti verrà richiesto di autorizzarlo nuovamente. Vedrai invece una finestra di dialogo che ti chiede dove desideri installare l'estensione se sei un membro o un collaboratore in più di uno GitHub spazio, o la pagina di configurazione per l' CodeCatalyst applicazione Amazon se appartieni a un solo GitHub spazio. Configura l'applicazione per l'accesso al repository che desideri consentire, quindi scegli Salva. Se il pulsante Salva non è attivo, apporta una modifica alla configurazione, quindi riprova.

3. Scegli se consentire l'accesso CodeCatalyst a tutti i repository attuali e futuri o scegli i GitHub repository specifici in cui desideri utilizzare. CodeCatalyst L'opzione predefinita prevede l'inclusione di tutti i GitHub repository nell' GitHub account, compresi i repository futuri a cui sarà possibile accedere da. CodeCatalyst
4. Controlla le autorizzazioni concesse a CodeCatalyst, quindi scegli Installa.

Dopo aver collegato l' GitHub account a CodeCatalyst, verrai reindirizzato alla pagina dei dettagli dell'estensione dei GitHub repository, dove puoi visualizzare e gestire GitHub gli account connessi e gli archivi collegati GitHub .

- Archivi Bitbucket: Connettiti a uno spazio di lavoro Bitbucket.

1. Scegli Connect Bitbucket workspace per accedere al sito esterno di Bitbucket.
2. Accedi al tuo spazio di lavoro Bitbucket utilizzando le tue credenziali Bitbucket e controlla le autorizzazioni concesse a. CodeCatalyst
3. Dal menu a discesa Autorizza per l'area di lavoro, scegli l'area di lavoro Bitbucket a cui desideri fornire l'accesso, quindi scegli Concedi l'accesso. CodeCatalyst

 Tip

Se in precedenza hai collegato un'area di lavoro Bitbucket allo spazio, non ti verrà richiesto di autorizzare nuovamente. Vedrai invece una finestra di dialogo che ti chiede dove desideri installare l'estensione se sei un membro o un collaboratore in più di uno spazio di lavoro Bitbucket, o la pagina di configurazione per l' CodeCatalyst applicazione Amazon se appartieni a un solo spazio di lavoro Bitbucket. Configura l'applicazione per l'accesso allo spazio di lavoro che desideri consentire, quindi scegli Concedi l'accesso. Se il pulsante Concedi l'accesso non è attivo, apporta una modifica alla configurazione, quindi riprova.

Dopo aver collegato l'area di lavoro Bitbucket a CodeCatalyst, verrai indirizzato alla pagina dei dettagli dell'estensione dei repository Bitbucket, dove puoi visualizzare e gestire le aree di lavoro Bitbucket connesse e gli archivi Bitbucket collegati.

- GitLab repository: Connect a un GitLab utente.

1. Scegli Connect GitLab user per accedere al sito esterno per GitLab.
2. Accedi al tuo GitLab utente utilizzando GitLab le tue credenziali e controlla le autorizzazioni concesse a. CodeCatalyst

 Tip

Se in precedenza hai collegato un GitLab utente allo spazio, non ti verrà richiesto di autorizzarlo nuovamente. Verrai invece reindirizzato alla console. CodeCatalyst

3. Scegli Authorize AWS Connector per. GitLab

Dopo aver collegato GitLab l'utente a CodeCatalyst, verrai indirizzato alla pagina dei dettagli GitLab delle estensioni dei repository, dove puoi visualizzare e gestire gli archivi connessi GitLab degli utenti e dei GitLab progetti collegati.

Dopo aver collegato la fonte di terze parti a CodeCatalyst, puoi collegare gli archivi di terze parti ai tuoi progetti. CodeCatalyst

Creazione del tuo progetto

1. Nella pagina Crea progetto, scegli l' GitHub account che hai collegato.
2. A seconda del provider di repository di terze parti che hai collegato, scegli i GitHub repository, i repository Bitbucket o il menu a discesa dei GitLab repository dei repository per visualizzare i repository di terze parti, quindi scegli il repository che desideri collegare al tuo progetto.
3. Nel campo di immissione di testo Assegna un nome al progetto, inserisci il nome che desideri assegnare al progetto. Il nome deve essere univoco all'interno del tuo spazio.
4. Seleziona Crea progetto.

Dopo aver installato i GitHub repository, i repository Bitbucket o l'estensione dei GitLab repository, aver collegato il provider di risorse e aver collegato il repository di terze parti al CodeCatalyst progetto, puoi utilizzarlo nei flussi di lavoro e negli ambienti di sviluppo. CodeCatalyst Puoi anche creare repository di terze parti nell' GitHub account connesso, nell'area di lavoro Bitbucket o nell'utente con codice generato da un blueprint. GitLab Puoi anche utilizzare i repository collegati con Amazon Q Developer, blueprint e altro ancora. Per ulteriori informazioni, consultare [Avvio automatico di un flusso di lavoro eseguito dopo eventi di repository di terze parti](#) e [Creazione di un ambiente di sviluppo](#).

Creare un progetto con un blueprint

Puoi fornire tutte le risorse del progetto e il codice di esempio con un blueprint di progetto. Per informazioni sui progetti, consulta la [Creazione di un progetto completo con CodeCatalyst progetti](#)

Per creare un progetto con un blueprint

1. Nella CodeCatalyst console, accedi allo spazio in cui desideri creare un progetto.
2. Nella dashboard dello spazio, scegli Crea progetto.

3. Scegli Inizia con un progetto.

Tip

Puoi scegliere di aggiungere un blueprint fornendo ad Amazon Q i requisiti del tuo progetto in modo che Amazon Q ti suggerisca un blueprint. Per ulteriori informazioni, consultare [Utilizzo di Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di funzionalità](#) e [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#). Questa funzionalità è disponibile solo nella regione Stati Uniti occidentali (Oregon).

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

4. Dalla scheda CodeCatalyst Blueprint o Space blueprint, scegli un blueprint, quindi scegli Avanti.
5. In Assegna un nome al progetto, inserisci il nome che desideri assegnare al progetto e i nomi delle risorse associate. Il nome deve essere univoco all'interno del tuo spazio.
6. (Facoltativo) Per impostazione predefinita, il codice sorgente creato dal blueprint viene archiviato in un CodeCatalyst repository. In alternativa, è possibile scegliere di archiviare il codice sorgente del blueprint in un repository di terze parti. Per ulteriori informazioni, consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#).

Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

È consigliabile assicurarsi sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Effettua una delle seguenti operazioni a seconda del provider di repository di terze parti che desideri utilizzare:

- GitHub repository: Connect a un GitHub account.

Scegli il menu a discesa Avanzato, scegli GitHub come provider del repository, quindi scegli l'GitHub account in cui desideri archiviare il codice sorgente creato dal blueprint.

Note

Se stai collegando un GitHub account, devi creare una connessione personale per stabilire una mappatura delle identità tra la tua identità e la tua CodeCatalyst identità. GitHub Per ulteriori informazioni, consultare [Connessioni personali](#) e [Accesso alle GitHub risorse con connessioni personali](#).

- Archivi Bitbucket: Connect uno spazio di lavoro Bitbucket.

Scegli il menu a discesa Avanzato, scegli Bitbucket come provider di repository, quindi scegli l'area di lavoro Bitbucket in cui desideri archiviare il codice sorgente creato dal blueprint.

- GitLab repository: Connect a un GitLab utente.

Scegli il menu a discesa Avanzato, scegli GitLab come provider del repository, quindi scegli l'GitLab utente in cui desideri archiviare il codice sorgente creato dal blueprint.

7. In Risorse del progetto, configura i parametri del blueprint. A seconda del blueprint, potresti avere la possibilità di assegnare un nome al repository di origine.
8. (Facoltativo) Per visualizzare i file di definizione con aggiornamenti basati sulle selezioni dei parametri di progetto effettuate, scegliete Visualizza codice o Visualizza flusso di lavoro da Genera anteprima del progetto.
9. (Facoltativo) Scegli Visualizza dettagli dalla scheda del progetto per visualizzare dettagli specifici sul progetto, come una panoramica dell'architettura del progetto, le connessioni e le autorizzazioni richieste e il tipo di risorse create dal blueprint.
10. Seleziona Crea progetto.

Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint

Quando crei un progetto o desideri aggiungere nuovi componenti a un progetto esistente, potresti non essere sicuro di quale blueprint utilizzare o di come integrare le funzionalità. CodeCatalyst include

l'integrazione con un assistente AI generativo chiamato Amazon Q in grado di analizzare i requisiti del progetto e suggerire un modello più adatto alle tue esigenze.

Puoi usare Amazon Q per aiutarti a creare un progetto con un blueprint che crea componenti in base ai tuoi requisiti oppure puoi usare Amazon Q per aiutarti ad aggiungere un blueprint a un progetto esistente. Ad esempio, per aggiungere risorse per un'applicazione Web o un'applicazione moderna a un progetto, specifica i requisiti e quindi le risorse verranno aggiunte con un blueprint consigliato. I problemi relativi ai componenti rimanenti possono essere creati automaticamente.

Amazon Q crea anche problemi relativi ai requisiti che non possono essere risolti con un blueprint suggerito. Inoltre, puoi assegnare tali problemi ad Amazon Q. Se assegni il problema ad Amazon Q, quest'ultima tenterà di creare una bozza di soluzione da valutare. Questo può aiutare te e il tuo team a concentrarti e ottimizzare il lavoro su questioni che richiedono la tua attenzione, mentre Amazon Q lavora a una soluzione per i problemi che non hai risorse per risolvere immediatamente.

Note

Realizzato da Amazon Bedrock: AWS implementa il rilevamento [automatico degli abusi](#).

Poiché le funzionalità *Scrivi una descrizione per me*, *Crea un riepilogo dei commenti*, *Attività consigliate*, *Usa Amazon Q per creare o aggiungere funzionalità a un progetto* e *Assegna i problemi ad Amazon Q con Amazon Q Developer Agent* per lo sviluppo software sono supportate da Amazon Bedrock, gli utenti possono utilizzare al meglio i controlli implementati in Amazon Bedrock per rafforzare la protezione, la sicurezza e l'uso responsabile dell'intelligenza artificiale (IA).

Di seguito sono riportate alcune best practice per aiutarti a creare progetti e aggiungere blueprint con Amazon Q.

Important

Le funzionalità di intelligenza artificiale generativa sono disponibili solo nella regione Stati Uniti occidentali (Oregon).

- Utilizza le istruzioni predefinite fornite da Amazon Q. Amazon Q fa meglio a scegliere i blueprint dai prompt forniti.

- Utilizza le opzioni di configurazione suggerite da Amazon Q per visualizzare in anteprima i blueprint. Scegli un blueprint per visualizzare in anteprima il codice di esempio e le risorse che verranno create dal blueprint.
- Usa uno spazio abilitato per Amazon Q. Per creare un progetto con Amazon Q o aggiungere funzionalità a un progetto con blueprint utilizzando Amazon Q, utilizza uno spazio abilitato per le funzionalità di intelligenza artificiale generativa. Per ulteriori informazioni, consulta [Abilitare o disabilitare le funzionalità di intelligenza artificiale generativa](#) per uno spazio.
- Ottieni ulteriori informazioni sui progetti consigliati da Amazon Q. Potresti voler saperne di più sul tipo di risorse del progetto, sul codice di esempio e sui componenti creati con uno specifico blueprint consigliato. Per ulteriori informazioni sui blueprint disponibili in CodeCatalyst, vedere. [Creazione di un progetto completo con CodeCatalyst progetti](#)
- Consenti ad Amazon Q di risolvere i problemi. Consenti ad Amazon Q di creare problemi per te, assegnarli e monitorarli. Per ulteriori informazioni, consulta [Tutorial: Utilizzo delle funzionalità di intelligenza artificiale CodeCatalyst generativa per velocizzare il lavoro di sviluppo](#).
- Annulla l'assegnazione ad Amazon Q di problemi su cui non si lavora più. Dopo aver completato l'esempio, annulla l'assegnazione ad Amazon Q di eventuali problemi su cui non stai più lavorando. Se Amazon Q ha terminato il lavoro su un problema o non è riuscito a trovare una soluzione, assicurati di annullare l'assegnazione di Amazon Q per evitare di raggiungere la quota massima per le funzionalità di intelligenza artificiale generativa. [Per ulteriori informazioni, consulta Gestione delle funzionalità e prezzi dell'IA generativa](#).
- Visualizza l'utilizzo per Amazon Q. Puoi visualizzare l'utilizzo delle funzionalità di intelligenza artificiale generativa a livello di utente. Vai a Le mie impostazioni per gestire le quote AI generative e visualizzare l'utilizzo in base al tuo ID Builder o all'identità Single Sign-On (SSO). Per ulteriori informazioni, consulta [Visualizzazione dell'utilizzo delle funzionalità di intelligenza artificiale generativa in uno spazio](#).

 Important

Le funzionalità di intelligenza artificiale generativa di CodeCatalyst sono soggette a quote. [Per ulteriori informazioni, consulta Amazon Q Developer Pricing, Abilitazione o disabilitazione delle funzionalità di intelligenza artificiale generativa per uno spazio e Fatturazione](#).

Le migliori pratiche per utilizzare i blueprint con i progetti

Di seguito sono riportate alcune best practice per aiutarti a creare un progetto con blueprint o ad aggiungere blueprint.

- Utilizza i blueprint forniti da CodeCatalyst per creare o aggiungere progetti. Puoi usare i blueprint per creare un progetto completo con codice sorgente e risorse per gli sviluppatori. Ad esempio, il blueprint dell'applicazione Web crea risorse applicative e infrastrutturali e distribuisce un'applicazione Web. È possibile creare un progetto con un blueprint o aggiungere un blueprint personalizzato a un progetto esistente. Per ulteriori informazioni, consulta [Creazione di un progetto con un blueprint](#). Visualizza qualsiasi blueprint in CodeCatalyst per visualizzare in anteprima il codice di esempio e le risorse che verranno create dal blueprint.
- Utilizza progetti personalizzati progettati dalla tua organizzazione. Puoi utilizzare progetti personalizzati per creare un progetto completo nel tuo spazio. Il blueprint personalizzato progettato dall'organizzazione può fornire standardizzazione e best practice, che possono anche aiutare a ridurre gli sforzi per la creazione di un nuovo progetto. In qualità di autore di blueprint personalizzati, puoi visualizzare i dettagli sui progetti che utilizzano il tuo blueprint in tutto il tuo spazio. La gestione del ciclo di vita consente di gestire centralmente il ciclo di vita dello sviluppo del software di ogni progetto e gli utenti dei blueprint possono utilizzare la gestione del ciclo di vita per rigenerare una base di codice a partire da opzioni o versioni aggiornate di un blueprint. Per ulteriori informazioni, consulta [Lavorare con la gestione del ciclo di vita come autore di progetti](#).
- Aggiungi il ruolo di sviluppatore o i ruoli IAM appropriati all'account del tuo progetto. Durante o dopo aver completato i passaggi di creazione del progetto, puoi configurare le autorizzazioni del blueprint scegliendo o creando ruoli IAM in un Account AWS ambiente connesso allo spazio.

Aggiungere risorse e attività ai progetti creati

Una volta che il progetto è pronto, puoi aggiungere risorse e attività.

- Per ulteriori informazioni sui CI/CD flussi di lavoro creati con il progetto, consulta [Guida introduttiva ai flussi di lavoro](#).
- Per utilizzare azioni di compilazione simili a quelle del tuo nuovo progetto che distribuiscono artefatti di compilazione in un bucket Amazon S3, consulta e. [Creazione con flussi di lavoro Tutorial: caricare artefatti su Amazon S3](#)

- Per iniziare con un progetto vuoto e iniziare a distribuire un'applicazione serverless simile con una distribuzione in stack, consulta [CloudFormation Tutorial: Implementazione di un'applicazione serverless](#)
- Per aggiungere una scheda di pianificazione dei problemi, consulta [Tieni traccia e organizza il lavoro con problemi in CodeCatalyst](#)
- Per visualizzare la panoramica del progetto, lo stato del progetto, l'attività recente del team e il lavoro assegnato, consulta [Ottenere un elenco di progetti](#).
- Per visualizzare il codice sorgente o creare una pull request, consulta [Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst](#).
- Per configurare notifiche che inviano avvisi di stato in caso di esito positivo o negativo del flusso di lavoro, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).
- Per invitare membri al tuo progetto, consulta [Concessione delle autorizzazioni di progetto agli utenti](#).
- Per configurare gli ambienti di sviluppo, vedi [Scrivi e modifica il codice con Dev Environments in CodeCatalyst](#).

Ottenere un elenco di progetti

Dal tuo CodeCatalyst spazio, puoi visualizzare i dettagli di ogni progetto per cui disponi delle autorizzazioni di progetto.

Per visualizzare un progetto, devi essere un membro del progetto o avere il ruolo di amministratore dello spazio per lo spazio.

Se non hai ancora creato un progetto, consulta [Creare un progetto](#). È necessario disporre del ruolo di amministratore dello spazio per lo spazio in cui si desidera creare un progetto.

- Nella panoramica del progetto, puoi visualizzare i membri del progetto, gli archivi dei sorgenti, le esecuzioni del flusso di lavoro, le richieste pull aperte, gli ambienti di sviluppo del progetto e i problemi.
- Nelle impostazioni del progetto, puoi visualizzare e gestire i dettagli del progetto, eliminare il progetto, invitare nuovi membri al progetto, gestire i membri del progetto e configurare le notifiche.

Visualizzazione delle attività del progetto e degli ambienti di sviluppo

Per visualizzare un riepilogo delle attività del progetto, ad esempio problemi aperti e richieste pull che ti vengono assegnate o create da te, e gli ambienti di sviluppo associati al progetto, usa la console.

Per visualizzare un progetto, devi essere un membro del progetto o avere il ruolo di amministratore dello spazio.

Per visualizzare gli archivi di origine, le esecuzioni del flusso di lavoro, i problemi, le richieste pull, gli ambienti di sviluppo e i problemi

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione seleziona Overview (Panoramica).
4. Visualizza le attività del progetto che ti sono state assegnate e create da te.
 - Visualizza i membri + Visualizza tutto per visualizzare un elenco dei membri del progetto.
 - Visualizza la scheda Repository per visualizzare i repository di origine associati al progetto.
 - Visualizza la scheda Workflow runs per visualizzare i flussi di lavoro associati al progetto.
 - Visualizza la scheda Open pull request per visualizzare un riepilogo dello stato del repository di codice, oltre alle richieste pull assegnate all'utente e create dall'utente.
 - Visualizza la scheda My Dev Environments per visualizzare un riepilogo degli ambienti di sviluppo associati al progetto.
 - Visualizza la scheda Problemi per visualizzare un riepilogo delle attività assegnate o delle attività che hai creato.

Visualizzazione di tutti i progetti in uno spazio

Nell'elenco Progetti del tuo spazio, puoi visualizzare tutti i progetti per i quali disponi delle autorizzazioni.

Per visualizzare un riepilogo delle attività del progetto, ad esempio problemi aperti e richieste pull che ti vengono assegnate o create da te, e gli ambienti di sviluppo associati al progetto, usa la console.

Per visualizzare un progetto, devi essere un membro del progetto o avere il ruolo di amministratore dello spazio.

Per visualizzare gli archivi di origine, le esecuzioni del flusso di lavoro, i problemi, le richieste pull, gli ambienti di sviluppo e i problemi

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.

2. Vai allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Visualizza il nome, il percorso, l'ID del progetto e la descrizione.

Visualizzazione delle impostazioni del progetto

Nelle impostazioni del progetto, puoi visualizzare i membri del progetto, i repository dei sorgenti, le esecuzioni del flusso di lavoro, le richieste pull aperte, gli ambienti di sviluppo del progetto e i problemi.

Per visualizzare un riepilogo delle attività del progetto, come i problemi aperti e le richieste pull che ti vengono assegnate o create da te, e gli ambienti di sviluppo associati al progetto, usa la console.

Per visualizzare gli archivi di origine, le esecuzioni del flusso di lavoro, i problemi, le richieste pull, gli ambienti di sviluppo e i problemi

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Visualizza il nome, il percorso, l'ID del progetto e la descrizione.

Passare a un progetto diverso in CodeCatalyst

Per passare a un progetto diverso, usa la console per scegliere da un elenco di progetti a cui hai accesso.

Per passare a un progetto diverso

1. Nella CodeCatalyst console, scegli il selettore di progetto in alto.
2. Espandi il menu a discesa e scegli il progetto verso cui desideri navigare.

Eliminazione di un progetto

È possibile eliminare un progetto per rimuovere tutti gli accessi alle risorse del progetto. È necessario disporre del ruolo di amministratore dello spazio o amministratore del progetto per eliminare un

progetto. Dopo aver eliminato un progetto, i membri del progetto non saranno in grado di accedere alle risorse del progetto e tutti i flussi di lavoro richiesti dagli archivi di origine di terze parti verranno interrotti.

Per eliminare il progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Passa allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Seleziona Delete project (Elimina progetto).
5. Invio **delete** per confermare l'eliminazione.
6. Seleziona Delete project (Elimina progetto).

Concessione delle autorizzazioni di progetto agli utenti

Puoi gestire i membri dei tuoi progetti utilizzando la CodeCatalyst console Amazon. Puoi aggiungere o rimuovere utenti, gestire i ruoli dei membri attuali, inviare inviti a partecipare al tuo progetto e annullare gli inviti che non sono ancora stati accettati.

Nella pagina dei membri per gli utenti dello spazio e del progetto, gli utenti possono avere più ruoli. Gli utenti con più ruoli mostreranno un indicatore quando hanno più ruoli e verranno visualizzati per primi con il ruolo con il maggior numero di autorizzazioni.

Ottenere un elenco dei membri e dei relativi ruoli nel progetto

Quando aggiungi un utente al tuo progetto, assegna un ruolo che concede le autorizzazioni al progetto come segue:

- Il ruolo di amministratore del progetto dispone di tutte le autorizzazioni in un progetto. Assegna questo ruolo solo agli utenti che devono amministrare ogni aspetto di un progetto, inclusa la modifica delle impostazioni del progetto, la gestione delle autorizzazioni del progetto e l'eliminazione del progetto. Per ulteriori informazioni, consulta [Ruolo di amministratore del progetto](#).
- Il ruolo Collaboratore dispone delle autorizzazioni necessarie per lavorare in un progetto. Assegna questo ruolo agli utenti che devono lavorare con codice, flussi di lavoro, problemi e azioni in un progetto. Per ulteriori informazioni, consulta [Ruolo di collaboratore](#).

- Il ruolo Revisore dispone delle autorizzazioni di revisione. Per informazioni dettagliate, vedi [Ruolo di revisore](#).
- Il ruolo Sola lettura dispone di autorizzazioni di lettura. Per informazioni dettagliate, vedi [Ruolo di sola lettura](#).

Non è necessario invitare al progetto un utente con il ruolo di amministratore dello spazio, poiché ha già accesso implicito a tutti i progetti nello spazio.

Quando inviti un utente al tuo progetto (senza assegnare il ruolo di amministratore dello spazio), l'utente verrà visualizzato nella tabella Membri del progetto sotto i progetti e nella tabella Membri del progetto sotto gli spazi.

Per visualizzare utenti e ruoli in uno spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Passa allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Scegli la scheda Membri.

La tabella Membri del progetto mostra tutti i membri che hanno un ruolo in un progetto.

 Tip

Se hai il ruolo di amministratore dello spazio, puoi visualizzare a quali progetti sei stato invitato direttamente. Vai alle impostazioni del progetto, quindi scegli I miei progetti.

La tabella degli amministratori dello spazio mostra gli utenti con il ruolo di amministratore dello spazio. Questi utenti vengono assegnati automaticamente (implicitamente) a tutti i progetti nello spazio e non hanno un ruolo in un progetto.

Nella colonna Status, i seguenti sono valori validi:

- CodeCatalyst Invitato: ha inviato l'invito ma l'utente non l'ha ancora accettato o rifiutato.
- Membro: l'utente ha accettato l'invito.

Argomenti

- [Invitare un utente a un progetto](#)
- [Annullamento di un invito](#)
- [Rimuovere un utente dal progetto](#)
- [Accettare o rifiutare un invito per un progetto](#)

Invitare un utente a un progetto

Puoi usare la console per invitare utenti al tuo progetto. Puoi invitare membri del tuo spazio o aggiungere nomi esterni al tuo spazio.

Per invitare utenti al progetto, è necessario accedere con il ruolo di amministratore del progetto o amministratore dello spazio.

Non è necessario invitare un utente con il ruolo di amministratore dello spazio al progetto perché ha già accesso implicito a tutti i progetti nello spazio.

Quando inviti un utente al tuo progetto (senza assegnare il ruolo di amministratore dello spazio), l'utente verrà visualizzato nella tabella Membri del progetto sotto i progetti e nella tabella Membri del progetto sotto gli spazi.

Per invitare un membro al progetto dalla scheda Impostazioni del progetto

1. Vai al tuo progetto.

Tip

Puoi scegliere quale progetto visualizzare nella barra di navigazione in alto.

2. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Scegli la scheda Membri.
4. In Membri del progetto, scegli Invita nuovo membro.
5. Digita l'indirizzo email del nuovo membro, scegli il ruolo per questo membro, quindi scegli Invita. Per ulteriori informazioni sui ruoli, consulta [Concessione dell'accesso con ruoli utente](#).

Per invitare un membro al tuo progetto dalla pagina di panoramica del progetto

1. Vai al tuo progetto.

 Tip

Puoi scegliere quale progetto visualizzare nella barra di navigazione in alto.

2. Scegli il pulsante Membri +.
3. Digita l'indirizzo email del nuovo membro, scegli il ruolo per questo membro, quindi scegli Invita. Per ulteriori informazioni sui ruoli, consulta [Concessione dell'accesso con ruoli utente](#).

Annullamento di un invito

Se hai inviato di recente un invito, puoi annullarlo purché l'invito non sia stato ancora accettato.

Per gestire gli inviti ai progetti, devi avere il ruolo di amministratore del progetto o amministratore dello spazio.

Per annullare l'invito di un membro del progetto

1. Vai al progetto a cui hai inviato un invito che desideri annullare.
2. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Visualizza la scheda Membri e verifica che lo stato del membro sia Invitato.

 Note

Puoi annullare solo un invito che non è stato ancora accettato.

4. Scegli l'opzione accanto alla riga con il membro invitato, quindi scegli Annulla invito.
5. Viene visualizzata una finestra di conferma. Scegli Annulla invito per confermare.

Rimuovere un utente dal progetto

Puoi usare la console per rimuovere un utente dal tuo progetto.

Per rimuovere un utente dal progetto, è necessario accedere con il ruolo di amministratore del progetto o amministratore dello spazio.

 Note

La rimozione di un utente da tutti i progetti all'interno di uno spazio rimuove automaticamente l'utente da quello spazio.

Per rimuovere un utente da un progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Passa allo spazio contenente il progetto che desideri visualizzare. In Progetti, scegli il tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Scegli la scheda Membri.
5. Scegli il selettore accanto al profilo che desideri rimuovere, quindi scegli Rimuovi.
6. Conferma che desideri rimuovere l'utente, quindi scegli Rimuovi.

Accettare o rifiutare un invito per un progetto

Potresti ricevere un'e-mail di invito a partecipare a un CodeCatalyst progetto Amazon. È possibile accettare o rifiutare l'invito.

Per accettare o rifiutare un invito

1. Apri l'e-mail di invito.
2. Scegli il link del progetto nell'email.
3. Scegli Accetta o Rifiuta.

Se scegli Rifiuta, viene inviata un'e-mail all'account di gestione del progetto per informarlo che hai rifiutato l'invito.

Consentire l'accesso al progetto tramite team

Dopo aver creato un progetto, puoi aggiungere team. I team consentono di raggruppare gli utenti in modo che possano condividere le autorizzazioni e gestire progetti, assegnare tracciamento, ruoli e risorse CodeCatalyst come membri del progetto e dello spazio.

È necessario avere il ruolo di amministratore del progetto per gestire i team del progetto.

I team vengono gestiti anche a livello di spazio in CodeCatalyst. Per ulteriori informazioni sui team in Spaces, consulta [Consentire l'accesso allo spazio tramite team](#).

Argomenti

- [Aggiungere un team a un progetto](#)
- [Assegnazione di ruoli di progetto a un team](#)
- [Rimuovere un ruolo di progetto per un team](#)

Aggiungere un team a un progetto

Puoi gestire team in cui i membri del team possono accedere alle risorse del tuo progetto.

Nella pagina dei membri per gli utenti dello spazio e del progetto, gli utenti possono avere più ruoli. Gli utenti con più ruoli mostreranno un indicatore quando hanno più ruoli e verranno visualizzati per primi con il ruolo con il maggior numero di autorizzazioni.

Per aggiungere un team

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto. Scegli Impostazioni del progetto, quindi scegli Teams.
3. Scegli Aggiungi team.
4. In Team, scegli una squadra dall'elenco delle squadre disponibili.
5. In Ruolo del progetto, scegli un ruolo dall'elenco dei ruoli di progetto disponibili in CodeCatalyst.
 - Amministratore del progetto: per i dettagli, consulta [Ruolo di amministratore del progetto](#).
 - Collaboratore: per i dettagli, vedere [Ruolo di collaboratore](#).
 - Revisore: per i dettagli, vedere. [Ruolo di revisore](#)
 - Sola lettura: per i dettagli, vedere [Ruolo di sola lettura](#).
6. Scegli Aggiungi team.

Assegnazione di ruoli di progetto a un team

Un team può disporre di autorizzazioni di ruolo, ad esempio Power user, in uno spazio. Puoi modificare il ruolo spaziale di un team, ma tieni presente che tutti i membri del team ereditano tali autorizzazioni.

Per aggiungere o modificare un ruolo nel progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni del progetto, quindi scegli Teams.
3. Per modificare un ruolo, scegli il selettore accanto al team in questo elenco, quindi scegli Cambia ruolo. Per aggiungere un ruolo, scegli Aggiungi ruolo di progetto. In Project, scegli il progetto che desideri aggiungere e in Ruolo scegli il ruolo. Scegli uno dei ruoli di progetto disponibili:
 - Amministratore del progetto: per i dettagli, consulta [Ruolo di amministratore del progetto](#).
 - Collaboratore - Per i dettagli, vedere [Ruolo di collaboratore](#).
 - Revisore - Per i dettagli, vedi. [Ruolo di revisore](#)
 - Sola lettura - Per i dettagli, vedi [Ruolo di sola lettura](#).
4. Scegli Save (Salva).

Rimuovere un ruolo di progetto per un team

In CodeCatalyst, puoi visualizzare i ruoli del progetto per il tuo team. Puoi anche visualizzare i membri di un team. Puoi rimuovere il ruolo di progetto per un team.

Per rimuovere un ruolo di progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo spazio. Scegli Impostazioni del progetto, quindi scegli Teams.
3. Scegli la scheda Ruoli del progetto.
4. Scegli il ruolo che desideri rimuovere.



Important

La rimozione di un ruolo da un team rimuove le autorizzazioni associate per tutti gli utenti del team.

5. Scegli Save (Salva).

Consentire l'accesso al progetto per le risorse della macchina

Le risorse della macchina sono risorse specifiche in cui CodeCatalyst vengono concesse le autorizzazioni per progetti o spazi in CodeCatalyst.

Note

Il termine risorsa macchina non si riferisce a un'infrastruttura cloud come un' EC2 istanza, ma si riferisce invece a un blueprint o a una risorsa di flusso di lavoro con autorizzazioni per uno spazio o un progetto.

Un esempio di utilizzo delle risorse delle macchine nei progetti include l'abilitazione di una risorsa blueprint per accedere a un progetto per conto dell'utente.

Una risorsa macchina rappresenta l'identità dell'utente rispetto alla risorsa autorizzata quando si accede CodeCatalyst tramite SSO. Le risorse della macchina vengono utilizzate per concedere le autorizzazioni alle risorse del progetto, come blueprint e flussi di lavoro. È possibile visualizzare le risorse macchina del progetto e scegliere di abilitare o disabilitare le risorse macchina per il progetto. Ad esempio, potresti voler disabilitare una risorsa macchina per gestire l'accesso e riattivarla in un secondo momento.

Queste operazioni sono disponibili per le risorse della macchina nei casi in cui una risorsa della macchina debba essere revocata o disabilitata. Ad esempio, se sospetti che le credenziali possano essere state compromesse, puoi disabilitare la risorsa della macchina. In genere, non è necessario utilizzare queste operazioni.

È necessario disporre del ruolo di amministratore dello spazio o del ruolo di amministratore del progetto per visualizzare questa pagina e gestire le risorse della macchina a livello di progetto.

Le risorse della macchina vengono gestite anche a livello di spazio in CodeCatalyst. Per ulteriori informazioni sui team che operano in spazi/progetti, consulta [Consentire l'accesso allo spazio per le risorse della macchina](#)

Argomenti

- [Visualizzazione dell'accesso al progetto per le risorse della macchina](#)

- [Disabilitazione dell'accesso al progetto per le risorse della macchina](#)
- [Abilitazione dell'accesso al progetto per le risorse della macchina](#)

Visualizzazione dell'accesso al progetto per le risorse della macchina

È possibile visualizzare un elenco delle risorse delle macchine utilizzate nel progetto.

È necessario avere il ruolo di amministratore dello spazio o il ruolo di amministratore del progetto.

Per visualizzare le risorse della macchina

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto, quindi scegli Impostazioni progetto. Scegli Risorse della macchina.
3. Nel menu a discesa, scegli l'azione Workflow per visualizzare solo le risorse della macchina per i flussi di lavoro. Scegli Blueprint per visualizzare solo le risorse della macchina per i blueprint.

Puoi anche filtrare in base a un nome utilizzando il campo Filtro.

Disabilitazione dell'accesso al progetto per le risorse della macchina

È possibile scegliere di disabilitare le risorse della macchina utilizzate nel progetto.

Important

La disabilitazione delle risorse della macchina rimuoverà tutte le autorizzazioni per tutti i blueprint o i flussi di lavoro associati nello spazio.

È necessario avere il ruolo di amministratore dello spazio o il ruolo di amministratore del progetto.

Per disabilitare le risorse della macchina

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto, quindi scegli Impostazioni progetto. Scegli Risorse della macchina.
3. Scegliere una delle seguenti opzioni.

 Important

La disabilitazione delle risorse della macchina rimuoverà tutte le autorizzazioni per tutti i blueprint o i flussi di lavoro associati nello spazio.

- Per disabilitarle singolarmente, scegli il selettore accanto a una o più risorse della macchina che desideri disabilitare. Scegli Disabilita, quindi scegli Questa risorsa.
- Per disattivare tutte le risorse, scegli Disabilita, quindi scegli Tutte le risorse.
- Per disabilitare tutte le azioni del flusso di lavoro, scegli Disabilita, quindi scegli Tutte le azioni del flusso di lavoro.
- Per disabilitare tutti i blueprint, scegli Disabilita, quindi scegli Tutti i blueprint.

Abilitazione dell'accesso al progetto per le risorse della macchina

È possibile scegliere di abilitare le risorse della macchina utilizzate nel progetto e che sono state disabilitate.

È necessario avere il ruolo di amministratore dello spazio o il ruolo di amministratore del progetto.

Per abilitare le risorse della macchina

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto, quindi scegli Impostazioni progetto. Scegli Risorse della macchina.
3. Scegliere una delle seguenti opzioni.
 - Per abilitarle singolarmente, scegli il selettore accanto a una o più risorse della macchina che desideri abilitare. Scegli Abilita, quindi scegli Questa risorsa.
 - Per abilitare tutte le risorse, scegli Abilita, quindi scegli Tutte le risorse.
 - Per abilitare tutte le azioni del flusso di lavoro, scegli Abilita, quindi scegli Tutte le azioni del flusso di lavoro.
 - Per abilitare tutti i blueprint, scegli Abilita, quindi scegli Tutti i blueprint.

Quote per progetti

La tabella seguente descrive le quote e i limiti per i progetti in Amazon CodeCatalyst. Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Numero massimo di progetti per spazio	100
Numero massimo di progetti a cui un utente può appartenere	1.000
Numero massimo di membri che possono appartenere a un progetto.	10.000
Nomi di progetto	I nomi dei progetti devono essere univoci all'interno di uno spazio. I nomi devono avere una lunghezza compresa tra 3 e 63 caratteri . I nomi rispettano la distinzione tra lettere maiuscole e minuscole. I nomi dei progetti devono iniziare con un carattere alfanumerico. Caratteri validi: A-Z, a-z, 0-9, spazi e., _ (trattino basso) - (trattino) I nomi dei progetti non possono contenere nessuno dei seguenti caratteri: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Descrizioni del progetto	Le descrizioni dei progetti possono contenere fino a 200 caratteri. Caratteri validi: A-Z, a-z, 0-9, spazi e., _ (trattino basso) - (trattino). Le descrizioni dei progetti sono facoltative.

Invio di notifiche da CodeCatalyst

Puoi impostare notifiche per monitorare i tuoi progetti e le tue risorse in CodeCatalyst. Gli utenti possono scegliere gli eventi del progetto per i quali desiderano ricevere e-mail in qualsiasi progetto di cui sono membri. Puoi anche scegliere di configurare le notifiche inviate a un intero team in

un'applicazione di messaggistica del team, come Slack, configurando l'accesso tra CodeCatalyst uno spazio e un'area di lavoro Slack e quindi configurando le notifiche per l'invio di un progetto a uno o più canali in quell'area di lavoro Slack. Dopo aver configurato l'accesso tra uno CodeCatalyst spazio e uno spazio di lavoro Slack, i membri del progetto avranno anche la possibilità di aggiungere il proprio membro Slack in IDs modo da poter ricevere notifiche direttamente sugli CodeCatalyst eventi negli spazi di lavoro e nei canali Slack collegati.

Note

L'insieme di eventi del progetto che possono essere inviati a Slack non è lo stesso insieme di eventi per i quali gli utenti possono scegliere di ricevere notifiche via e-mail.

Argomenti

- [Come funzionano le notifiche?](#)
- [Guida introduttiva alle notifiche di Slack](#)
- [Invio di notifiche via email e su Slack da CodeCatalyst](#)

Come funzionano le notifiche?

Puoi configurare il tuo progetto per fornire notifiche all'applicazione di messaggistica del tuo team, come Slack.

Quali autorizzazioni sono necessarie per le notifiche?

Qualsiasi membro del progetto può configurare, visualizzare, aggiornare o eliminare le impostazioni di notifica per un canale in CodeCatalyst. Tuttavia, solo gli utenti con il ruolo di amministratore dello spazio possono aggiungere o eliminare aree di lavoro Slack. Tutti gli utenti possono configurare gli eventi del progetto a cui desiderano ricevere e-mail per i progetti a cui appartengono. CodeCatalyst

Per quali CodeCatalyst eventi posso configurare le notifiche?

Puoi configurare l'invio CodeCatalyst di notifiche a uno o più canali Slack sugli eventi del flusso di lavoro. Una volta configurate le notifiche tra un CodeCatalyst progetto e Slack, gli utenti del progetto possono scegliere di aggiungere il proprio ID membro Slack per ricevere messaggi diretti sugli eventi nei canali Slack. CodeCatalyst Gli utenti che aggiungono un membro Slack IDs riceveranno menzioni

dirette IDs nei canali Slack configurati per i loro progetti, contribuendo a sensibilizzare l'opinione pubblica sugli eventi a cui tengono.

Puoi anche scegliere per quali eventi vuoi ricevere email. Queste e-mail vengono inviate all'indirizzo e-mail configurato per il tuo ID AWS Builder.

Come vengono visualizzate le notifiche?

Puoi configurare l'invio CodeCatalyst di notifiche a uno o più canali Slack. Devi autorizzarti a concedere le autorizzazioni CodeCatalyst per accedere al tuo spazio di lavoro Slack. Una volta fornita l'autorizzazione, CodeCatalyst puoi inviare notifiche ai canali Slack che configuri. Se un membro del progetto sceglie di aggiungere il proprio ID membro Slack, può ricevere menzioni sugli CodeCatalyst eventi nei canali Slack configurati per quel progetto.

Come faccio a configurare le notifiche?

Le notifiche e-mail sono configurate come parte di CodeCatalyst. Gli utenti del progetto possono scegliere per quali eventi ricevere e-mail nella pagina *Le mie impostazioni*.

Per configurare le notifiche Slack per le risorse del progetto, devi completare le seguenti attività di alto livello.

Per impostare le notifiche (attività di alto livello)

1. In CodeCatalyst, configuri una connessione tra CodeCatalyst e un client di messaggistica, come Slack. Una volta connesso, uno spazio di lavoro Slack sarà disponibile per tutti i progetti presenti nello spazio.

Note

Solo gli utenti con un ruolo di amministratore di Space possono aggiungere o eliminare un'area di lavoro Slack.

2. Nel tuo progetto in CodeCatalyst, aggiungi il canale in cui desideri che il team riceva le notifiche.
3. In CodeCatalyst, attivi le notifiche per vari eventi, ad esempio un errore di esecuzione del flusso di lavoro, e specifichi il canale in cui desideri che vengano inviate.

Per informazioni dettagliate sulle fasi, consulta [Guida introduttiva alle notifiche di Slack](#).

Una volta configurate le notifiche tra uno CodeCatalyst spazio e Slack, gli utenti possono scegliere di aggiungere il proprio membro Slack IDs per ricevere messaggi diretti sugli CodeCatalyst eventi nei canali Slack configurati per i loro progetti,

Guida introduttiva alle notifiche di Slack

Dopo aver creato un progetto, puoi configurare le notifiche di Slack che aiutano il tuo team a monitorare le risorse del progetto.

Questi passaggi ti guidano nella configurazione delle notifiche di Slack per la prima volta.

CodeCatalyst Se hai già configurato le notifiche, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).

Note

L'insieme di eventi del progetto che possono essere inviati ai canali di notifica non è lo stesso insieme di eventi per i quali gli utenti possono scegliere di ricevere notifiche via e-mail. Per ulteriori informazioni, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).

Argomenti

- [Prerequisiti](#)
- [Passaggio 1: Connettiti CodeCatalyst al tuo spazio di lavoro Slack](#)
- [Passaggio 2: aggiungi il tuo canale Slack a CodeCatalyst](#)
- [Passaggio 3: prova le notifiche da inviare CodeCatalyst a Slack](#)
- [Fase 4: Fasi successive](#)

Prerequisiti

Prima di iniziare, avrai bisogno di:

- Uno CodeCatalyst spazio. Per informazioni sulla creazione di uno CodeCatalyst spazio e sul primo accesso, consulta [Configura e accedi a CodeCatalyst](#).
- Un CodeCatalyst progetto. Per ulteriori informazioni, consulta [Creare un progetto](#).
- Un CodeCatalyst account con il ruolo di amministratore del progetto o amministratore dello spazio. Per ulteriori informazioni, consulta [Concessione dell'accesso con ruoli utente](#).

- Un account Slack e un'area di lavoro Slack a cui è possibile accedere da CodeCatalyst
- Un canale Slack dove CodeCatalyst inviare notifiche. Il canale può essere pubblico o privato.

Passaggio 1: Connettiti CodeCatalyst al tuo spazio di lavoro Slack

Solo gli utenti con il ruolo di amministratore di Space possono aggiungere o eliminare aree di lavoro Slack. L'aggiunta o l'eliminazione di un'area di lavoro Slack influisce su tutti i progetti presenti nello spazio. Per stabilire la connessione tra Slack CodeCatalyst e Slack, CodeCatalyst esegue un handshake di OAuth autenticazione sicuro con il tuo spazio di lavoro Slack.

Usa le seguenti istruzioni per connetterti al tuo spazio di lavoro CodeCatalyst Slack.

Note

Questa operazione deve essere eseguita solo una volta per ogni area di lavoro Slack. Puoi quindi impostare le notifiche tramite il canale Slack.

Per connetterti CodeCatalyst al tuo spazio di lavoro Slack

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Scegli la scheda Notifiche.
5. Scegli Configura notifiche.
6. Scegli Connect to Slack workspace.
7. Leggi il contenuto della finestra di dialogo, quindi scegli Connect to Slack workspace.
8. Nel messaggio Amazon Q Developer nelle applicazioni di chat:
 - a. In alto a destra, scegli l'area di lavoro Slack che contiene il tuo canale.
 - b. Scegli Permetti.

Verrai reindirizzato alla console. CodeCatalyst

9. Continua su [Passaggio 2: aggiungi il tuo canale Slack a CodeCatalyst](#).

Passaggio 2: aggiungi il tuo canale Slack a CodeCatalyst

Ti serve l'ID del canale Slack a cui aggiungere il tuo canale. CodeCatalyst

Per ottenere l'ID del tuo canale Slack

1. Accedi a Slack. Per ulteriori informazioni, consulta [Accedere a Slack](#).
2. Vai all'area di lavoro di Slack che contiene il canale in cui desideri che vengano inviate le notifiche. Per maggiori informazioni, consulta [Passare da un'area di lavoro Slack all'altra o Accedere ad aree di lavoro Slack aggiuntive](#).
3. Nel riquadro di navigazione, apri il menu contestuale (clic con il tasto destro) del canale in cui desideri inviare le notifiche e scegli Apri dettagli del canale.

L'ID del canale viene visualizzato nella parte inferiore della finestra di dialogo.

4. Copia il valore dell'ID del canale. Questo valore servirà nella fase successiva.

Utilizzando l'ID del canale che hai appena copiato, ora puoi connettere il tuo canale Slack a CodeCatalyst

Per aggiungere il tuo canale Slack a CodeCatalyst

1. Prima di iniziare, se il tuo canale Slack è privato, aggiungi l'app Amazon Q Developer nelle applicazioni di chat al canale come segue:
 - a. Nella casella dei messaggi del tuo canale Slack, inserisci **@aws** e scegli l'app aws dalla finestra di dialogo.
 - b. Premere Invio.

Viene visualizzato un messaggio Slackbot che indica che Amazon Q Developer nelle applicazioni di chat non è nel canale privato.
 - c. Scegli Invitali per invitare Amazon Q Developer nelle applicazioni di chat sul canale.
2. Nella CodeCatalyst console, scegli Avanti.
3. In Channel ID, incolla l'ID del canale Slack che hai ottenuto in precedenza.
4. Nel nome del canale, inserisci un nome. Ti consigliamo di utilizzare il nome del canale Slack.
5. Scegli Next (Successivo).
6. In Seleziona gli eventi di notifica, scegli il tipo di evento per cui desideri ricevere notifiche.
7. Scegli Fine.

Passaggio 3: prova le notifiche da inviare CodeCatalyst a Slack

Dopo aver configurato il progetto per inviare notifiche sullo stato del flusso di lavoro, puoi visualizzare le notifiche in Slack.

Per visualizzare le notifiche in Slack

1. Nel tuo CodeCatalyst progetto, [avvia un flusso di lavoro manualmente](#) per completare l'esecuzione di un flusso di lavoro e ricevi una notifica di stato al termine dell'esecuzione.
2. In Slack, visualizza il canale che hai configurato per le notifiche. Le tue notifiche mostrano lo stato più recente di ogni workflow eseguito e se è fallito o ha avuto esito positivo.

Fase 4: Fasi successive

Una volta configurato uno spazio di lavoro Slack per il tuo CodeCatalyst spazio, puoi aggiungere altri canali Slack, i CodeCatalyst progetti esistenti e aggiungerli a nuovi progetti dopo averli creati. Puoi anche far sapere agli utenti del progetto che possono configurare notifiche Slack personali per i loro membri IDs Slack e configurare gli eventi per i quali riceveranno email. Per ulteriori informazioni, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).

Invio di notifiche via email e su Slack da CodeCatalyst

È possibile CodeCatalyst configurare l'invio di notifiche sugli eventi che si verificano nel progetto. CodeCatalyst può inviare notifiche a client di messaggistica come i canali Slack. L' CodeCatalyst invio di messaggi ai canali Slack aiuta a garantire che tutto il team sia a conoscenza di eventi importanti, come gli errori del flusso di lavoro. Facoltativamente, puoi scegliere di inserire CodeCatalyst @mention you nei messaggi Slack che invia in modo da ricevere un messaggio diretto (DM) corrispondente.

CodeCatalyst può anche inviarti notifiche direttamente tramite e-mail. Verranno inviate notifiche e-mail sugli eventi di qualsiasi progetto a cui sei membro. Queste e-mail verranno inviate all'indirizzo e-mail configurato nel tuo ID AWS Builder.

Note

Gli eventi che possono essere inviati ai canali Slack possono essere diversi da quelli inviati tramite e-mail.

Argomenti

- [Configurazione delle notifiche e-mail](#)
- [Invio di notifiche ai canali Slack](#)
- [Configurazione dei messaggi diretti Slack](#)
- [Modifica delle notifiche per un canale di notifica](#)
- [Rimuovere un canale](#)

Configurazione delle notifiche e-mail

Puoi scegliere di ricevere notifiche e-mail sugli eventi di qualsiasi progetto di cui sei membro. Queste e-mail verranno inviate all'indirizzo e-mail configurato nel tuo ID AWS Builder. Per impostazione predefinita, riceverai e-mail su tutti gli eventi del progetto per i quali è possibile inviare e-mail.

Per configurare le notifiche e-mail per gli eventi del progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. In Notifiche e-mail, trova il progetto nell'elenco in cui desideri configurare le notifiche e-mail e scegli Modifica.
4. Seleziona gli eventi per i quali desideri ricevere e-mail, quindi scegli Salva.

Invio di notifiche ai canali Slack

Puoi configurare l'invio CodeCatalyst di notifiche sugli eventi del progetto ai canali Slack del tuo team. In questo modo, puoi contribuire a garantire che l'intero team sia a conoscenza di eventi importanti, ad esempio quando l'esecuzione di un flusso di lavoro fallisce.

 Note

Qualsiasi membro di un progetto può gestire le notifiche inviate ai canali di quel progetto. Tuttavia, solo gli utenti con il ruolo di amministratore dello spazio possono aggiungere o eliminare aree di lavoro Slack.

Utilizza le seguenti istruzioni per aggiungere un canale Slack a cui verranno inviate le notifiche.

Per aggiungere un canale Slack per le notifiche

1. Se stai aggiungendo il tuo primo canale Slack, guarda invece. [Guida introduttiva alle notifiche di Slack](#)

Dopo aver configurato il primo canale, torna a questa procedura per configurare canali aggiuntivi.

2. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Vai al tuo progetto.
4. Nel riquadro di navigazione, scegli Impostazioni del progetto.
5. Scegli la scheda Notifiche.
6. Scegli Add channel (Aggiungi canale).
7. Scegli Scegli l'area di lavoro, quindi seleziona l'area di lavoro Slack che contiene il canale a cui desideri inviare le notifiche.

Se il tuo spazio di lavoro Slack non è nell'elenco, puoi aggiungerlo seguendo le istruzioni in. [Guida introduttiva alle notifiche di Slack](#)

8. Prima di inserire un ID di canale, se il canale Slack che desideri aggiungere è privato, completa questi passaggi:
 - a. Nella casella dei messaggi del tuo canale Slack, inserisci **@aws** e scegli l'app aws dal pop-up.
 - b. Premere Invio.

Viene visualizzato un messaggio Slackbot che indica che Amazon Q Developer nelle applicazioni di chat non è nel canale privato.

- c. Scegli Invitali per invitare Amazon Q Developer nelle applicazioni di chat sul canale.

9. Nel campo ID CodeCatalyst del canale, inserisci l'ID del canale Slack. Per trovare l'ID, vai su Slack e, nel riquadro di navigazione, fai clic con il pulsante destro del mouse sul canale e scegli Apri dettagli del canale.

L'ID del canale viene visualizzato nella parte inferiore della finestra di dialogo.

10. In Nome del canale, inserisci un nome. Ti consigliamo di utilizzare il nome del canale Slack.
11. In Seleziona gli eventi di notifica, scegli il tipo di evento per cui desideri ricevere notifiche.
12. Scegli Aggiungi.

Configurazione dei messaggi diretti Slack

Se un CodeCatalyst progetto è stato configurato per [inviare notifiche a un canale Slack](#), tali notifiche possono essere inviate anche come messaggi diretti (DMs). Ricevere notifiche direttamente a te DMs può aiutarti a sensibilizzare l'opinione pubblica sugli eventi che accadono nei progetti in cui svolgi un ruolo. Per abilitarlo DMs, devi aggiungere il tuo ID membro Slack a CodeCatalyst.

Per configurare i messaggi diretti di Slack

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. Nelle notifiche personali di Slack, scegli Connetti ID Slack, quindi scegli Connetti all'area di lavoro Slack. Si aprirà una finestra separata.

Tip

Questa opzione non è configurabile a meno che un utente con il ruolo di amministratore dello spazio non abbia aggiunto un'area di lavoro Slack per il tuo spazio. CodeCatalyst Per ulteriori informazioni, consulta [Guida introduttiva alle notifiche di Slack](#) e [Invio di notifiche ai canali Slack](#).

4. Nella finestra di richiesta delle autorizzazioni, assicurati che il nome dell'area di lavoro corrisponda all'area di lavoro Slack configurata per lo spazio. CodeCatalyst Scegli Consenti per consentire agli sviluppatori Amazon Q nelle applicazioni di chat di accedere all'area di lavoro. La finestra si chiuderà e l'area di lavoro Slack mostrerà lo stato della connessione come Connect.

 Tip

Se lo stato della connessione non cambia, controlla se si è verificato un errore durante la connessione dell'area di lavoro Slack. Potrebbe essere necessario scorrere verso l'alto per visualizzare l'errore.

5. Per non ricevere più notifiche personali su Slack, scegli l'area di lavoro Slack connessa, quindi scegli Disconnetti l'ID Slack.

Modifica delle notifiche per un canale di notifica

Puoi modificare i canali a cui vanno indirizzate le notifiche e puoi disattivare del tutto notifiche specifiche.

Per modificare le notifiche

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto.
3. Nel riquadro di navigazione, scegli Impostazioni del progetto.
4. Scegli la scheda Notifiche.
5. Scegli Modifica notifiche.
6. Esegui una di queste operazioni:
 - Per inviare una notifica a un canale specifico, scegli il canale dall'elenco a discesa.
 - Per disattivare una notifica a livello globale, scegli l'interruttore accanto alla notifica.
 - Per interrompere l'invio di una notifica a un canale specifico, scegli la X sul canale.
7. Seleziona Salva.

Rimuovere un canale

Puoi rimuovere un canale Slack da Amazon CodeCatalyst. Rimuovendo un canale Slack, le notifiche relative a un CodeCatalyst progetto scelto non verranno più inviate al canale.

Per rimuovere un canale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Nella pagina delle impostazioni del progetto, scegli la scheda Notifiche.
4. Scegli l'indicatore accanto al canale che desideri rimuovere, quindi scegli Rimuovi canale. Scegli Ok nella finestra di conferma.

Configura CodeCatalyst progetti con progetti

I blueprint sono generatori di codice arbitrari che rappresentano un componente architettonico di un progetto. CodeCatalyst Il componente può essere costituito da qualsiasi cosa, da un flusso di lavoro in un singolo file all'intero progetto completo di codice di esempio. I blueprint accettano un insieme arbitrario di opzioni e le utilizzano per generare un set arbitrario di codice di output che viene inoltrato a un progetto. Man mano che il blueprint viene aggiornato con le migliori pratiche più recenti o nuove opzioni, può rigenerare le parti pertinenti della base di codice in progetti che contengono quel blueprint.

Puoi utilizzare un CodeCatalyst blueprint Amazon per creare un progetto completo con un repository di sorgenti, codice sorgente di esempio, CI/CD flussi di lavoro, report di compilazione e test e strumenti integrati di tracciamento dei problemi. Un CodeCatalyst blueprint genera risorse e codice sorgente in base a un set di parametri di configurazione. Quando si utilizza un blueprint CodeCatalyst gestito, il blueprint scelto determina quali risorse vengono aggiunte al progetto, nonché gli strumenti per la CodeCatalyst creazione o la configurazione, in modo da poter tenere traccia e utilizzare le risorse del progetto. In qualità di utente del blueprint, puoi creare un progetto con un blueprint o aggiungerlo a un progetto esistente. CodeCatalyst È possibile aggiungere più blueprint al progetto e ognuno di essi può essere applicato come componente indipendente. Ad esempio, è possibile creare un progetto con un blueprint di applicazione Web e quindi aggiungere un blueprint di sicurezza in un secondo momento. Quando uno dei blueprint viene aggiornato, è possibile incorporare le modifiche o le correzioni nel progetto tramite la gestione del ciclo di vita. Per ulteriori informazioni, consultare [Creazione di un progetto completo con CodeCatalyst progetti](#) e [Utilizzo della gestione del ciclo di vita come utente del blueprint](#).

In qualità di autore di blueprint, puoi anche creare e pubblicare blueprint personalizzati per consentire ai membri del tuo CodeCatalyst spazio di utilizzare le risorse del progetto. I progetti personalizzati possono essere sviluppati per soddisfare esigenze specifiche per i progetti del tuo spazio. Dopo aver aggiunto un blueprint personalizzato al catalogo dei blueprint del tuo spazio, puoi gestirlo e continuare ad apportare aggiornamenti in modo che i progetti del tuo spazio rimangano aggiornati con le migliori pratiche più recenti. Per ulteriori informazioni, consulta [Standardizzazione dei progetti con progetti personalizzati in CodeCatalyst](#). [Per visualizzare l'SDK dei blueprint e i blueprint di esempio, consulta l'archivio open source. GitHub](#)

È possibile che la standardizzazione e le migliori pratiche siano già in atto. Invece di creare e sviluppare un blueprint personalizzato da zero, puoi scegliere di convertire un repository di sorgenti

esistente con codice sorgente in un blueprint personalizzato. Per ulteriori informazioni, consulta [Conversione degli archivi di origine in blueprint personalizzati](#).

Argomenti

- [Creazione di un progetto con un blueprint](#)
- [Aggiungere un modello in un progetto per integrare le risorse](#)
- [Dissociazione di un blueprint da un progetto per interrompere gli aggiornamenti](#)
- [Modifica delle versioni del blueprint in un progetto](#)
- [Modifica della descrizione di un blueprint in un progetto](#)
- [Utilizzo della gestione del ciclo di vita come utente del blueprint](#)
- [Creazione di un progetto completo con CodeCatalyst progetti](#)
- [Standardizzazione dei progetti con progetti personalizzati in CodeCatalyst](#)
- [Quote per i progetti in CodeCatalyst](#)

Creazione di un progetto con un blueprint

Puoi creare rapidamente un progetto utilizzando un blueprint dal catalogo Amazon CodeCatalyst Blueprints o dal catalogo spaziale del tuo team con blueprint personalizzati. A seconda del blueprint, il progetto viene creato con risorse specifiche. Puoi anche collaborare con Amazon Q, un assistente AI generativo, durante la creazione di nuovi progetti o l'aggiunta di componenti a progetti esistenti. Puoi fornire ad Amazon Q i requisiti per il tuo progetto interagendo con esso in un'interfaccia simile a una chat. In base ai tuoi requisiti, Amazon Q suggerisce un modello e delinea anche i requisiti che non possono essere soddisfatti. Puoi quindi procedere con il suggerimento di Amazon Q se sei soddisfatto e creerà le risorse necessarie, come un repository di sorgenti con codice per le tue esigenze. Per ulteriori informazioni, consultare [Creare un progetto con un blueprint](#), [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#) e [Creazione di un progetto completo con CodeCatalyst progetti](#).

Dopo aver creato un progetto, puoi aggiungere altri progetti al tuo CodeCatalyst progetto dal CodeCatalyst catalogo o dal catalogo del tuo spazio con progetti personalizzati. I blueprint rappresentano componenti architettonici, quindi è possibile utilizzare più blueprint insieme nel progetto per incorporare le best practice del team. Questo ti dà anche la possibilità di assicurarti che il tuo progetto sia aggiornato con le ultime modifiche ai componenti in evoluzione. Per ulteriori informazioni sull'utilizzo dei blueprint nel progetto, consulta [Utilizzo della gestione del ciclo di vita come utente del blueprint](#)

Aggiungere un modello in un progetto per integrare le risorse

Puoi aggiungere più blueprint in un progetto per incorporare componenti funzionali, risorse e governance. I tuoi progetti possono supportare vari elementi gestiti in modo indipendente in blueprint separati. L'aggiunta di blueprint a un progetto riduce la necessità di creare manualmente risorse e rendere funzionali i componenti software. I tuoi progetti possono inoltre rimanere aggiornati man mano che i requisiti evolvono. Per ulteriori informazioni sull'aggiunta di blueprint nel progetto, consulta [Utilizzo della gestione del ciclo di vita come utente del blueprint](#)

Durante la configurazione dei dettagli di un blueprint, puoi anche scegliere di archiviare il codice sorgente del blueprint in un repository preferito di terze parti, dove puoi comunque gestire il blueprint e utilizzare le funzionalità di gestione del ciclo di vita per mantenere aggiornato il progetto. Per ulteriori informazioni, consultare [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#) e [Utilizzo della gestione del ciclo di vita come utente del blueprint](#).

Important

Per aggiungere un blueprint al CodeCatalyst progetto, è necessario accedere con un account che abbia il ruolo di amministratore dello spazio, Power user o amministratore del progetto nello spazio.

Tip

Dopo aver aggiunto un blueprint al progetto, puoi configurare le notifiche e-mail e Slack per fornire aggiornamenti per le ultime modifiche al blueprint. Per ulteriori informazioni, consulta [Invio di notifiche da CodeCatalyst](#).

Per aggiungere un blueprint al tuo progetto

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio, quindi scegli il progetto a cui desideri aggiungere un blueprint.
3. Nel riquadro di navigazione, scegli Blueprint, quindi scegli Aggiungi blueprint.

 Tip

Puoi scegliere di aggiungere un blueprint fornendo ad Amazon Q i requisiti del tuo progetto in modo che Amazon Q ti suggerisca un blueprint. Per ulteriori informazioni, consultare [Utilizzo di Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di funzionalità](#) e [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#). Questa funzionalità è disponibile solo nella regione Stati Uniti occidentali (Oregon).

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

4. Scegli un blueprint dalla scheda Blueprint o un CodeCatalyst blueprint personalizzato dalla scheda Space blueprint, quindi scegli Avanti.
5. In Dettagli del blueprint, scegli una versione del blueprint dal menu a discesa della versione di Target. La versione più recente del catalogo viene selezionata automaticamente.
6. (Facoltativo) Per impostazione predefinita, il codice sorgente creato dal blueprint viene archiviato in un CodeCatalyst repository. In alternativa, è possibile scegliere di archiviare il codice sorgente del blueprint in un repository di terze parti. Per ulteriori informazioni, consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#).

Effettua una delle seguenti operazioni a seconda del provider di repository di terze parti che desideri utilizzare:

- GitHub repository: Connect a un GitHub account.

Scegli il menu a discesa Avanzato, scegli GitHub come provider del repository, quindi scegli l'GitHub account in cui desideri archiviare il codice sorgente creato dal blueprint.

 Note

Se utilizzi una connessione a un GitHub account, devi creare una connessione personale per stabilire una mappatura delle identità tra la tua identità e la tua CodeCatalyst identità. Per ulteriori informazioni, consultare [Connessioni personali](#) e [Accesso alle GitHub risorse con connessioni personali](#).

- Archivi Bitbucket: Connect uno spazio di lavoro Bitbucket.

Scegli il menu a discesa Avanzato, scegli Bitbucket come provider di repository, quindi scegli l'area di lavoro Bitbucket in cui desideri archiviare il codice sorgente creato dal blueprint.

- GitLab repository: Connect a un GitLab utente.

Scegli il menu a discesa Avanzato, scegli GitLab come provider del repository, quindi scegli l'GitLab utente in cui desideri archiviare il codice sorgente creato dal blueprint.

7. In Configura blueprint, configura i parametri del blueprint. A seconda del blueprint, potresti avere la possibilità di assegnare un nome al repository di origine.
8. Esamina le differenze tra la versione corrente del blueprint e la versione aggiornata. La differenza visualizzata in una pull request mostra le modifiche tra la versione corrente e la versione più recente, che è la versione desiderata al momento della creazione della pull request. Se non viene visualizzata alcuna modifica, le versioni potrebbero essere identiche oppure potresti aver scelto la stessa versione sia per la versione corrente che per quella desiderata.
9. Quando sei sicuro che la pull request contenga il codice e le modifiche che desideri rivedere, scegli Aggiungi progetto. Dopo aver creato la pull request, puoi aggiungere commenti. I commenti possono essere aggiunti alla pull request o alle singole righe dei file, nonché alla pull request complessiva. È possibile aggiungere collegamenti a risorse come i file utilizzando il @ segno seguito dal nome del file.

Note

Il blueprint non verrà applicato finché la pull request non sarà approvata e unita. Per ulteriori informazioni, consultare [Revisione di una richiesta di pull](#) e [Unione di una richiesta di pull](#).

Gli autori del blueprint possono anche aggiungere un blueprint personalizzato ai progetti in spazi specifici che non dispongono del blueprint per creare nuovi progetti o aggiungerlo a progetti esistenti. Per ulteriori informazioni, consulta [Pubblicazione e aggiunta di un blueprint personalizzato in spazi e progetti specifici](#).

Se non desideri più ricevere aggiornamenti per un blueprint, puoi dissociare il blueprint dal tuo progetto. Per ulteriori informazioni, consulta [Dissociazione di un blueprint da un progetto per interrompere gli aggiornamenti](#).

Dissociazione di un blueprint da un progetto per interrompere gli aggiornamenti

Se non desideri nuovi aggiornamenti da un blueprint, puoi dissociare il blueprint dal progetto. Le risorse e i componenti software funzionali aggiunti al progetto dal blueprint rimarranno nel progetto.

Important

Per dissociare un blueprint dal CodeCatalyst progetto, è necessario accedere con un account che abbia il ruolo di amministratore dello spazio, Power user o amministratore del progetto nello spazio.

Per dissociare un blueprint dal progetto

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella CodeCatalyst console, accedi allo spazio, quindi scegli il progetto in cui desideri dissociare un blueprint.
3. Nel pannello di navigazione scegli Blueprints (Progetti).
4. Scegli il blueprint con le risorse da dissociare, scegli il menu a discesa Azioni, quindi scegli Dissocia blueprint.
5. Entra `confirm` per confermare la dissociazione.
6. Scegli Conferma.

Modifica delle versioni del blueprint in un progetto

Se hai creato un progetto con un blueprint o aggiunto un blueprint a un progetto esistente, riceverai una notifica delle nuove versioni del blueprint. Prima che la versione del blueprint venga aggiornata tramite una pull request approvata, puoi visualizzare le modifiche al codice e gli ambienti interessati. La gestione del ciclo di vita consente di modificare le versioni di uno o più blueprint applicati al progetto, in modo che ogni versione del blueprint possa essere modificata senza influire sulle altre aree del progetto. È inoltre possibile sovrascrivere gli aggiornamenti dei blueprint. Per ulteriori informazioni, consulta [Utilizzo della gestione del ciclo di vita come utente del blueprint](#).

 Important

Per modificare la versione di un blueprint nel CodeCatalyst progetto, è necessario accedere con un account che abbia il ruolo di amministratore dello spazio, Power user o amministratore del progetto nello spazio.

 Tip

Dopo aver aggiunto un blueprint al progetto, puoi configurare le notifiche e-mail e Slack per fornire aggiornamenti per le ultime modifiche al blueprint. Per ulteriori informazioni, consulta [Invio di notifiche da CodeCatalyst](#).

Per aggiornare un blueprint alla versione più recente

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri aggiornare la versione di un blueprint.
3. Nella dashboard dello spazio, scegli il progetto con il blueprint che desideri aggiornare.
4. Nel pannello di navigazione scegli Blueprints (Progetti).
5. Nella colonna Stato, scegli il link per modificare la versione del catalogo (ad esempio, Modifica versione del catalogo 0.3.109).
6. Scegli il menu a discesa Azioni, quindi scegli Aggiorna versione. La versione più recente viene selezionata automaticamente.
7. (Facoltativo) In Configura blueprint, configura i parametri del blueprint.
8. (Facoltativo) Nella scheda Modifiche al codice, esamina le differenze tra la versione corrente del blueprint e la versione aggiornata. La differenza visualizzata in una pull request è rappresentata dalle modifiche tra la versione corrente e la versione più recente, che è la versione desiderata al momento della creazione della pull request. Se non viene visualizzata alcuna modifica, le versioni potrebbero essere identiche oppure potresti aver scelto la stessa versione sia per la versione corrente che per quella desiderata.
9. Quando sei sicuro che la pull request contenga il codice e le modifiche che desideri rivedere, scegli Applica aggiornamento. Dopo aver creato la pull request, puoi aggiungere commenti. I commenti possono essere aggiunti alla pull request o alle singole righe dei file e alla pull request

complessiva. È possibile aggiungere collegamenti a risorse come i file utilizzando il @ segno, seguito dal nome del file.

Note

Il blueprint non verrà aggiornato finché la pull request non sarà approvata e unita.

Per ulteriori informazioni, consultare [Revisione di una richiesta di pull](#) e [Unione di una richiesta di pull](#).

Note

Se sono disponibili richieste pull esistenti per l'aggiornamento di una versione del blueprint, chiudi le richieste pull precedenti prima di crearne una nuova. Quando scegli **Aggiorna versione**, verrai indirizzato all'elenco delle richieste pull in sospeso per il blueprint. Puoi anche visualizzare le richieste pull in sospeso dalla scheda **Blueprints** nelle Impostazioni del progetto e nella pagina di riepilogo del progetto. Per ulteriori informazioni, consulta [Visualizzazione delle richieste pull](#).

Per modificare la versione di un blueprint

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri aggiornare la versione di un blueprint.
3. Nella dashboard dello spazio, scegli il progetto con il blueprint che desideri aggiornare.
4. Nel riquadro di navigazione, scegli **Blueprint**, quindi scegli il pulsante di opzione per il blueprint che desideri aggiornare.
5. Scegli il menu a discesa **Azioni**, quindi scegli **Configura blueprint**.
6. Dal menu a discesa **Versione di Target**, scegli la versione che desideri utilizzare. La versione più recente viene selezionata automaticamente.
7. (Facoltativo) In **Configura blueprint**, configura i parametri del blueprint.
8. (Facoltativo) Nella scheda **Modifiche al codice**, esamina le differenze tra la versione corrente del blueprint e la versione aggiornata. La differenza visualizzata in una pull request è rappresentata dalle modifiche tra la versione corrente e la versione più recente, che è la versione desiderata al momento della creazione della pull request. Se non viene visualizzata alcuna modifica, le

versioni potrebbero essere identiche oppure potresti aver scelto la stessa versione sia per la versione corrente che per quella desiderata.

9. Quando sei sicuro che la pull request contenga il codice e le modifiche che desideri rivedere, scegli **Applica aggiornamento**. Dopo aver creato la pull request, puoi aggiungere commenti. I commenti possono essere aggiunti alla pull request o alle singole righe dei file e alla pull request complessiva. È possibile aggiungere collegamenti a risorse come i file utilizzando il @ segno, seguito dal nome del file.

Note

Il blueprint non verrà aggiornato finché la pull request non sarà approvata e unita. Per ulteriori informazioni, consultare [Revisione di una richiesta di pull](#) e [Unione di una richiesta di pull](#).

Note

Se sono disponibili richieste pull esistenti per l'aggiornamento di una versione del blueprint, chiudi le richieste pull precedenti prima di crearne una nuova. Quando scegli **Aggiorna versione**, verrai indirizzato all'elenco delle richieste pull in sospeso per il blueprint. Puoi anche visualizzare le richieste pull in sospeso dalla scheda **Blueprints** nelle Impostazioni del progetto e nella pagina di riepilogo del progetto. Per ulteriori informazioni, consulta [Visualizzazione delle richieste pull](#).

Modifica della descrizione di un blueprint in un progetto

Puoi modificare la descrizione di un blueprint utilizzato per creare un progetto o applicato dopo la creazione di un progetto. Un blueprint può essere utilizzato più di una volta in un progetto. Per differenziare lo scopo dei blueprint nel progetto, è possibile utilizzare le relative descrizioni. Le descrizioni possono essere utilizzate anche per identificare i componenti che stai aggiungendo da un blueprint specifico.

⚠ Important

Per modificare la descrizione di un blueprint personalizzato nel proprio spazio, è necessario accedere con un account che abbia il ruolo di amministratore dello spazio, Power user o amministratore di progetto nello spazio.

Per modificare la descrizione di un blueprint nel progetto

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi al tuo spazio, quindi scegli il progetto con le impostazioni del blueprint che desideri aggiornare.
3. Nel pannello di navigazione scegli Blueprints (Progetti).
4. Scegli il blueprint con la descrizione che desideri aggiornare, scegli il menu a discesa Azioni, quindi scegli Impostazioni.
5. Nel campo di immissione del testo descrittivo del progetto, inserisci una descrizione per identificare il progetto.
6. Scegli Save (Salva).

Utilizzo della gestione del ciclo di vita come utente del blueprint

La gestione del ciclo di vita è la capacità di rigenerare una base di codice da opzioni o versioni aggiornate di un blueprint. Ciò consente all'autore di un blueprint di gestire centralmente il ciclo di vita di sviluppo del software di ogni progetto che contiene un particolare blueprint. Ad esempio, l'aggiunta di una correzione di sicurezza al blueprint di un'applicazione Web consentirà a ogni progetto che contiene o creato dal blueprint dell'applicazione Web di rilevare automaticamente tale correzione. Questo stesso framework di gestione consente inoltre all'utente del blueprint di modificare le opzioni del blueprint dopo averle selezionate.

Argomenti

- [Utilizzo della gestione del ciclo di vita su progetti esistenti](#)
- [Utilizzo della gestione del ciclo di vita su più blueprint in un progetto](#)
- [Gestione dei conflitti nelle richieste pull del ciclo di vita](#)
- [Disattivazione delle modifiche alla gestione del ciclo di vita](#)
- [Sovrascrivere la gestione del ciclo di vita di un blueprint in un progetto](#)

Utilizzo della gestione del ciclo di vita su progetti esistenti

È possibile utilizzare la gestione del ciclo di vita per progetti creati a partire da blueprint o su progetti esistenti non associati a nessun blueprint. Ad esempio, è possibile aggiungere un blueprint di pratiche di sicurezza standard in un'applicazione five-year-old Java che non è mai stata creata a partire da un blueprint. Il blueprint genera un flusso di lavoro di scansione di sicurezza e altro codice correlato. Quella parte della base di codice nell'applicazione Java verrà ora aggiornata automaticamente con le migliori pratiche del team ogni volta che vengono apportate modifiche al blueprint.

Utilizzo della gestione del ciclo di vita su più blueprint in un progetto

Poiché i blueprint rappresentano componenti architettonici, è spesso possibile utilizzare più blueprint insieme nello stesso progetto. Ad esempio, un progetto potrebbe essere costituito da un blueprint di API Web centrale creato da un ingegnere della piattaforma aziendale, insieme a un blueprint di controllo del rilascio creato dal team addetto alla sicurezza delle app. Ciascuno di questi progetti può essere aggiornato in modo indipendente e memorizzerà le risoluzioni di fusione applicate in passato.

Note

Essendo componenti architettonici arbitrari, non tutti i progetti hanno senso insieme o funzioneranno logicamente insieme, anche se cercheranno comunque di fondersi l'uno nell'altro.

Gestione dei conflitti nelle richieste pull del ciclo di vita

Occasionalmente, le richieste pull del ciclo di vita potrebbero generare conflitti di fusione. Questi problemi possono essere risolti manualmente. Le risoluzioni vengono memorizzate nei successivi aggiornamenti del blueprint.

Disattivazione delle modifiche alla gestione del ciclo di vita

Gli utenti possono rimuovere un blueprint da un progetto per dissociare tutti i riferimenti al blueprint e disattivare gli aggiornamenti del ciclo di vita. Per motivi di sicurezza, ciò non rimuove né influisce sul codice o sulle risorse del progetto, incluso ciò che è stato aggiunto dal blueprint. Per ulteriori informazioni, consulta [Dissociazione di un blueprint da un progetto per interrompere gli aggiornamenti](#).

Sovrascrivere la gestione del ciclo di vita di un blueprint in un progetto

Se desideri sovrascrivere gli aggiornamenti di un blueprint a file specifici del tuo progetto, puoi includere un file di proprietà nel tuo repository. GitLabLa specifica [Code Owners](#) è la linea guida consigliata. Il blueprint rispetta sempre il file dei proprietari del codice più di ogni altra cosa e può generarne uno di esempio come il seguente:

```
new BlueprintOwnershipFile(sourceRepo, {
  resynthesis: {
    strategies: [
      {
        identifier: 'dont-override-sample-code',
        description: 'This strategy is applied accross all sample code. The
blueprint will create sample code, but skip attempting to update it.',
        strategy: MergeStrategies.neverUpdate,
        globs: [
          '**/src/**',
          '**/css/**',
        ],
      },
    ],
  },
});
```

Questo genera un file `.ownership-file` con i seguenti contenuti:

```
[dont-override-sample-code] @amazon-codecatalyst/blueprints.import-from-git
# This strategy is applied accross all sample code. The blueprint will create sample
code, but skip attempting to update it.
# Internal merge strategy: neverUpdate
**/src/**
**/css/**
```

Creazione di un progetto completo con CodeCatalyst progetti

Quando crei un progetto utilizzando un blueprint, CodeCatalyst crea un progetto completo con un repository di sorgenti, codice sorgente di esempio, CI/CD flussi di lavoro, report di compilazione e test e strumenti integrati di tracciamento dei problemi. Un blueprint di progetto utilizza il codice per fornire l'infrastruttura cloud, le risorse e gli artefatti sorgente di esempio per diversi tipi di applicazioni e framework.

Per ulteriori informazioni, consulta [Creare un progetto](#). È necessario essere l'amministratore dello spazio per creare un progetto.

Argomenti

- [Blueprint disponibili](#)
- [Ricerca di informazioni sul modello del progetto](#)

Blueprint disponibili

Nome del blueprint	Descrizione del blueprint
API Web ASP.NET Core	Questo modello crea un'applicazione API Web ASP.NET 6 ASP.NET Core. Il blueprint utilizza lo strumento di AWS distribuzione per .NET e offre un'opzione per configurare Amazon Elastic Container Service o AWS Elastic Beanstalk come obiettivo di distribuzione. AWS App Runner
AWS Glue ETL	Questo modello crea un'implementazione di riferimento ETL (Extract Transform Load) di esempio utilizzando AWS CDK, Glue AWS , AWS Lambda e Amazon Athena per convertire i valori separati da virgole (,) in Apache Parquet. CSVs
DevOps pipeline di distribuzione	Questo modello crea una pipeline di distribuzione utilizzando l'architettura di riferimento AWS Deployment Pipeline che distribuisce un'applicazione di riferimento in più fasi. AWS
API Java con AWS Fargate	Questo modello crea un progetto di servizio Web containerizzato. Il progetto utilizza l'interfaccia della riga di AWS comando di Copilot per creare e distribuire un servizio Web Java Spring Boot containerizzato supportato da Amazon DynamoDB su Amazon ECS. Il

Nome del blueprint	Descrizione del blueprint
	progetto distribuisce un'app containerizzata in un cluster Amazon ECS su elaborazione serverless. AWS Fargate L'app memorizza i dati in una tabella DynamoDB. Una volta eseguito correttamente il flusso di lavoro, il servizio Web di esempio è disponibile pubblicamente tramite Application Load Balancer.
Moderna applicazione web a tre livelli	Questo modello genera codice in Python per il livello applicativo e il framework front-end Vue per creare e distribuire un'applicazione web moderna a 3 livelli ben architettata.
Applicazione server.NET	Questo modello crea AWS Lambda funzioni utilizzando gli strumenti Lambda dell'interfaccia CLI.NET. Il blueprint fornisce opzioni per le AWS Lambda funzioni, inclusa una scelta tra C# o F#.
API Node.js con AWS Fargate	Questo modello crea un progetto di servizio Web containerizzato. Il progetto utilizza la AWS CLI di Copilot per creare e distribuire un servizio Web Express/Node.js containerizzato su Amazon Elastic Container Service. Il progetto distribuisce un'app containerizzata in un cluster Amazon ECS su elaborazione serverless. AWS Fargate Una volta eseguito correttamente il flusso di lavoro, il servizio Web di esempio è disponibile pubblicamente tramite Application Load Balancer.

Nome del blueprint	Descrizione del blueprint
Modello applicativo senza server (SAM)	Questo modello crea un progetto che utilizza un modello applicativo serverless (SAM) per creare e distribuire un'API. Puoi scegliere SDK for Java TypeScript o SDK for Python come linguaggio di programmazione.
Microservizio senza server RESTful	Questo modello crea un'API REST che utilizza AWS Lambda e Amazon API Gateway con un riferimento al servizio To Do. Puoi scegliere SDK for Java TypeScript o SDK for Python come linguaggio di programmazione.
Applicazione a pagina singola	Questo modello crea un'applicazione a pagina singola (SPA) che utilizza i framework React, Vue e Angular. Per l'hosting, scegli tra AWS Amplify Hosting o Amazon CloudFront Amazon S3.
Sito Web statico	Questo progetto crea un sito Web statico utilizzando i generatori di siti statici Hugo o Jekyll . I generatori di siti statici utilizzano file di input di testo (come Markdown) per generare pagine Web statiche. Sono ideali per contenuti informativi che cambiano raramente, come pagine di prodotti, documentazione e blog. Il modello utilizza il AWS CDK per distribuire pagine Web statiche su uno AWS Amplify o su Amazon S3 +. CloudFront
Applicazione web To Do	Questo progetto crea un'applicazione web serverless To Do con componenti frontend e backend. Puoi scegliere SDK for Java TypeScript o SDK for Python come linguaggio di programmazione.

Nome del blueprint	Descrizione del blueprint
Abbonati a un blueprint esterno	Questo blueprint crea un flusso di lavoro per ogni pacchetto importato. Questi flussi di lavoro vengono eseguiti una volta al giorno per verificare la presenza di nuove versioni dei pacchetti in NPM. Se esiste una nuova versione, il flusso di lavoro tenta di aggiungerla al tuo CodeCatalyst spazio come progetto personalizzato. L'azione avrà esito negativo se un pacchetto non può essere trovato o non è un blueprint. Il pacchetto di destinazione deve essere su NPM e il pacchetto deve essere un blueprint. Lo spazio deve essere sottoscritto a un livello che supporti i blueprint personalizzati.
Chatbot GenAI di Bedrock	Questo modello crea un chatbot di intelligenza artificiale generativa con Amazon Bedrock e Claude di Anthropic. Con questo modello, puoi creare e implementare il tuo parco giochi LLM sicuro e protetto dall'accesso, che può essere personalizzato in base ai tuoi dati. Per ulteriori informazioni, consulta la documentazione del Chatbot di Bedrock GenAI .

Ricerca di informazioni sul modello del progetto

Diversi progetti di progetto sono disponibili in. CodeCatalyst Per ogni progetto, sono disponibili un riepilogo e un file README. Il riepilogo descrive le risorse installate dal blueprint, mentre il file README spiega il blueprint in dettaglio e fornisce istruzioni su come utilizzarlo.

Standardizzazione dei progetti con progetti personalizzati in CodeCatalyst

Puoi standardizzare lo sviluppo e le migliori pratiche per i progetti del tuo CodeCatalyst spazio con progetti personalizzati. I blueprint personalizzati possono essere utilizzati per definire vari aspetti di

un CodeCatalyst progetto, come le definizioni del flusso di lavoro e il codice dell'applicazione. Dopo aver utilizzato un blueprint personalizzato per creare un nuovo progetto o averlo applicato a progetti esistenti, tutte le modifiche al blueprint sono disponibili per tali progetti come aggiornamenti di pull request. In qualità di autore del blueprint, puoi visualizzare i dettagli sui progetti che utilizzano i tuoi blueprint in tutto lo spazio a tua disposizione, in modo da vedere come gli standard vengono applicati ai diversi progetti. La gestione del ciclo di vita di un blueprint ti consente di gestire centralmente il ciclo di vita di sviluppo del software di ogni progetto, assicurandoti che i progetti del tuo settore continuino a seguire le migliori pratiche con le ultime modifiche o correzioni. Per ulteriori informazioni, consulta [Lavorare con la gestione del ciclo di vita come autore di progetti](#).

I blueprint personalizzati offrono la possibilità di aggiornare le versioni dei blueprint rispetto al progetto precedente tramite la risintesi. La risintesi è il processo di riesecuzione della sintesi dei blueprint con versioni aggiornate o la possibilità di incorporare correzioni e modifiche nei progetti esistenti. Per ulteriori informazioni, consulta [Concetti di progetti personalizzati](#).

È possibile che la standardizzazione e le migliori pratiche siano già in atto. Invece di creare e sviluppare un blueprint personalizzato da zero, puoi scegliere di convertire un repository di sorgenti esistente con codice sorgente in un blueprint personalizzato. Per ulteriori informazioni, consulta [Conversione degli archivi di origine in blueprint personalizzati](#).

[Per visualizzare l'SDK dei blueprint e i blueprint di esempio, consulta il repository open source. GitHub](#)

Argomenti

- [Concetti di progetti personalizzati](#)
- [Guida introduttiva ai blueprint personalizzati](#)
- [Tutorial: creazione e aggiornamento di un'applicazione React](#)
- [Conversione degli archivi di origine in blueprint personalizzati](#)
- [Lavorare con la gestione del ciclo di vita come autore di progetti](#)
- [Sviluppo di un modello personalizzato per soddisfare i requisiti del progetto](#)
- [Pubblicazione di un blueprint personalizzato in uno spazio](#)
- [Impostazione delle autorizzazioni di pubblicazione per un blueprint personalizzato](#)
- [Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali](#)
- [Modifica delle versioni del catalogo per un blueprint personalizzato](#)
- [Visualizzazione di dettagli, versioni e progetti di un blueprint personalizzato](#)
- [Rimozione di un blueprint personalizzato da un catalogo di space blueprint](#)

- [Eliminazione di un blueprint o di una versione personalizzati pubblicati](#)
- [Gestione delle dipendenze, delle mancate corrispondenze e degli strumenti](#)
- [Contribuisci](#)

Concetti di progetti personalizzati

Di seguito sono riportati alcuni concetti e termini che è necessario conoscere quando si lavora con blueprint personalizzati in CodeCatalyst.

Argomenti

- [Progetto Blueprint](#)
- [Progetti spaziali](#)
- [Catalogo dei progetti spaziali](#)
- [Sintesi](#)
- [Risintesi](#)
- [Opzioni parziali](#)
- [Projen](#)

Progetto Blueprint

Un progetto blueprint ti dà la possibilità di sviluppare e pubblicare progetti nel tuo spazio. Durante il processo di creazione del progetto viene creato un archivio di origine e il nome del repository è quello scelto quando si inseriscono i dettagli delle risorse del progetto. Durante il processo di creazione del blueprint, se si sceglie di generare una versione del workflow, nel blueprint viene creato un workflow di pubblicazione con il blueprint Blueprint Builder. Il flusso di lavoro pubblica automaticamente la versione più recente.

Progetti spaziali

Puoi visualizzare e gestire tutti i blueprint dalla tabella Space blueprint quando accedi alla sezione Blueprint del tuo spazio. Una volta pubblicati nel tuo spazio, i blueprint vengono resi disponibili come blueprint di spazio da aggiungere e rimuovere dal catalogo dei blueprint dello spazio. Puoi anche gestire le autorizzazioni di pubblicazione ed eliminare i blueprint dalla sezione Blueprint del tuo spazio. Per ulteriori informazioni, consulta [Visualizzazione di dettagli, versioni e progetti di un blueprint personalizzato](#).

Catalogo dei progetti spaziali

È possibile visualizzare tutti i progetti personalizzati aggiunti dal catalogo dei blueprint di uno spazio. Qui un membro dello spazio può scegliere il blueprint personalizzato per creare un nuovo progetto. Questo catalogo è diverso dal CodeCatalyst catalogo, che ha già progetti disponibili per tutti i membri dello spazio. Per ulteriori informazioni, consulta [Creazione di un progetto completo con CodeCatalyst progetti](#).

Sintesi

La sintesi è il processo di generazione di un pacchetto di CodeCatalyst progetto che rappresenta il codice sorgente, la configurazione e le risorse di un progetto. Il pacchetto viene quindi utilizzato dalle operazioni API di CodeCatalyst distribuzione per l'implementazione in un progetto. Il processo può essere eseguito localmente durante lo sviluppo di un blueprint personalizzato per emulare la creazione del progetto senza dover creare un progetto in. CodeCatalyst I seguenti comandi possono essere utilizzati per eseguire la sintesi:

```
yarn blueprint:synth          # fast mode
yarn blueprint:synth --cache  # wizard emulation mode
```

Il progetto si avvia da solo chiamando la `blueprint.ts` classe principale con quell'opzione `unita.defaults.json`. Un nuovo pacchetto di progetto viene generato nella cartella `synth/synth.[options-name]/proposed-bundle/`. L'output include il pacchetto di progetto generato da un blueprint personalizzato, in base alle opzioni impostate, incluse le [opzioni parziali](#) che potresti aver configurato.

Risintesi

La risintesi è il processo di rigenerazione di un blueprint con diverse opzioni di blueprint o versioni di blueprint di progetti esistenti. In qualità di autore del blueprint, puoi definire strategie di unione personalizzate nel codice del blueprint personalizzato. È inoltre possibile definire i limiti di proprietà in un `.ownership-file` file per specificare in quali parti del codebase è consentito aggiornare un blueprint. Sebbene il blueprint personalizzato possa proporre aggiornamenti al `.ownership-file`, gli sviluppatori di progetti che utilizzano il blueprint personalizzato possono determinare i limiti di proprietà dei propri progetti. È possibile eseguire la risintesi localmente e testare e aggiornare prima di pubblicare il blueprint personalizzato. Utilizzate i seguenti comandi per eseguire la risintesi:

```
yarn blueprint:resynth        # fast mode
yarn blueprint:resynth --cache # wizard emulation mode
```

Il progetto si avvia da solo chiamando la `blueprint.ts` classe principale con quell'opzione `unita.defaults.json`. Un nuovo pacchetto di progetto viene generato nella cartella `synth/resynth.[options-name]`. L'output include il pacchetto di progetto generato da un blueprint personalizzato, in base alle opzioni impostate, incluse le [opzioni parziali](#) che potresti aver configurato.

I seguenti contenuti vengono creati dopo i processi di sintesi e risintesi:

- `proposed-bundle`: l'output della sintesi quando viene eseguita con nuove opzioni per la versione del blueprint di destinazione.
- `existing-bundle` - Una simulazione del tuo progetto esistente. Se non c'è nulla in questa cartella, viene generata con lo stesso output di `proposed-bundle`.
- `ancestor-bundle`: un simulato di ciò che il tuo blueprint genererebbe se eseguito con una versione precedente, opzioni precedenti o una combinazione. Se non c'è nulla in questa cartella, viene generata con lo stesso output di `proposed-bundle`.
- `resolved-bundle`: il pacchetto viene sempre rigenerato e per impostazione predefinita è un'unione a tre vie tra, e. `proposed-bundle` `existing-bundle` `ancestor-bundle`. Questo pacchetto fornisce un'emulazione di ciò che una risintesi produrrebbe localmente.

Per ulteriori informazioni sui pacchetti di output Blueprint, consulta. [Generazione di file con risintesi](#)

Opzioni parziali

È possibile aggiungere varianti di opzione in base `src/wizard-configuration/` alle quali non è necessario enumerare l'intera `Options` interfaccia e le opzioni vengono unite all'inizio del file. `defaults.json`. Ciò consente di personalizzare i casi di test in base a opzioni particolari.

Esempio:

`Options`interfaccia:

```
{
  language: "Python" | "Java" | "Typescript",
  repositoryName: string
  ...
}
```

`defaults.json`file:

```
{
```

```
language: "Python",
repositoryName: "Myrepo"
...
}
```

Test di configurazione aggiuntivi:

- ```
#wizard-config-typescript-test.json
{
 language: "Typescript",
}
```

- ```
#wizard-config-java-test.json
{
  language: "Java",
}
```

Projen

Projen è uno strumento open source che i progetti personalizzati utilizzano per mantenersi aggiornati e coerenti. I progetti sono disponibili come pacchetti Projen perché questo framework offre la possibilità di creare, raggruppare e pubblicare progetti e si può utilizzare l'interfaccia per gestire le configurazioni e le impostazioni di un progetto.

Puoi usare Projen per aggiornare i progetti su larga scala, anche dopo la loro creazione. Lo strumento Projen è la tecnologia alla base della sintesi dei progetti che genera un pacchetto di progetti. Projen possiede la configurazione di un progetto e non dovrebbe avere alcun impatto su di te come autore del blueprint. Puoi eseguire l'operazione `yarn projen` per rigenerare la configurazione del tuo progetto dopo aver aggiunto le dipendenze oppure puoi modificare le opzioni nel file `projenrc.ts`. Projen è anche lo strumento di generazione sottostante per progetti personalizzati per sintetizzare un progetto. [Per ulteriori informazioni, consulta la pagina del progetto. GitHub](#) Per saperne di più su come lavorare con Projen, consulta la documentazione di [Projen](#).

Guida introduttiva ai blueprint personalizzati

Durante il processo di creazione di un blueprint, puoi configurare il blueprint e generare un'anteprima delle risorse del progetto. Ogni blueprint personalizzato è gestito da un CodeCatalyst progetto, che per impostazione predefinita contiene un flusso di lavoro per la pubblicazione nel catalogo dei blueprint dello spazio.

Durante la configurazione dei dettagli del blueprint personalizzato, puoi anche scegliere di archiviare il codice sorgente del blueprint in un repository di terze parti, dove puoi comunque gestire il blueprint personalizzato e utilizzare le funzionalità di gestione del ciclo di vita per mantenere sincronizzati i progetti del tuo spazio quando il blueprint personalizzato viene modificato. Per ulteriori informazioni, consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#) e [Lavorare con la gestione del ciclo di vita come autore di progetti](#).

Se disponi già di un repository di origine con standardizzazione e best practice, puoi scegliere di convertire tale repository di origine in un blueprint personalizzato. Per ulteriori informazioni, consulta [Conversione degli archivi di origine in blueprint personalizzati](#).

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un blueprint personalizzato in CodeCatalyst](#)
- [Fase 2: Sviluppa un blueprint personalizzato con componenti](#)
- [Passaggio 3: Visualizza l'anteprima di un blueprint personalizzato](#)
- [\(Facoltativo\) Passaggio 4: Pubblicare una versione di anteprima del blueprint personalizzata](#)

Prerequisiti

Prima di creare un blueprint personalizzato, considera i seguenti requisiti:

- Il tuo CodeCatalyst spazio deve essere di livello Enterprise. Per ulteriori informazioni, consulta [Managing billing](#) nella Amazon CodeCatalyst Administrator Guide.
- È necessario disporre del ruolo di amministratore di Space o Power user per creare blueprint personalizzati. Per ulteriori informazioni, consulta [Concessione dell'accesso con ruoli utente](#).

Fase 1: Creare un blueprint personalizzato in CodeCatalyst

Quando crei un blueprint personalizzato dalle impostazioni del tuo spazio, viene creato automaticamente un repository. Il repository include tutte le risorse necessarie per sviluppare il blueprint prima di pubblicarlo nel catalogo dei blueprint dello spazio.

Per creare un blueprint personalizzato

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri creare un blueprint personalizzato.

3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Scegli Crea progetto.
5. In Assegna un nome al tuo blueprint, inserisci il nome che desideri assegnare al progetto e i nomi delle risorse associate. Il nome deve essere univoco all'interno del tuo spazio.
6. (Facoltativo) Per impostazione predefinita, il codice sorgente creato dal blueprint viene archiviato in un CodeCatalyst repository. In alternativa, è possibile scegliere di archiviare il codice sorgente del blueprint in un repository di terze parti. Per ulteriori informazioni, consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#).

Effettua una delle seguenti operazioni a seconda del provider di repository di terze parti che desideri utilizzare:

- GitHub repository: Connect a un GitHub account.

Scegli il menu a discesa Avanzato, scegli GitHub come provider del repository, quindi scegli l'GitHub account in cui desideri archiviare il codice sorgente creato dal blueprint.

Note

Se utilizzi una connessione a un GitHub account, devi creare una connessione personale per stabilire una mappatura delle identità tra la tua identità e la tua CodeCatalyst identità. GitHub Per ulteriori informazioni, consulta [Connessioni personali](#) e [Accesso alle GitHub risorse con connessioni personali](#).

- Archivi Bitbucket: Connect uno spazio di lavoro Bitbucket.

Scegli il menu a discesa Avanzato, scegli Bitbucket come provider di repository, quindi scegli l'area di lavoro Bitbucket in cui desideri archiviare il codice sorgente creato dal blueprint.

- GitLab repository: Connect a un GitLab utente.

Scegli il menu a discesa Avanzato, scegli GitLab come provider del repository, quindi scegli l'GitLab utente in cui desideri archiviare il codice sorgente creato dal blueprint.

7. In Dettagli del progetto, procedi come segue:
 - a. Nel campo di immissione del testo del nome visualizzato del Blueprint, inserisci un nome che verrà visualizzato nel catalogo dei blueprint del tuo spazio.
 - b. Nel campo di immissione del testo Descrizione, inserisci una descrizione per il tuo blueprint personalizzato.

- c. Nel campo di immissione del testo del nome dell'autore, inserisci il nome dell'autore per il tuo blueprint personalizzato.
- d. (Facoltativo) Scegli le impostazioni avanzate.
 - i. Scegli + Aggiungi per aggiungere i tag che vengono aggiunti al package .j son file.
 - ii. Scegli il menu a discesa Licenza, quindi scegli una licenza per il tuo blueprint personalizzato.
 - iii. Nel campo di immissione del testo del nome del pacchetto Blueprint, inserisci un nome per identificare il tuo pacchetto blueprint.
 - iv. Per impostazione predefinita, un flusso di lavoro di rilascio viene generato utilizzando un blueprint di pubblicazione all'interno del progetto chiamato Blueprint Builder. Il flusso di lavoro pubblica la versione più recente del blueprint nel tuo spazio quando inserisci le modifiche, poiché le autorizzazioni di pubblicazione sono abilitate dal flusso di lavoro di rilascio. Per disattivare la generazione del flusso di lavoro, deseleziona la casella di controllo Rilascia flusso di lavoro.
8. (Facoltativo) Un progetto blueprint viene fornito con un codice predefinito per supportare la pubblicazione del blueprint nel catalogo dei blueprint dello spazio. Per visualizzare i file di definizione con aggiornamenti basati sulle selezioni dei parametri di progetto effettuate, scegli Visualizza codice o Visualizza flusso di lavoro da Genera anteprima del blueprint.
9. Scegli Crea progetto.

Se non hai disattivato la generazione del flusso di lavoro per il tuo blueprint personalizzato, il flusso di lavoro inizia automaticamente a essere eseguito quando viene creato il blueprint. Una volta completata l'esecuzione del flusso di lavoro, il blueprint personalizzato è disponibile per essere aggiunto al catalogo dei blueprint del tuo spazio per impostazione predefinita. Puoi disattivare le autorizzazioni di pubblicazione se non desideri che la versione più recente del blueprint venga pubblicata automaticamente nel tuo spazio. Per ulteriori informazioni, consulta [Impostazione delle autorizzazioni di pubblicazione per un blueprint personalizzato](#) e [Esecuzione di un workflow](#).

Poiché il flusso di lavoro di pubblicazione chiamato blueprint-release viene creato utilizzando un blueprint, il blueprint può essere trovato come blueprint applicato al progetto. Per ulteriori informazioni, consulta [Aggiungere un modello in un progetto per integrare le risorse](#) e [Dissociazione di un blueprint da un progetto per interrompere gli aggiornamenti](#).

Fase 2: Sviluppa un blueprint personalizzato con componenti

Una procedura guidata per il blueprint viene generata quando si crea un blueprint personalizzato e può essere modificata con componenti durante lo sviluppo del blueprint personalizzato. È possibile aggiornare i `src/defaults.json` file `src/blueprints.js` and per modificare la procedura guidata.

Important

Se desideri utilizzare pacchetti blueprint provenienti da fonti esterne, considera i rischi che tali pacchetti possono comportare. Sei responsabile dei blueprint personalizzati che aggiungi al tuo spazio e del codice che generano.

Crea un ambiente di sviluppo nel tuo CodeCatalyst progetto con un ambiente di sviluppo integrato (IDE) supportato prima di configurare il codice del blueprint. Un ambiente di sviluppo è necessario per lavorare con gli strumenti e i pacchetti richiesti.

Per creare un ambiente di sviluppo

1. Nel riquadro di navigazione, effettuate una delle seguenti operazioni:
 - a. Scegli Panoramica, quindi vai alla sezione I miei ambienti di sviluppo.
 - b. Scegli Code, quindi scegli Dev Environments.
 - c. Scegli Code, scegli Archivi di origine e scegli il repository che hai creato durante la creazione del blueprint.
2. Scegli Create Dev Environment.
3. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti](#) di sviluppo.
4. Scegli Lavora in un ramo esistente e dal menu a discesa Existing branch scegli il feature branch che hai creato.
5. (Facoltativo) Nel campo Alias - campo di immissione di testo opzionale, inserisci un alias per identificare l'ambiente di sviluppo.
6. Scegli Create (Crea) . Durante la creazione dell'ambiente di sviluppo, la colonna di stato dell'ambiente di sviluppo visualizza Avvio e la colonna di stato indica In esecuzione quando è stato creato l'ambiente di sviluppo.

Per ulteriori informazioni, consulta [Scrivi e modifica il codice con Dev Environments in CodeCatalyst](#).

Per sviluppare il tuo blueprint personalizzato

1. In un terminale funzionante, usa il seguente yarn comando per installare le dipendenze:

```
yarn
```

Gli strumenti e i pacchetti necessari sono resi disponibili tramite CodeCatalyst Dev Environment, incluso Yarn. Se stai lavorando su un progetto personalizzato senza un ambiente di sviluppo, installa prima Yarn sul tuo sistema. Per ulteriori informazioni, consulta la documentazione di installazione di [Yarn](#).

2. Sviluppa il tuo blueprint personalizzato in modo che sia configurato secondo le tue preferenze. Puoi modificare la procedura guidata del tuo blueprint aggiungendo componenti. Per ulteriori informazioni, consulta [Sviluppo di un modello personalizzato per soddisfare i requisiti del progetto](#), [Modifica delle funzionalità del blueprint con una procedura guidata di front-end](#) e [Pubblicazione di un blueprint personalizzato in uno spazio](#).

Passaggio 3: Visualizza l'anteprima di un blueprint personalizzato

Dopo aver configurato e sviluppato il tuo blueprint personalizzato, puoi visualizzare in anteprima e pubblicare la versione di anteprima del tuo blueprint nel tuo spazio. Una versione di anteprima ti dà la possibilità di verificare che il blueprint sia quello che desideri prima che venga utilizzato per creare nuovi progetti o applicato a progetti esistenti.

Per visualizzare in anteprima un blueprint personalizzato

1. In un terminale funzionante, utilizzate il seguente yarn comando:

```
yarn blueprint:preview
```

2. Vai al `See this blueprint at`: link fornito per visualizzare in anteprima il tuo blueprint personalizzato.
3. Verifica che l'interfaccia utente, incluso il testo, appaia come previsto in base alla configurazione. Se desideri modificare il blueprint personalizzato, puoi modificare il `blueprint.ts` file, risintetizzare il blueprint e quindi pubblicare nuovamente una versione di anteprima. Per ulteriori informazioni, consulta [Risintesi](#).

(Facoltativo) Passaggio 4: Pubblicare una versione di anteprima del blueprint personalizzata

Puoi pubblicare una versione di anteprima del tuo blueprint personalizzato nel tuo spazio se desideri aggiungerlo al catalogo dei blueprint del tuo spazio. Ciò consente di visualizzare il blueprint come utente prima di aggiungere la versione non in anteprima al catalogo. La versione di anteprima consente di pubblicare senza utilizzare una versione effettiva. Ad esempio, se lavori su una 0.0.1 versione, puoi pubblicare e aggiungere una versione di anteprima, in modo da pubblicare e aggiungere nuovi aggiornamenti per una seconda versione 0.0.2.

Per pubblicare una versione di anteprima di un blueprint personalizzato

Vai al `Enable version [version number] at`: link fornito per abilitare il tuo blueprint personalizzato. Questo collegamento viene fornito quando si esegue il yarn comando in [Passaggio 3: Visualizza l'anteprima di un blueprint personalizzato](#).

Dopo aver creato, sviluppato, visualizzato in anteprima e pubblicato il blueprint personalizzato, puoi pubblicare e aggiungere la versione finale del blueprint al catalogo dei blueprint del tuo spazio. Per ulteriori informazioni, consulta [Pubblicazione di un blueprint personalizzato in uno spazio](#) e [Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali](#).

Tutorial: creazione e aggiornamento di un'applicazione React

In qualità di autore di progetti, puoi sviluppare e aggiungere progetti personalizzati al catalogo dei blueprint del tuo spazio. Questi blueprint possono quindi essere utilizzati dai membri dello spazio per creare nuovi progetti o aggiungerli a progetti esistenti. Puoi continuare ad apportare modifiche ai tuoi blueprint che vengono poi resi disponibili come aggiornamenti tramite pull request.

Questo tutorial fornisce una guida dettagliata dal punto di vista dell'autore del blueprint e dal punto di vista dell'utente del blueprint. Il tutorial mostra come creare un progetto di applicazione web React a pagina singola. Il blueprint viene quindi utilizzato per creare un nuovo progetto. Quando il blueprint viene aggiornato con le modifiche, il progetto creato dal blueprint incorpora tali modifiche tramite una pull request.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un blueprint personalizzato](#)
- [Passaggio 2: Visualizza il flusso di lavoro di rilascio](#)

- [Fase 3: Aggiungere il blueprint al catalogo](#)
- [Passaggio 4: Crea un progetto con blueprint](#)
- [Fase 5: Aggiornare il blueprint](#)
- [Passaggio 6: Aggiornare la versione del catalogo pubblicata del progetto alla nuova versione](#)
- [Passaggio 7: Aggiorna il progetto con la nuova versione del blueprint](#)
- [Passaggio 8: Visualizza le modifiche nel progetto](#)

Prerequisiti

Per creare e aggiornare un blueprint personalizzato, è necessario aver completato le attività seguenti:
[Configura e accedi a CodeCatalyst](#)

- Disponi di un ID AWS Builder a cui accedere. CodeCatalyst
- Appartenete a uno spazio e fatevi assegnare il ruolo di amministratore dello spazio o Power user in quello spazio. Per ulteriori informazioni, consulta [Creare uno spazio](#), [Concessione delle autorizzazioni di spazio agli utenti](#) e [Ruolo di amministratore dello spazio](#).

Fase 1: Creare un blueprint personalizzato

Quando si crea un blueprint personalizzato, viene creato un CodeCatalyst progetto che contiene il codice sorgente del blueprint e gli strumenti e le risorse di sviluppo. Il progetto è il luogo in cui svilupperai, testerai e pubblicherai il blueprint.

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri creare un blueprint.
3. Scegli Impostazioni per accedere alle impostazioni dello spazio.
4. Nella scheda Impostazioni dello spazio, scegli Blueprint e scegli Crea blueprint.
5. Aggiorna i campi nella procedura guidata di creazione del blueprint con i seguenti valori:
 - Nel nome del progetto, immettere. `react-app-blueprint`
 - In Blueprint Display Name, immettere. `react-app-blueprint`
6. Facoltativamente, scegli Visualizza codice per visualizzare in anteprima il codice sorgente del blueprint per il tuo blueprint. Allo stesso modo, scegli Visualizza flusso di lavoro per visualizzare in anteprima il flusso di lavoro che verrà creato nel progetto che crea e pubblica il blueprint.
7. Scegli Crea progetto.

8. Una volta creato il blueprint, verrai indirizzato al progetto del blueprint. Questo progetto contiene il codice sorgente del blueprint, insieme agli strumenti e alle risorse necessari per sviluppare, testare e pubblicare il blueprint. È stato generato un flusso di lavoro di rilascio che ha pubblicato automaticamente il blueprint nello spazio.
9. Ora che il blueprint e il progetto blueprint sono stati creati, il passaggio successivo consiste nel configurarli aggiornando il codice sorgente. È possibile utilizzare Dev Environments per aprire e modificare il repository dei sorgenti direttamente nel browser.

Nel pannello di navigazione, scegli Codice, quindi scegli Dev Environments.

10. Scegli Crea ambiente di sviluppo, quindi scegli AWS Cloud9 (nel browser).
11. Mantieni le impostazioni predefinite e scegli Crea.
12. Nel AWS Cloud9 terminale, accedi alla directory del tuo progetto blueprint eseguendo il seguente comando:

```
cd react-app-blueprint
```

13. Una `static-assets` cartella viene creata e riempita automaticamente quando viene creato un blueprint. In questo tutorial, eliminerai la cartella predefinita e ne genererai una nuova per un blueprint di app React.

Elimina la cartella `static-assets` eseguendo il seguente comando:

```
rm -r static-assets
```

AWS Cloud9 è costruito su una piattaforma basata su Linux. Se utilizzi un sistema operativo Windows, puoi invece utilizzare il seguente comando:

```
rmdir /s /q static-assets
```

14. Ora che la cartella predefinita è stata eliminata, crea una `static-assets` cartella per un blueprint di react-app eseguendo il comando seguente:

```
npx create-react-app static-assets
```

Se richiesto, inserisci `y` per procedere.

Una nuova applicazione React è stata creata nella `static-assets` cartella con i pacchetti necessari. Le modifiche devono essere trasferite nel repository dei CodeCatalyst sorgenti remoti.

15. Assicurati di avere le modifiche più recenti, quindi esegui il commit e invia le modifiche all'archivio dei CodeCatalyst sorgenti del blueprint eseguendo i seguenti comandi:

```
git pull
```

```
git add .
```

```
git commit -m "Add React app to static-assets"
```

```
git push
```

Quando una modifica viene inserita nell'archivio dei sorgenti del blueprint, il flusso di lavoro di rilascio viene avviato automaticamente. Questo flusso di lavoro incrementa la versione del blueprint, crea il blueprint e lo pubblica nel tuo spazio. Nel passaggio successivo, passerai al flusso di lavoro di rilascio eseguito per vedere come sta andando.

Passaggio 2: Visualizza il flusso di lavoro di rilascio

1. Nella CodeCatalyst console, nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il flusso di lavoro Blueprint-Release.
3. Puoi vedere che il flusso di lavoro include azioni per creare e pubblicare il blueprint.
4. In Ultima esecuzione, scegli il link di esecuzione del flusso di lavoro per visualizzare l'esecuzione in base alla modifica del codice che hai apportato.
5. Una volta completata l'esecuzione, viene pubblicata la nuova versione del blueprint. Le versioni pubblicate dei blueprint possono essere visualizzate nelle Impostazioni dello spazio, ma non possono essere utilizzate nei progetti finché non vengono aggiunte al catalogo dei blueprint dello spazio. Nel passaggio successivo, aggiungerai il blueprint al catalogo.

Fase 3: Aggiungere il blueprint al catalogo

L'aggiunta di un blueprint al catalogo dei blueprint dello spazio rende il blueprint disponibile per l'uso in tutti i progetti in uno spazio. I membri dello spazio possono utilizzare il blueprint per creare nuovi progetti o aggiungerli a progetti esistenti.

1. Nella CodeCatalyst console, torna allo spazio.
2. Scegli Impostazioni, quindi scegli Blueprints.
3. Scegli react-app-blueprint, quindi scegli Aggiungi al catalogo.
4. Seleziona Salva.

Passaggio 4: Crea un progetto con blueprint

Ora che il blueprint è stato aggiunto al catalogo, può essere utilizzato nei progetti. In questo passaggio, creerai un progetto con il blueprint appena creato. In una fase successiva, aggiornerai questo progetto aggiornando e pubblicando una nuova versione del blueprint.

1. Scegli la scheda Progetti, quindi scegli Crea progetto.
2. Scegli Space blueprint, quindi scegli react-app-blueprint.

Note

Una volta scelto il blueprint, puoi vedere il contenuto del file del blueprint. README .md

3. Scegli Next (Successivo).

4.

Note

Il contenuto di questa procedura guidata per la creazione del progetto può essere configurato nel blueprint.

Immettete il nome del progetto come utente del blueprint. Per questo tutorial, digita `react-app-project`. Per ulteriori informazioni, consulta [Sviluppo di un modello personalizzato per soddisfare i requisiti del progetto](#).

Successivamente, effettuerai un aggiornamento del blueprint e aggiungerai la nuova versione al catalogo, che utilizzerai per aggiornare questo progetto.

Fase 5: Aggiornare il blueprint

Dopo aver utilizzato un blueprint per creare un nuovo progetto o averlo applicato a progetti esistenti, puoi continuare ad apportare aggiornamenti come autore del blueprint. In questo passaggio,

apporterai modifiche al blueprint e pubblicherai automaticamente una nuova versione nello spazio. La nuova versione può quindi essere aggiunta come versione del catalogo.

1. Vai al react-app-blueprint progetto creato in [Tutorial: creazione e aggiornamento di un'applicazione React](#).
2. Apri l'ambiente di sviluppo creato in [Tutorial: creazione e aggiornamento di un'applicazione React](#).
 - a. Nel riquadro di navigazione, scegli Codice, quindi scegli Dev Environments.
 - b. Nella tabella, individua l'ambiente di sviluppo, quindi scegli Apri in AWS Cloud9 (nel browser).
3. Quando è stato eseguito il workflow di rilascio del blueprint, la versione del blueprint è stata incrementata aggiornando il file `package.json`. Effettua la modifica eseguendo il seguente comando nel terminale: AWS Cloud9

```
git pull
```

4. Passa alla `static-assets` cartella eseguendo il seguente comando:

```
cd /projects/react-app-blueprint/static-assets
```

5. Crea un `hello-world.txt` file nella `static-assets` cartella eseguendo il seguente comando:

```
touch hello-world.txt
```

AWS Cloud9 è costruito su una piattaforma basata su Linux. Se utilizzi un sistema operativo Windows, puoi invece utilizzare il seguente comando:

```
echo > hello-world.txt
```

6. Nella barra di navigazione a sinistra, fai doppio clic sul `hello-world.txt` file per aprirlo nell'editor e aggiungi i seguenti contenuti:

```
Hello, world!
```

Salvare il file.

7. Assicurati di avere le modifiche più recenti, quindi esegui il commit e invia le modifiche all'archivio dei CodeCatalyst sorgenti del blueprint eseguendo i seguenti comandi:

```
git pull
```

```
git add .
```

```
git commit -m "prettier setup"
```

```
git push
```

L'invio delle modifiche ha avviato il flusso di lavoro di rilascio, che pubblicherà automaticamente la nuova versione del blueprint nello spazio.

Passaggio 6: Aggiornare la versione del catalogo pubblicata del progetto alla nuova versione

Dopo aver utilizzato un blueprint per creare un nuovo progetto o averlo applicato a progetti esistenti, è comunque possibile aggiornare il blueprint come autore del blueprint. In questo passaggio, apporterai modifiche al blueprint e cambierai la versione del catalogo del blueprint.

1. Nella CodeCatalyst console, torna allo spazio.
2. Scegli Impostazioni, quindi scegli Blueprints.
3. Scegli react-app-blueprint, quindi scegli Gestisci la versione del catalogo.
4. Scegli la nuova versione, quindi scegli Salva.

Passaggio 7: Aggiorna il progetto con la nuova versione del blueprint

Una nuova versione è ora disponibile nel catalogo dei progetti dello spazio. Come utente del blueprint, puoi aggiornare la versione del progetto creato in [Passaggio 4: Crea un progetto con blueprint](#). In questo modo avrai la certezza di disporre delle ultime modifiche e correzioni necessarie per soddisfare le best practice.

1. Nella CodeCatalyst console, vai al react-app-projectprogetto creato in [Passaggio 4: Crea un progetto con blueprint](#).

2. Nel pannello di navigazione scegli Blueprints (Progetti).
3. Scegli Aggiorna blueprint nella casella delle informazioni.
4. Nel pannello a destra delle modifiche al codice, puoi vedere `hello-world.txt` gli `package.json` aggiornamenti.
5. Scegli Applica aggiornamento.

La scelta di Applica aggiornamento crea una pull request nel progetto con le modifiche rispetto alla versione aggiornata del blueprint. Per apportare gli aggiornamenti al progetto, è necessario unire la pull request. Per ulteriori informazioni, consulta [Revisione di una richiesta di pull](#) e [Unione di una richiesta di pull](#).

1. Nella tabella Blueprints, trova il blueprint. Nella colonna Stato, scegli Pending pull request, quindi scegli il link alla pull request aperta.
2. Controlla la pull request, quindi scegli Merge.
3. Scegliete Fast forward merge per mantenere i valori predefiniti, quindi scegliete Merge.

Passaggio 8: Visualizza le modifiche nel progetto

Le modifiche al blueprint sono ora disponibili nel progetto in seguito [Passaggio 7: Aggiorna il progetto con la nuova versione del blueprint](#). In qualità di utente del blueprint, è possibile visualizzare le modifiche nell'archivio di origine.

1. Nel riquadro di navigazione, scegli Archivi di origine, quindi scegli il nome del repository di origine creato al momento della creazione del progetto.
2. In File, puoi visualizzare il `hello-world.txt` file creato in. [Fase 5: Aggiornare il blueprint](#)
3. Scegli `hello-world.txt` per visualizzare il contenuto del file.

La gestione del ciclo di vita offre agli autori di blueprint la possibilità di gestire centralmente il ciclo di vita di sviluppo del software di ogni progetto che contiene un particolare blueprint. Come illustrato in questo tutorial, è possibile inviare aggiornamenti al blueprint che possono quindi essere incorporati da progetti che hanno utilizzato il blueprint per creare un nuovo progetto o lo hanno applicato a un progetto esistente. Per ulteriori informazioni, consulta [Lavorare con la gestione del ciclo di vita come autore di progetti](#).

Conversione degli archivi di origine in blueprint personalizzati

I blueprint personalizzati ti consentono di incorporare best practice o nuove opzioni in più progetti in uno spazio. CodeCatalyst Oltre a poter creare e sviluppare un nuovo blueprint personalizzato partendo da zero, è anche possibile convertire un repository di sorgenti esistente CodeCatalyst o di terze parti con codice e best practice consolidate in un progetto di blueprint. È possibile evitare di copiare gli artefatti pertinenti dal repository esistente in un progetto di blueprint. Dopo aver convertito un repository di origine in un blueprint personalizzato, è possibile aggiornare, pubblicare e aggiungere il blueprint come qualsiasi altro blueprint personalizzato.

Dopo aver convertito un repository di origine in un blueprint personalizzato, il repository di origine viene ristrutturato per diventare un progetto blueprint, il contenuto del repository viene spostato in una `static-assets` cartella all'interno del repository e le risorse pertinenti necessarie per un blueprint vengono aggiunte al repository. Quando il blueprint personalizzato viene utilizzato per creare progetti o aggiunto a un progetto esistente, anche le definizioni del flusso di lavoro memorizzate nell'archivio di origine convertito vengono aggiunte ai progetti dal blueprint.

Note

Risorse come ambienti e segreti non vengono incluse quando si converte un repository di origine in un blueprint personalizzato. È necessario copiare o aggiungere manualmente tali risorse dopo aver convertito un repository di origine in un blueprint personalizzato.

Important

Per convertire un repository di origine in un blueprint personalizzato, è necessario accedere con un account che abbia il ruolo di amministratore del progetto, amministratore dello spazio o Power user nello spazio.

Per convertire un repository di origine in un blueprint personalizzato dall'elenco dei repository di origine

1. [Aprire la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella CodeCatalyst console, accedi allo spazio, quindi scegli il progetto in cui desideri convertire un repository di origine in un blueprint personalizzato.

3. Nel pannello di navigazione, scegli Codice, scegli Archivi di origine, quindi scegli il pulsante di opzione del repository di origine che desideri convertire in un blueprint personalizzato.
4. Scegli Converti in blueprint per convertire il tuo repository di origine in un blueprint personalizzato.

Puoi anche convertire i CodeCatalyst repository in blueprint personalizzati dalla pagina del repository di origine nella console. CodeCatalyst

Per convertire un repository di origine in un blueprint personalizzato dalla pagina del repository di origine

1. [Aprire la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella CodeCatalyst console, accedi allo spazio, quindi scegli il progetto in cui desideri convertire un repository di origine in un blueprint personalizzato.
3. Nel pannello di navigazione, scegli Codice, scegli Archivi di origine, quindi scegli il nome del repository di CodeCatalyst origine che desideri convertire in un blueprint personalizzato.
4. Scegli il menu a discesa Altro, quindi scegli Converti in blueprint per convertire il repository di origine in un blueprint personalizzato.

Una volta convertito il repository di origine in un blueprint personalizzato, viene eseguito automaticamente un flusso di lavoro di rilascio. Una volta completata con successo l'esecuzione, il blueprint personalizzato viene pubblicato nell'elenco dei blueprint personalizzati del tuo spazio. Da lì, puoi aggiungere il blueprint personalizzato convertito al catalogo del tuo spazio per creare nuovi progetti o aggiungerlo a progetti esistenti. Per ulteriori informazioni, consulta [Pubblicazione di un blueprint personalizzato in uno spazio](#) e [Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali](#).

 Important

Per aggiungere un blueprint personalizzato al catalogo dei blueprint del tuo spazio, devi accedere con un account che abbia il ruolo di amministratore dello spazio o Power user nello spazio.

Lavorare con la gestione del ciclo di vita come autore di progetti

La gestione del ciclo di vita ti consente di mantenere sincronizzati un gran numero di progetti da un'unica fonte comune di best practice. Ciò consente di ridimensionare la propagazione delle correzioni e la manutenzione di un numero qualsiasi di progetti durante l'intero ciclo di vita di sviluppo del software. La gestione del ciclo di vita semplifica le campagne interne, le correzioni di sicurezza, gli audit, gli aggiornamenti di runtime, le modifiche alle migliori pratiche e altre pratiche di manutenzione perché tali standard sono definiti in un unico posto e aggiornati automaticamente a livello centrale quando vengono pubblicati nuovi standard.

Quando viene pubblicata una nuova versione del blueprint, a tutti i progetti che lo contengono viene richiesto di aggiornarsi alla versione più recente. In qualità di autore del blueprint, puoi anche vedere la versione di un particolare blueprint contenuta in ogni progetto per motivi di conformità. In caso di conflitti in un repository di origine esistente, la gestione del ciclo di vita crea richieste pull. Per tutte le altre risorse, come Dev Environment, tutti gli aggiornamenti della gestione del ciclo di vita creano esclusivamente nuove risorse. Gli utenti sono liberi di unire o meno tali richieste pull. Quando le richieste pull in sospeso vengono unite, la versione del blueprint, incluse le opzioni, utilizzata nel progetto viene quindi aggiornata. Per informazioni su come utilizzare la gestione del ciclo di vita come utente del blueprint, consulta e. [Utilizzo della gestione del ciclo di vita su progetti esistenti](#) [Utilizzo della gestione del ciclo di vita su più blueprint in un progetto](#)

Argomenti

- [Test della gestione del ciclo di vita per gli output di bundle e i conflitti di unione](#)
- [Utilizzo di strategie di unione per generare pacchetti e specificare file](#)
- [Accesso agli oggetti contestuali per i dettagli del progetto](#)

Test della gestione del ciclo di vita per gli output di bundle e i conflitti di unione

Puoi testare localmente la gestione del ciclo di vita del tuo blueprint e unire la risoluzione dei conflitti. Nella synth/ directory viene generata una serie di bundle che rappresentano le varie fasi di un aggiornamento del ciclo di vita. Per testare la gestione del ciclo di vita, puoi eseguire il seguente comando yarn sul tuo blueprint: `yarn blueprint: resynth` Per saperne di più sulla risintesi e sui pacchetti, vedi e. [Risintesi](#) [Generazione di file con risintesi](#)

Utilizzo di strategie di unione per generare pacchetti e specificare file

È possibile utilizzare strategie di unione per generare pacchetti con risintesi e specificare file per gli aggiornamenti della gestione del ciclo di vita dei blueprint personalizzati. Sfruttando le strategie

resynthesis e merge, è possibile gestire gli aggiornamenti e controllare quali file vengono aggiornati durante le distribuzioni. Puoi anche scrivere strategie personalizzate per controllare il modo in cui le modifiche vengono unite ai progetti esistenti. CodeCatalyst

Argomenti

- [Generazione di file con risintesi](#)
- [Utilizzo di strategie di fusione](#)
- [Specificazione dei file per gli aggiornamenti della gestione del ciclo di vita](#)
- [Scrittura di strategie di fusione](#)

Generazione di file con risintesi

Resynthesis può unire il codice sorgente prodotto da un blueprint con il codice sorgente precedentemente generato dallo stesso blueprint, permettendo la propagazione delle modifiche a un blueprint ai progetti esistenti. Le unioni vengono eseguite dalla funzione tra i pacchetti di output del blueprint. `resynth()` Resynthesis genera innanzitutto tre pacchetti che rappresentano aspetti diversi del blueprint e dello stato del progetto. Può essere eseguito manualmente localmente con il `yarn blueprint:resynth` comando, che creerà i bundle se non esistono già. L'utilizzo manuale dei pacchetti ti consentirà di simulare e testare localmente il comportamento della risintesi. Per impostazione predefinita, i blueprint eseguono solo la risintesi tra i repository sottostanti, `src/*` poiché solo quella parte del pacchetto è in genere sotto il controllo del codice sorgente. Per ulteriori informazioni, consulta [Risintesi](#).

- `existing-bundle`- Questo pacchetto è una rappresentazione dello stato del progetto esistente. Questo è costruito artificialmente dal calcolo di sintesi per fornire al modello un contesto su cosa c'è nel progetto in cui viene implementato (se non altro). Se qualcosa esiste già in questa posizione quando si esegue la risintesi in locale, verrà ripristinato e considerato come una simulazione. Altrimenti, verrà impostato sul contenuto di `ancestor-bundle`
- `ancestor-bundle`- Questo è il pacchetto che rappresenta l'output del progetto se è stato sintetizzato con alcune opzioni e/o versioni precedenti. Se è la prima volta che questo blueprint viene aggiunto a un progetto, allora l'antenato non esiste, quindi è impostato sugli stessi contenuti di `existing-bundle`. A livello locale, se questo pacchetto esiste già in questa posizione, verrà rispettato come simulato.
- `proposed-bundle`- Questo è il pacchetto che simula il modello se è stato sintetizzato con alcune nuove opzioni e/o versioni. Questo è lo stesso pacchetto che verrebbe prodotto dalla funzione. `synth()` A livello locale, questo pacchetto viene sempre sovrascritto.

Ogni pacchetto viene creato durante una fase di risintesi a cui è possibile accedere dalla classe `blueprint` sotto `this.context.resynthesisPhase`

- **resolved-bundle**- Questo è il pacchetto finale, che è una rappresentazione di ciò che viene impacchettato e distribuito in un progetto. CodeCatalyst È possibile visualizzare quali file e differenze vengono inviati ai meccanismi di distribuzione. Questo è l'output della `resynth()` funzione che risolve le fusioni tra gli altri tre pacchetti.

L'unione a tre vie viene applicata prendendo la differenza tra `ancestor-bundle` e `proposed-bundle` e aggiungendola a `existing-bundle` per generare il `resolved-bundle`. Tutte le strategie di unione risolvono i file in `resolved-bundle`. Resynthesis risolve la portata di questi pacchetti con le strategie di fusione del progetto `resynth()` e produce il pacchetto risolto a partire dal risultato.

Utilizzo di strategie di fusione

È possibile utilizzare una strategia di fusione fornita dalla libreria Blueprints. Queste strategie forniscono modi per risolvere gli output e i conflitti dei file menzionati nella sezione. [Generazione di file con risintesi](#)

- **alwaysUpdate**- Una strategia che si risolve sempre nel file proposto.
- **neverUpdate**- Una strategia che si risolve sempre nel file esistente.
- **onlyAdd**- Una strategia che si risolve nel file proposto quando un file esistente non esiste già. Altrimenti, si risolve nel file esistente.
- **threeWayMerge**- Una strategia che esegue un'unione a tre vie tra i file predecessori esistenti, proposti e comuni. Il file risolto può contenere indicatori di conflitto se i file non possono essere uniti in modo pulito. Il contenuto dei file forniti deve essere codificato in UTF-8 affinché la strategia produca un output significativo. La strategia tenta di rilevare se i file di input sono binari. Se la strategia rileva un conflitto di fusione in un file binario, restituisce sempre il file proposto.
- **preferProposed**- Una strategia che esegue un'unione a tre vie tra i file predecessori esistenti, proposti e comuni. Questa strategia risolve i conflitti selezionando il lato del file proposto di ogni conflitto.
- **preferExisting**- Una strategia che esegue un'unione a tre vie tra i file predecessori esistenti, proposti e comuni. Questa strategia risolve i conflitti selezionando il lato del file esistente di ogni conflitto.

[Per visualizzare il codice sorgente per le strategie di fusione, consulta il repository open source. GitHub](#)

Specificazione dei file per gli aggiornamenti della gestione del ciclo di vita

Durante la risintesi, i blueprint controllano il modo in cui le modifiche vengono unite in un archivio di origine esistente. Tuttavia, potresti non voler inviare aggiornamenti a ogni singolo file del tuo blueprint. Ad esempio, il codice di esempio come i fogli di stile CSS deve essere specifico del progetto. La strategia di fusione a tre vie è l'opzione predefinita se non si specifica un'altra strategia. I blueprint possono specificare quali file possiedono e quali no specificando le strategie di unione sulla struttura del repository stesso. I blueprint possono aggiornare le proprie strategie di fusione e le strategie più recenti possono essere utilizzate durante la risintesi.

```
const sourceRepo = new SourceRepository(this, {
  title: 'my-repo',
});
sourceRepo.setResynthStrategies([
  {
    identifier: 'dont-override-sample-code',
    description: 'This strategy is applied accross all sample code. The blueprint
will create sample code, but skip attempting to update it.',
    strategy: MergeStrategies.neverUpdate,
    globs: [
      '**/src/**',
      '**/css/**',
    ],
  },
]);
```

È possibile specificare più strategie di fusione e l'ultima strategia ha la precedenza. I file scoperti sono di default three-way-merge simili a Git. Il MergeStrategies costruito fornisce diverse strategie di fusione, ma puoi scriverne di tue. Le strategie fornite aderiscono al driver di strategia [git merge](#).

Scrittura di strategie di fusione

Oltre a utilizzare una delle strategie di build merge fornite, puoi anche scrivere le tue strategie. Le strategie devono aderire a un'interfaccia strategica standard. È necessario scrivere una funzione strategica che acquisisca le versioni di un file da `existing-bundle` `proposed-bundle` `ancestor-bundle`, e le unisca in un unico file risolto. Per esempio:

```

type StrategyFunction = (
  /**
   * file from the ancestor bundle (if it exists)
   */
  commonAncestorFile: ContextFile | undefined,
  /**
   * file from the existing bundle (if it exists)
   */
  existingFile: ContextFile | undefined,
  /**
   * file from the proposed bundle (if it exists)
   */
  proposedFile: ContextFile | undefined,
  options?: {})
  /**
   * Return: file you'd like in the resolved bundle
   * passing undefined will delete the file from the resolved bundle
   */
=> ContextFile | undefined;

```

Se i file non esistono (non sono definiti), il percorso del file non esiste in quel particolare pacchetto di posizioni.

Esempio:

```

strategies: [
  {
    identifier: 'dont-override-sample-code',
    description: 'This strategy is applied across all sample code. The
blueprint will create sample code, but skip attempting to update it.',
    strategy: (ancestor, existing, proposed) => {
      const resolvedfile = ...
      ...
      // do something
      ...
      return resolvedfile
    },
    globs: [
      '**/src/**',
      '**/css/**',
    ],
  },
],

```

```
],
```

Accesso agli oggetti contestuali per i dettagli del progetto

In qualità di autore del blueprint, puoi accedere al contesto del progetto del blueprint durante la sintesi per ottenere informazioni come lo spazio e i nomi dei progetti o i file esistenti nell'archivio dei sorgenti di un progetto. Puoi anche ottenere dettagli come la fase di risintesi generata dal blueprint. Ad esempio, puoi accedere al contesto per sapere se stai risintetizzando per generare un pacchetto predecessore o un pacchetto proposto. Il contesto di codice esistente può quindi essere utilizzato per trasformare il codice nel repository. Ad esempio, puoi scrivere la tua strategia di risintesi per impostare standard di codice specifici. La strategia può essere aggiunta al `blueprint.ts` file per piccoli progetti oppure è possibile creare un file separato per le strategie.

L'esempio seguente mostra come trovare file nel contesto di un progetto, impostare un generatore di flussi di lavoro e impostare una strategia di risintesi fornita da blueprint per un particolare file:

```
const contextFiles = this.context.project.src.findAll({
  fileGlobs: ['**/package.json'],
});

// const workflows = this.context.project.src.findAll({
//   fileGlobs: ['**/.codecatalyst/**/*.yaml'],
// });

const security = new WorkflowBuilder(this, {
  Name: 'security-workflow',
});
new Workflow(this, repo, security.getDefinition());
repo.setResynthStrategies([
  {
    identifier: 'force-security',
    globs: ['**/.codecatalyst/security-workflow.yaml'],
    strategy: MergeStrategies.alwaysUpdate,
  },
]);

for (const contextFile of contextFiles) {
  const packageObject = JSON.parse(contextFile.buffer.toString());
  new SourceFile(internalRepo, contextFile.path, JSON.stringify({
    ...packageObject,
  }, null, 2));
}
```

```
}  
}
```

Sviluppo di un modello personalizzato per soddisfare i requisiti del progetto

Prima di pubblicare un blueprint personalizzato, puoi svilupparlo per soddisfare requisiti specifici. Puoi sviluppare il tuo blueprint personalizzato e testarlo creando un progetto durante l'anteprima. È possibile sviluppare il blueprint personalizzato per includere componenti del progetto, come codice sorgente specifico, connessioni di account, flussi di lavoro, problemi o qualsiasi altro componente che può essere creato in CodeCatalyst.

Important

Se desideri utilizzare pacchetti blueprint provenienti da fonti esterne, considera i rischi che possono derivare da tali pacchetti. Sei responsabile dei blueprint personalizzati che aggiungi al tuo spazio e del codice che generano.

Important

Per sviluppare un blueprint personalizzato nel tuo CodeCatalyst spazio, devi accedere con un account che abbia il ruolo di amministratore dello spazio o Power user nello spazio.

Per sviluppare o aggiornare un blueprint personalizzato

1. Riprendi il tuo ambiente di sviluppo. Per ulteriori informazioni, consulta [Ripresa del lavoro in un ambiente di sviluppo](#).

Se non disponi di un ambiente di sviluppo, devi prima crearne uno. Per ulteriori informazioni, consulta [Creazione di un ambiente di sviluppo](#).

2. Apri un terminale funzionante nel tuo ambiente di sviluppo.
3. Se hai scelto un flusso di lavoro di rilascio durante la creazione del blueprint, la versione più recente del blueprint viene pubblicata automaticamente. Estrai le modifiche per assicurarvi che il package .json file abbia la versione incrementata. Utilizza il seguente comando :

```
git pull
```

4. Nel `src/blueprint.ts` file, modifica le opzioni del tuo blueprint personalizzato. L'`Options` interfaccia viene interpretata dalla CodeCatalyst procedura guidata in modo dinamico per generare un'interfaccia utente di selezione (UI). È possibile sviluppare un blueprint personalizzato aggiungendo componenti e tag supportati. Per ulteriori informazioni, consultare [Modifica delle funzionalità del blueprint con una procedura guidata di front-end](#), [Aggiungere componenti di ambiente a un blueprint](#), [Aggiungere componenti regionali a un blueprint](#), [Aggiungere componenti del repository e del codice sorgente a un blueprint](#), [Aggiungere componenti del flusso di lavoro a un blueprint](#) e [Aggiungere componenti Dev Environments a un blueprint](#).

Puoi anche visualizzare l'SDK dei blueprint e i blueprint di esempio per ricevere ulteriore assistenza durante lo sviluppo del tuo blueprint personalizzato. [Per ulteriori informazioni, consulta il repository open source. GitHub](#)

I blueprint personalizzati forniscono pacchetti di anteprima come risultato di una sintesi riuscita. Il pacchetto del progetto rappresenta il codice sorgente, la configurazione e le risorse di un progetto e viene utilizzato dalle operazioni API di CodeCatalyst distribuzione per la distribuzione in un progetto. Se desideri continuare a sviluppare il tuo blueprint personalizzato, esegui nuovamente il processo di sintesi del blueprint. Per ulteriori informazioni, consulta [Concetti di progetti personalizzati](#).

Modifica delle funzionalità del blueprint con una procedura guidata di front-end

Una procedura guidata di selezione del blueprint attiva CodeCatalyst viene generata automaticamente dall'`Options` interfaccia nel file. `blueprint.ts` [La procedura guidata front-end supporta le modifiche e le funzionalità di un blueprint utilizzando commenti e tag in stile JSDOC. Options](#) È possibile utilizzare commenti e tag in stile JSDOC per eseguire attività. Ad esempio, puoi selezionare il testo visualizzato sopra un'opzione, abilitare funzionalità come la convalida dell'input o rendere un'opzione comprimibile. La procedura guidata funziona interpretando un albero di sintassi astratto (AST) generato dal tipo dell' TypeScript interfaccia. `Options` La procedura guidata si configura automaticamente nel miglior modo possibile nel tipo descritto. Non tutti i tipi sono supportati. Altri tipi supportati includono il selettore di regione e il selettore di ambiente.

Di seguito è riportato un esempio di procedura guidata che utilizza commenti e tag JSDOC con blueprint: `Options`

```
export interface Options {  
  /**  
   * What do you want to call your new blueprint?  }  
}
```

```

    * @validationRegex /^[a-zA-Z0-9_]+$ /
    * @validationMessage Must contain only upper and lowercase letters, numbers and
underscores
    */
    blueprintName: string;

    /**
    * Add a description for your new blueprint.
    */
    description?: string;

    /**
    * Tags for your Blueprint:
    * @collapsed true
    */
    tags?: string[];
}

```

Per impostazione predefinita, il nome visualizzato di ciascuna opzione dell'`Options` interfaccia viene visualizzato in `camelCase`. Il testo normale nel commento in stile JSDOC viene visualizzato come testo sopra l'opzione nella procedura guidata.

Argomenti

- [Tag supportati](#)
- [TypeScript Tipi supportati](#)
- [Comunicare con l'utente durante la sintesi](#)

Tag supportati

I seguenti tag JSDOC sono supportati da un blueprint personalizzato `Options` nella procedura guidata di front-end.

`@inlinePolicy. /path/to/policy/file.json`

- Richiede: opzione per essere un tipo. `Role`
- Utilizzo: consente di comunicare le politiche in linea necessarie per un ruolo. Il `policy.json` percorso dovrebbe essere in codice sorgente. Usa questo tag quando hai bisogno di una politica personalizzata per un ruolo.
- Dipendenze - `blueprint-cli 0.1.12` e oltre

- Esempio - @inlinePolicy ./deployment-policy.json

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @inlinePolicy ./path/to/deployment-policy.json
     */
    cdkRole: Role[];
  };
};
```

@trustPolicy. /path/to/policy/file.json

- Richiede: opzione per essere un tipo. Role
- Utilizzo: consente di comunicare le politiche di fiducia necessarie per un ruolo. Il policy.json percorso dovrebbe essere in codice sorgente. Usa questo tag quando hai bisogno di una politica personalizzata per un ruolo.
- Dipendenze - blueprint-cli 0.1.12 e oltre
- Esempio - @trustPolicy ./trust-policy.json

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @trustPolicy ./path/to/trust-policy.json
     */
    cdkRole: Role[];
  };
};
```

Espressione Regex @validationRegex

- Richiede che l'opzione sia una stringa.
- Utilizzo: esegue la convalida dell'input sull'opzione utilizzando l'espressione regex e i display specificati. @validationMessage
- Esempio - @validationRegex /^[a-zA-Z0-9_]+\$
- Raccomandazione: da utilizzare con @validationMessage. Per impostazione predefinita, il messaggio di convalida è vuoto.

Stringa @validationMessage

- Richiede - @validationRegex o altri errori per verificare l'utilizzo.
- Utilizzo: visualizza un messaggio di convalida in @validation* caso di errore.
- Esempio -@validationMessage Must contain only upper and lowercase letters, numbers, and underscores.
- Raccomandazione: utilizzare con@validationMessage. Per impostazione predefinita, il messaggio di convalida è vuoto.

@collapsed boolean (opzionale)

- Richiede: N/A
- Utilizzo: valore booleano che consente a una sottoopzione di essere comprimibile. Se l'annotazione compressa è presente, il suo valore predefinito è true. L'impostazione del valore su @collapsed false crea una sezione comprimibile inizialmente aperta.
- Esempio - @collapsed true

stringa @displayName

- Richiede - N/A
- Utilizzo: modifica il nome visualizzato dell'opzione. Consente formati diversi da CamelCase per il nome visualizzato.
- Esempio - @displayName Blueprint Name

stringa @displayName

- Richiede - N/A
- Utilizzo: modifica il nome visualizzato dell'opzione. Consente formati diversi da [CamelCase](#) per il nome visualizzato.
- Esempio - @displayName Blueprint Name

Numero @defaultEntropy

- Richiede: l'opzione deve essere una stringa.
- Utilizzo: aggiunge all'opzione una stringa alfanumerica randomizzata di una lunghezza specificata.

- Esempio - `@defaultEntropy 5`

stringa `@placeholder` (opzionale)

- Richiede: N/A
- Utilizzo: modifica il segnaposto predefinito del campo di testo.
- Esempio - `@placeholder type project name here`

Numero `@textArea` (opzionale)

- Richiede: N/A
- Utilizzo: converte la stringa immessa in un componente dell'area di testo per sezioni di testo più grandi. L'aggiunta di un numero definisce il numero di righe. L'impostazione predefinita è cinque righe.
- Esempio - `@textArea 10`

`@hidden` boolean (opzionale)

- Richiede: N/A
- Utilizzo: nasconde il file all'utente a meno che il controllo di convalida non fallisca. Il valore predefinito è `true`.
- Esempio - `@hidden`

`@button` boolean (opzionale)

- Richiede: N/A
- Utilizzo: l'annotazione deve trovarsi su una proprietà booleana. Aggiunge un pulsante che verrà sintetizzato come vero quando viene selezionato. Non è un interruttore.
- Esempio - `buttonExample: boolean;`

```
/**
 * @button
 */
buttonExample: boolean;
```

@showName boolean (opzionale)

- Richiede: N/A
- Utilizzo: può essere utilizzato solo su un tipo di connessione a un account. Mostra l'immissione di un nome nascosto. L'impostazione predefinita è `default_environment`.
- Esempio - `@showName true`

```
/**
 * @showName true
 */
accountConnection: AccountConnection<{
    ...
}>;
```

@showEnvironmentType boolean (opzionale)

- Richiede: N/A
- Utilizzo: può essere utilizzato solo su un tipo di connessione a un account. Mostra il menu a discesa del tipo di ambiente nascosto. Tutte le connessioni sono predefinite su `production`. Le opzioni sono Non produzione o Produzione.
- Esempio - `@showEnvironmentType true`

```
/**
 * @showEnvironmentType true
 */
accountConnection: AccountConnection<{
    ...
}>;
```

@forceDefault boolean (opzionale)

- Richiede: N/A
- Utilizzo: utilizza il valore predefinito fornito dall'autore del blueprint anziché il valore utilizzato in precedenza dall'utente.
- Esempio - `forceDefaultExample: any;`

```
/**
```

```

    * @forceDefault
    */
    forceDefaultExample: any;

```

@requires BlueprintName

- Richiede: annota l'interfaccia. `Options`
- Utilizzo: avvisa l'utente di aggiungere elementi specificati `blueprintName` al progetto come requisito per il blueprint corrente.
- Esempio - `@requires '@amazon-codecatalyst/blueprints.blueprint-builder'`

```

/*
 * @requires '@amazon-codecatalyst/blueprints.blueprint-builder'
 */
export interface Options extends ParentOptions {
  ...

```

Regex @filter

- Richiede: annota l'interfaccia `Selector` o `MultiSelect`.
- Utilizzo: filtra il menu a discesa nella procedura guidata in base alle opzioni che corrispondono all'espressione regolare specificata.
- Esempio - `@filter /blueprintPackageName/`

```

/**
 * @filter /myPackageName/
 */
blueprintInstantiation?: Selector<BlueprintInstantiation>;
...

```

TypeScript Tipi supportati

I seguenti TypeScript tipi sono supportati da un blueprint personalizzato `Options` nella procedura guidata di front-end.

Numero

- Richiede: opzione per essere un tipo. `number`
- Utilizzo: genera un campo di immissione numerico.
- Esempio - `age: number`

```
{  
  age: number  
  ...  
}
```

Stringa

- Richiede: l'opzione deve essere un tipo `string`.
- Utilizzo: genera un campo di input di tipo stringa.
- Esempio - `name: string`

```
{  
  age: string  
  ...  
}
```

Elenco stringhe

- Richiede: l'opzione deve essere una matrice di tipo `string`.
- Utilizzo: genera un input per un elenco di stringhe.
- Esempio - `isProduction: boolean`

```
{  
  isProduction: boolean  
  ...  
}
```

Checkbox

- Richiede: opzione per essere un `boolean`.

- Utilizzo: genera una casella di controllo.
- Esempio - `isProduction`: `boolean`

```
{
  isProduction: boolean
  ...
}
```

Radio

- Richiede: l'opzione deve essere un'unione di tre o meno stringhe.
- Utilizzo: genera una radio selezionata.

Note

Quando ci sono quattro o più elementi, questo tipo viene visualizzato come un menu a discesa.

- Esempio - `color`: `'red' | 'blue' | 'green'`

```
{
  color: 'red' | 'blue' | 'green'
  ...
}
```

Elenco a discesa

- Richiede - Opzione per essere un'unione di quattro o più stringhe.
- Utilizzo: genera un menu a discesa.
- Esempio - `runtimes`: `'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'`

```
{
  runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'
  ...
}
```

Sezione espandibile

- Richiede: opzione per essere un oggetto.
- Utilizzo: genera una sezione espandibile. Le opzioni dell'oggetto verranno annidate all'interno della sezione espandibile della procedura guidata.
- Esempio -

```
{
  expandableSectionTitle: {
    nestedString: string;
    nestedNumber: number;
  }
}
```

Tupla

- Richiede: l'opzione deve essere di tipo `Tuple`.
- Utilizzo: genera un input a pagamento con valore chiave.
- Esempio - tuple: `Tuple[string, string]>`

```
{
  tuple: Tuple[string, string]>;
  ...
}
```

Elenco di tuple

- Richiede: opzione per essere un array di tipo `Tuple`.
- Utilizzo: genera un input per l'elenco delle tuple.
- Esempio - tupleList: `Tuple[string, string]>[]`

```
{
  tupleList: Tuple[string, string]>[];
  ...
}
```

Selector

- Richiede: l'opzione deve essere di tipo `Selector`.
- Utilizzo: genera un elenco a discesa dei repository o dei blueprint di origine applicati a un progetto.
- Esempio - `sourceRepo`: `Selector<SourceRepository>`

```
{
  sourceRepo: Selector<SourceRepository>;
  sourceRepoOrAdd: Selector<SourceRepository | string>;
  blueprintInstantiation: Selector<BlueprintInstantiation>;
  ...
}
```

Selezione multipla

- Richiede: opzione per essere di tipo `Selector`
- Utilizzo: genera un input multiselezione.
- Esempio - `multiselect`: `MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>`

```
{
  multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>;
  ...
}
```

Comunicare con l'utente durante la sintesi

In qualità di autore del blueprint, puoi rispondere agli utenti oltre ai semplici messaggi di convalida. Ad esempio, un membro dello spazio potrebbe visualizzare una combinazione di opzioni che produce un blueprint non chiaro. I blueprint personalizzati supportano la capacità di comunicare messaggi di errore agli utenti richiamando la sintesi. Il blueprint di base implementa una `throwSynthesisError(...)` funzione che prevede un messaggio di errore chiaro. È possibile richiamare il messaggio utilizzando quanto segue:

```
//blueprint.ts
this.throwSynthesisError({
  name: BlueprintSynthesisErrorTypes.BlueprintSynthesisError,
  message: 'hello from the blueprint! This is a custom error communicated to the user.'
```



```
})
```

Generazione di input e rendering degli elementi della procedura guidata front-end

Puoi generare input guidati con la procedura guidata dinamica e creare dinamicamente elementi della procedura guidata front-end per il tuo blueprint personalizzato. `KVInput`

Argomenti

- [Creazione di ambienti di sviluppo](#)
- [Creazione dinamica di elementi della procedura guidata](#)

Creazione di ambienti di sviluppo

Il `KVInput` tipo `Dynamic` può essere utilizzato per generare input guidati front-end utilizzando le impostazioni predefinite del blueprint personalizzato. [Per visualizzare la maggior parte degli schemi, consulta la definizione dinamica. up-to-date `KVInput`](#)

L'esempio seguente mostra come è possibile utilizzare `Options` per modellare un oggetto:

```
import { DynamicKVInput } from '@amazon-codecatalyst/blueprints.blueprint';

export interface Options extends ParentOptions {

    parameters: DynamicKVInput[];

}
```

L'esempio seguente mostra come impostare i parametri predefiniti con diverse proprietà:

```
{
  "parameters": [
    {
      "key": "AWS_REGION",
      "value": "us-west-2",
      "displayType": "region",
      "possibleValues": [
        "us-west-1",
        "us-west-2",
        "us-east-1",
        "us-east-2"
      ]
    }
  ]
}
```

```
    ],
    "displayName": "AWS Region",
    "description": "AWS Region to deploy the solution to."
  },
  {
    "key": "SchedulingActive",
    "value": "Yes",
    "displayType": "dropdown",
    "possibleValues": [
      "Yes",
      "No"
    ],
    "displayName": "Scheduling Active",
    "description": "Activate or deactivate scheduling."
  },
  {
    "key": "ScheduledServices",
    "value": "Both",
    "displayType": "dropdown",
    "possibleValues": [
      "EC2",
      "RDS",
      "Both"
    ],
    "displayName": "Scheduled Services",
    "description": "Services to schedule."
  },
  {
    "key": "ScheduleRdsClusters",
    "value": "No",
    "displayType": "dropdown",
    "possibleValues": [
      "Yes",
      "No"
    ],
    "displayName": "Schedule RDS Clusters",
    "description": "Enable scheduling of Aurora clusters for RDS service."
  },
  {
    "key": "CreateRdsSnapshot",
    "value": "No",
    "displayType": "dropdown",
    "possibleValues": [
      "Yes",
```

```

        "No"
    ],
    "displayName": "Create RDS Snapshot",
    "description": "Create snapshot before stopping RDS instances (does not
apply to Aurora Clusters).",
  },
  {
    "key": "MemorySize",
    "value": "128",
    "displayType": "dropdown",
    "possibleValues": [
      "128",
      "384",
      "512",
      "640",
      "768",
      "896",
      "1024",
      "1152",
      "1280",
      "1408",
      "1536"
    ],
    "displayName": "Memory Size",
    "description": "Size of the Lambda function running the scheduler, increase
size when processing large numbers of instances.",
  },
  {
    "key": "UseCloudWatchMetrics",
    "value": "No",
    "displayType": "dropdown",
    "possibleValues": [
      "Yes",
      "No"
    ],
    "displayName": "Use CloudWatch Metrics",
    "description": "Collect instance scheduling data using CloudWatch metrics."
  },
  {
    "key": "LogRetentionDays",
    "value": "30",
    "displayType": "dropdown",
    "possibleValues": [
      "1",

```

```

        "3",
        "5",
        "7",
        "14",
        "30",
        "60",
        "90",
        "120",
        "150",
        "180",
        "365",
        "400",
        "545",
        "731",
        "1827",
        "3653"
    ],
    "displayName": "Log Retention Days",
    "description": "Retention days for scheduler logs."
},
{
    "key": "Trace",
    "value": "No",
    "displayType": "dropdown",
    "possibleValues": [
        "Yes",
        "No"
    ],
    "displayName": "Trace",
    "description": "Enable debug-level logging in CloudWatch logs."
},
{
    "key": "EnableSSMMaintenanceWindows",
    "value": "No",
    "displayType": "dropdown",
    "possibleValues": [
        "Yes",
        "No"
    ],
    "displayName": "Enable SSM Maintenance Windows",
    "description": "Enable the solution to load SSM Maintenance Windows, so
that they can be used for EC2 instance Scheduling."
},
{

```

```

        "key": "DefaultTimezone",
        "value": "UTC",
        "displayType": "string",
        "displayName": "Default Timezone",
        "description": "Default timezone to use for scheduling."
    },
    {
        "key": "Regions",
        "value": "us-west-2",
        "displayType": "string",
        "displayName": "Regions",
        "description": "List of regions in which instances should be scheduled,
leave blank for current region only."
    },
    {
        "key": "UsingAWSOrganizations",
        "value": "No",
        "displayType": "dropdown",
        "possibleValues": [
            "Yes",
            "No"
        ],
        "displayName": "Using AWS Organizations",
        "description": "Use AWS Organizations to automate spoke account
registration."
    },
    {
        "key": "Principals",
        "displayType": "string",
        "optional": false,
        "displayName": "Principals",
        "description": "(Required) If using AWS Organizations, provide the
Organization ID. Eg. o-xxxxyyy. Else, provide a comma separated list of spoke account
ids to schedule. Eg.: 1111111111, 2222222222 or {param: ssm-param-name}"
    },
    {
        "key": "Namespace",
        "value": "Default",
        "displayType": "string",
        "displayName": "Namespace",
        "description": "Provide unique identifier to differentiate between multiple
solution deployments (No Spaces). Example: Dev"
    },
    {

```

```
        "key": "SchedulerFrequency",
        "value": 5,
        "displayType": "number",
        "displayName": "Scheduler Frequency",
        "description": "Scheduler running frequency in minutes."
      }
    ]
  }
}
```

Creazione dinamica di elementi della procedura guidata

Lo stesso schema utilizzato per la creazione degli input della procedura guidata può essere utilizzato per rirenderizzare dinamicamente una procedura guidata durante la sintesi. Questo può essere usato per rispondere a domande di follow-up da parte di un utente quando necessario.

```
//blueprint.ts

export interface Options extends ParentOptions {
  ...
  dynamicOptions: OptionsSchemaDefinition<'optionsIdentifier', KVSchema>;
}
```

La procedura guidata può quindi essere impostata durante il periodo di sintesi utilizzando un `Options` componente.

```
import {
  OptionsSchemaDefinition,
  OptionsSchema,
} from '@amazon-codecatalyst/blueprints.blueprint';

...

// dynamically renders a number in the place where 'optionsIdentifier' was set in the
original options type.
new OptionsSchema<KVSchema>(this, 'optionsIdentifier', [
  {
    "key": "SchedulerFrequency",
    "value": 5,
    "displayType": "number",
    "displayName": "Scheduler Frequency",
    "description": "Scheduler running frequency in minutes."
  }
])
```

```
]);
```

Aggiungere componenti di ambiente a un blueprint

La procedura guidata per i progetti personalizzati viene generata dinamicamente dall'Optionsinterfaccia esposta tramite la procedura guidata. I blueprint supportano la generazione di componenti dell'interfaccia utente (UI) a partire dai tipi esposti.

Per importare i componenti dell'ambiente Amazon CodeCatalyst Blueprint

Nel tuo `blueprint.ts` file, aggiungi quanto segue:

```
import {...} from '@amazon-codecatalyst/codecatalyst-environments'
```

Argomenti

- [Creazione di ambienti di sviluppo](#)
- [Elenco degli ambienti](#)
- [Esempi di interfacce fittizie](#)

Creazione di ambienti di sviluppo

L'esempio seguente mostra come distribuire l'applicazione nel cloud:

```
export interface Options extends ParentOptions {  
  ...  
  myNewEnvironment: EnvironmentDefinition{  
    thisIsMyFirstAccountConnection: AccountConnection{  
      thisIsARole: Role['lambda', 's3', 'dynamo'];  
    };  
  };  
}
```

L'interfaccia genera un componente dell'interfaccia utente che richiede un nuovo ambiente (`myNewEnvironment`) con una singola connessione a un account (`thisIsMyFirstAccountConnection`). Viene inoltre generato un ruolo sulla connessione dell'account (`thisIsARole`) con `['lambda', 's3', 'dynamo']` le funzionalità minime richieste per il ruolo. Non tutti gli utenti dispongono di connessioni all'account, quindi dovresti verificare se un utente non connette un account o non collega un account a un ruolo. I ruoli possono anche essere

annotati con `@inlinePolicies`. Per ulteriori informazioni, consulta [@inlinePolicy. /path/to/policy/file.json](#).

Il componente ambientale richiede un comando `name` e `environmentType`. Il codice seguente è la forma predefinita minima richiesta:

```
{
  ...
  "myNewEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION"
  },
}
```

Il componente dell'interfaccia utente richiede quindi l'immissione di vari campi. Man mano che compili i campi, il progetto assume una forma completamente espansa. Può esserti utile includere il mock completo nel `defaults.json` file per scopi di test e sviluppo.

Elenco degli ambienti

La specificazione di un tipo di matrice `EnvironmentDefinition` genererà un elenco di ambienti nell'interfaccia utente della procedura guidata.

```
export interface Options extends ParentOptions {
  ...
  /**
   * @showName readOnly
   */
  myEnvironments: EnvironmentDefinition<{
    thisIsMyFirstAccountConnection: AccountConnection<{
      thisIsARole: Role<['lambda', 's3', 'dynamo']>;
    }>;
  }>[];
}
```

L'esempio seguente mostra le impostazioni predefinite per un elenco di ambienti:

```
{
  ...
  "myEnvironments": [
```



```
{
  "name": "myProductionEnvironment",
  "environmentType": "PRODUCTION"
},
{
  "name": "myDevelopmentEnvironment",
  "environmentType": "DEVELOPMENT"
},
]
```

Esempi di interfacce fittizie

Interfaccia fittizia semplice

```
{
  ...
  "thisIsMyEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION",
    "thisIsMySecondAccountConnection": {
      "id": "12345678910",
      "name": "my-account-connection-name",
      "secondAdminRole": {
        "arn": "arn:aws:iam::12345678910:role/ConnectedQuokkaRole",
        "name": "ConnectedQuokkaRole",
        "capabilities": [
          "lambda",
          "s3",
          "dynamo"
        ]
      }
    }
  }
}
```

Interfaccia fittizia complessa

```
export interface Options extends ParentOptions {
  /**
   * The name of an environment
   * @displayName This is a Environment Name
   * @collapsed
```

```

*/
thisIsMyEnvironment: EnvironmentDefinition{
  /**
   * comments about the account that is being deployed into
   * @displayName This account connection has an overridden name
   * @collapsed
   */
  thisIsMyFirstAccountConnection: AccountConnection{
    /**
     * Blah blah some information about the role that I expect
     * e.g. here's a copy-pastable policy: [to a link]
     * @displayName This role has an overridden name
     */
    adminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
    /**
     * Blah blah some information about the second role that I expect
     * e.g. here's a copy-pastable policy: [to a link]
     */
    lambdaRole: Role['lambda', 's3'];
  };
  /**
   * comments about the account that is being deployed into
   */
  thisIsMySecondAccountConnection: AccountConnection{
    /**
     * Blah blah some information about the role that I expect
     * e.g. here's a copy-pastable policy: [to a link]
     */
    secondAdminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
    /**
     * Blah blah some information about the second role that I expect
     * e.g. here's a copy-pastable policy: [to a link]
     */
    secondLambdaRole: Role['lambda', 's3'];
  };
};
}

```

Interfaccia fittizia completa

```

{
  ...
  "thisIsMyEnvironment": {

```

```
"name": "my-production-environment",
"environmentType": "PRODUCTION",
"thisIsMySecondAccountConnection": {
  "id": "12345678910",
  "name": "my-connected-account",
  "secondAdminRole": {
    "name": "LambdaQuokkaRole",
    "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
    "capabilities": [
      "admin",
      "lambda",
      "s3",
      "cloudfront"
    ]
  },
  "secondLambdaRole": {
    "name": "LambdaQuokkaRole",
    "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
    "capabilities": [
      "lambda",
      "s3"
    ]
  }
},
"thisIsMyFirstAccountConnection": {
  "id": "12345678910",
  "name": "my-connected-account",
  "adminRole": {
    "name": "LambdaQuokkaRole",
    "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
    "capabilities": [
      "admin",
      "lambda",
      "s3",
      "cloudfront"
    ]
  },
  "lambdaRole": {
    "name": "LambdaQuokkaRole",
    "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
    "capabilities": [
      "lambda",
      "s3"
    ]
  }
}
```

```
    }  
  }  
},  
}
```

Aggiungere componenti segreti a un blueprint

I segreti possono essere utilizzati CodeCatalyst per archiviare dati sensibili a cui è possibile fare riferimento nei flussi di lavoro. Puoi aggiungere un segreto al tuo blueprint personalizzato e farvi riferimento nel tuo flusso di lavoro. Per ulteriori informazioni, consulta [Mascherare i dati utilizzando segreti](#).

Per importare CodeCatalyst i blueprint di Amazon, digita una regione

Nel tuo `blueprint.ts` file, aggiungi quanto segue:

```
import { Secret, SecretDefinition } from '@amazon-codecatalyst/blueprint-  
component.secrets'
```

Argomenti

- [Creare un segreto](#)
- [Fare riferimento a un segreto in un flusso di lavoro](#)

Creare un segreto

L'esempio seguente crea un componente dell'interfaccia utente che richiede all'utente di inserire un valore segreto e una descrizione opzionale:

```
export interface Options extends ParentOptions {  
  ...  
  mySecret: SecretDefinition;  
}  
  
export class Blueprint extends ParentBlueprint {  
  constructor(options_: Options) {  
    new Secret(this, options.secret);  
  }  
}
```

Il componente segreto richiede un. name Il codice seguente è la forma predefinita minima richiesta:

```
{
  ...
  "secret": {
    "name": "secretName"
  },
}
```

Fare riferimento a un segreto in un flusso di lavoro

Il seguente blueprint di esempio crea un segreto e un flusso di lavoro che fa riferimento al valore segreto. Per ulteriori informazioni, consulta [Fare riferimento a un segreto in un flusso di lavoro](#).

```
export interface Options extends ParentOptions {
  ...
  /**
   *
   * @validationRegex /^\\w+$/
   */
  username: string;

  password: SecretDefinition;
}

export class Blueprint extends ParentBlueprint {
  constructor(options_: Options) {
    const password = new Secret(this, options_.password);

    const workflowBuilder = new WorkflowBuilder(this, {
      Name: 'my_workflow',
    });

    workflowBuilder.addAction({
      actionName: 'download_files',
      input: {
        Sources: ['WorkflowSource'],
      },
      output: {
        Artifacts: [{ Name: 'download', Files: ['file1'] }],
      },
    });
  }
}
```

```
    steps: [  
      `curl -u ${options_.username}:${password.reference} https://example.com`,  
    ],  
  });  
  
  new Workflow(  
    this,  
    repo,  
    workflowBuilder.getDefinition(),  
  );  
}
```

Per ulteriori informazioni sull'utilizzo dei segreti in CodeCatalyst, vedere [Mascherare i dati utilizzando segreti](#).

Aggiungere componenti regionali a un blueprint

Il tipo di regione può essere aggiunto all'`Options`interfaccia del tuo blueprint personalizzato per generare un componente nella procedura guidata del blueprint, puoi inserire una o più regioni AWS. Il tipo gion può essere importato dal blueprint di base nel file. `blueprint.ts` Per ulteriori informazioni, consulta le [regioni AWS](#).

Per importare CodeCatalyst i blueprint di Amazon, digita una regione

Nel tuo `blueprint.ts` file, aggiungi quanto segue:

```
import { Region } from '@amazon-codecatalyst/blueprints.blueprint'
```

Il parametro del tipo di regione è una serie di codici regionali AWS tra cui scegliere, oppure puoi utilizzarlo `*` per includere tutte le regioni AWS supportate.

Argomenti

- [Annotazioni](#)
- [Esempi di componenti regionali](#)

Annotazioni

JSDoc i tag possono essere aggiunti a ciascun campo dell'`Options` interfaccia per personalizzare l'aspetto e il comportamento di un campo nella procedura guidata. Per il tipo di regione, sono supportati i seguenti tag:

- L'`@displayName` annotazione può essere utilizzata per modificare l'etichetta del campo nella procedura guidata.

Esempio: `@displayName AWS Region`

- L'`@placeholder` annotazione può essere utilizzata per modificare il segnaposto del componente di selezione/multiselezione.

Esempio: `@placeholder Choose AWS Region`

Esempi di componenti regionali

Scelta di una regione da un elenco specificato

```
export interface Options extends ParentOptions {  
    ...  
    /**  
     * @displayName Region  
     */  
    region: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>;  
}
```

Scelta di una o più regioni da un elenco specificato

```
export interface Options extends ParentOptions {  
    ...  
    /**  
     * @displayName Regions  
     */  
    multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];  
}
```

Scelta di una regione AWS

```
export interface Options extends ParentOptions {
```

```
...
/**
 * @displayName Region
 */
region: Region<['*']>;
}
```

Scelta di una o più regioni da un elenco specificato

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

Aggiungere componenti del repository e del codice sorgente a un blueprint

Un repository viene utilizzato da Amazon CodeCatalyst per archiviare il codice. Il repository prende un nome come input. La maggior parte dei componenti è archiviata in un repository, ad esempio file di codice sorgente, flussi di lavoro e altri componenti come gli ambienti di sviluppo gestiti (MDE). Il componente del repository di origine esporta anche i componenti utilizzati per la gestione di file e risorse statiche. I repository hanno vincoli relativi ai nomi. Per ulteriori informazioni, consulta [Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst](#).

```
const repository = new SourceRepository(this, {
  title: 'my-new-repository-title',
});
```

Per importare i componenti del repository e del codice sorgente di Amazon CodeCatalyst Blueprints

Nel tuo `blueprint.ts` file, aggiungi quanto segue:

```
import {...} from '@caws-blueprint-component/caws-source-repositories'
```

Argomenti

- [Aggiungere un file](#)
- [Aggiungere un file generico](#)

- [Copiare i file](#)
- [Individuazione di più file](#)
- [Creare un nuovo repository e aggiungere file](#)

Aggiungere un file

È possibile scrivere un file di testo in un repository con il `SourceFile` costruito. L'operazione è uno dei casi d'uso più comuni e richiede un repository, un percorso di file e contenuti di testo. Se il percorso del file non esiste all'interno di un repository, il componente crea tutte le cartelle richieste.

```
new SourceFile(repository, `path/to/my/file/in/repo/file.txt`, 'my file contents');
```

Note

Se si scrivono due file nella stessa posizione all'interno dello stesso repository, l'implementazione più recente sovrascrive quella precedente. È possibile utilizzare la funzionalità per sovrapporre il codice generato ed è particolarmente utile per estendere il codice che i blueprint personalizzati potrebbero aver generato.

Aggiungere un file generico

Puoi scrivere bit arbitrari nel tuo repository. È possibile leggere da un buffer e utilizzare il costruito. `File`

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

Copiare i file

Puoi iniziare con il codice generato copiando e incollando il codice iniziale e quindi generando altro codice su quella base. Inserite il codice all'interno della `static-assets` directory, quindi indirizzate il codice con il costruito. `StaticAsset` Il percorso in questo caso inizia sempre dalla radice della `static-assets` directory.

```
const starterCode = new StaticAsset('path/to/file/file.txt')
```

```
const starterCodeText = new StaticAsset('path/to/file/file.txt').toString()
const starterCodeRawContent = new StaticAsset('path/to/image/hello.png').content()

const starterCodePath = new StaticAsset('path/to/image/hello.png').path()
// starterCodePath is equal to 'path/to/image/hello.png'
```

Una sottoclasse di `StaticAsset` è `SubstitutionAsset`. La sottoclasse funziona esattamente allo stesso modo, ma puoi invece eseguire una sostituzione con i baffi sul file. Può essere utile per eseguire la generazione di stili. `copy-and-replace`

La sostituzione statica degli asset utilizza un motore di template Moustache per eseguire il rendering dei file statici inseriti nel repository di origine generato. Le regole di creazione dei modelli Moustache vengono applicate durante il rendering, il che significa che tutti i valori sono codificati in HTML per impostazione predefinita. Per eseguire il rendering del codice HTML senza escape, utilizzate la sintassi Triple Mustache. `{{{name}}}` [Per ulteriori informazioni, consulta le regole di creazione dei modelli per moustache.](#)

Note

L'esecuzione di un sostituto su file che non sono interpretabili dal testo può produrre errori.

```
const starterCodeText = new SubstitutionAsset('path/to/file/file.txt').substitute({
  'my_variable': 'subbed value1',
  'another_variable': 'subbed value2'
})
```

Individuazione di più file

Le risorse statiche supportano il targeting globale tramite una funzione statica `on StaticAsset` e le relative sottoclassi chiamate `findAll(...)`, che restituiscono un elenco di risorse statiche precaricate con i relativi percorsi, contenuti e altro. È possibile concatenare l'elenco con `File` costruzioni per copiare e incollare i contenuti nella directory. `static-assets`

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

Creare un nuovo repository e aggiungere file

È possibile utilizzare un componente del repository per creare un nuovo repository in un progetto generato. È quindi possibile aggiungere file o flussi di lavoro al repository creato.

```
import { SourceRepository } from '@amazon-codecatalyst/codecatalyst-source-repositories';
...
const repository = new SourceRepository(this, { title: 'myRepo' });
```

L'esempio seguente mostra come aggiungere file e flussi di lavoro a un repository esistente:

```
import { SourceFile } from '@amazon-codecatalyst/codecatalyst-source-repositories';
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows';
...
new SourceFile(repository, 'README.md', 'This is the content of my readme');
new Workflow(this, repository, {/**...workflowDefinition...**/});
```

La combinazione delle due parti di codice genera un unico repository denominato myRepo con un file sorgente README.md e un CodeCatalyst flusso di lavoro alla radice.

Aggiungere componenti del flusso di lavoro a un blueprint

Un flusso di lavoro viene utilizzato dai CodeCatalyst progetti Amazon per eseguire azioni basate su trigger. Puoi utilizzare i componenti del flusso di lavoro per creare e mettere insieme file YAML del flusso di lavoro. Per ulteriori informazioni, consulta [Definizione YAML del flusso di lavoro](#).

Per importare i componenti dei flussi di lavoro di Amazon CodeCatalyst Blueprint

Nel tuo blueprint.ts file, aggiungi quanto segue:

```
import { WorkflowBuilder, Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
```

Argomenti

- [Esempi di componenti del flusso di lavoro](#)
- [Connessione a un ambiente](#)

Esempi di componenti del flusso di lavoro

WorkflowBuilder componente

È possibile utilizzare una classe per creare una definizione di flusso di lavoro. La definizione può essere data a un componente del flusso di lavoro per il rendering in un repository.

```
import { WorkflowBuilder } from '@amazon-codecatalyst/codecatalyst-workflows'

const workflowBuilder = new WorkflowBuilder({} as Blueprint, {
  Name: 'my_workflow',
});

// trigger the workflow on pushes to branch 'main'
workflowBuilder.addBranchTrigger(['main']);

// add a build action
workflowBuilder.addAction({
  // give the action a name
  actionName: 'build_and_do_some_other_stuff',

  // the action pulls from source code
  input: {
    Sources: ['WorkflowSource'],
  },

  // the output attempts to autodiscover test reports, but not in the node modules
  output: {
    AutoDiscoverReports: {
      Enabled: true,
      ReportNamePrefix: AutoDiscovered,
      IncludePaths: ['**/*'],
      ExcludePaths: ['*/node_modules/**/*'],
    },
  },
});

// execute some arbitrary steps
steps: [
  'npm install',
  'npm run myscript',
  'echo hello-world',
],

// add an account connection to the workflow
environment: convertToWorkflowEnvironment(myEnv),
```

```
});
```

Componente Workflow Project

L'esempio seguente mostra come utilizzare un componente Projen per scrivere un flusso di lavoro YAML in un repository:

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'

...

const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// can also pass in any object and have it rendered as a yaml. This is unsafe and may
// not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

Connessione a un ambiente

Molti flussi di lavoro devono essere eseguiti in una connessione a un account AWS. I flussi di lavoro gestiscono questo problema consentendo alle azioni di connettersi ad ambienti con specifiche di account e nomi di ruolo.

```
import { convertToWorkflowEnvironment } from '@amazon-codecatalyst/codecatalyst-
workflows'

const myEnv = new Environment(...);

// can be passed into a workflow constructor
const workflowEnvironment = convertToWorkflowEnvironment(myEnv);

// add a build action
workflowBuilder.addAction({
  ...
  // add an account connection to the workflow
```

```
environment: convertToWorkflowEnvironment(myEnv),
});
```

Aggiungere componenti Dev Environments a un blueprint

Gli ambienti di sviluppo gestiti (MDE) vengono utilizzati per creare e configurare ambienti di lavoro MDE. CodeCatalyst Il componente genera un file. `devfile.yaml` Per ulteriori informazioni, vedere [Introduzione a Devfile](#) e [Modifica di un devfile del repository per un ambiente di sviluppo](#).

```
new Workspace(this, repository, SampleWorkspaces.default);
```

Per importare i componenti delle aree di lavoro di Amazon CodeCatalyst Blueprint

Nel tuo `blueprint.ts` file, aggiungi quanto segue:

```
import {...} from '@amazon-codecatalyst/codecatalyst-workspaces'
```

Aggiungere componenti relativi ai problemi a un blueprint

In CodeCatalyst, puoi monitorare funzionalità, attività, bug e qualsiasi altro lavoro coinvolto nel tuo progetto. Ogni lavoro è conservato in un registro distinto chiamato problema. Ogni problema può avere una descrizione, un assegnatario, uno stato e altre proprietà, che è possibile cercare, raggruppare e filtrare. Puoi visualizzare i problemi utilizzando le visualizzazioni predefinite oppure puoi creare visualizzazioni personalizzate con filtri, ordinamenti o raggruppamenti personalizzati. Per ulteriori informazioni sui concetti relativi ai problemi, consulta e. [Concetti relativi ai problemi](#) [Quote per le emissioni in CodeCatalyst](#)

Il componente `issue` genera una rappresentazione JSON di un problema. Il componente accetta come input un campo ID e una definizione del problema.

Per importare CodeCatalyst i blueprint di Amazon, emette componenti.

Nel tuo `blueprint.ts` file, aggiungi quanto segue:

```
import {...} from '@amazon-codecatalyst/blueprint-component.issues'
```

Argomenti

- [Problemi, componenti, esempi](#)

Problemi, componenti, esempi

Creazione di un problema

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myFirstIssue', {
  title: 'myFirstIssue',
  content: 'This is an example issue.',
});
```

Creazione di un problema ad alta priorità

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
...
const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// Creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// Can also pass in any object and have it rendered as a yaml. This is unsafe and may
  not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

Creazione di un problema a bassa priorità con etichette

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myThirdIssue', {
  title: 'myThirdIssue',
  content: 'This is an example of a low priority issue with a label.',
  priority: 'LOW',
  labels: ['exampleLabel'],
});
```

Utilizzo degli strumenti Blueprint e della CLI

La [CLI del blueprint](#) fornisce strumenti per gestire e lavorare con i tuoi blueprint personalizzati.

Argomenti

- [Lavorare con gli strumenti Blueprint](#)
- [Strumento di caricamento delle immagini](#)

Lavorare con gli strumenti Blueprint

Per lavorare con gli strumenti del blueprint

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Riprendi il tuo ambiente di sviluppo. Per ulteriori informazioni, consulta [Ripresa del lavoro in un ambiente di sviluppo](#).

Se non disponi di un ambiente di sviluppo, devi prima crearne uno. Per ulteriori informazioni, consulta [Creazione di un ambiente di sviluppo](#).

3. In un terminale funzionante, esegui il seguente comando per installare la CLI del blueprint:

```
npm install -g @amazon-codecatalyst/blueprint-util.cli
```

4. Nel `blueprint.ts` file, importa gli strumenti che desideri utilizzare nel seguente formato:

```
import { <tooling-function-name> } from '@amazon-codecatalyst/blueprint-util.cli/lib/<tooling-folder-name>/<tooling-file-name>;
```

Tip

Puoi andare su [CodeCatalyst blueprints GitHub repository](#) per trovare il nome dell'utensile che desideri utilizzare.

Se desideri utilizzare lo strumento di caricamento delle immagini, aggiungi quanto segue allo script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/image-upload-tool/upload-image-to-aws';
```

Examples (Esempi)

- Se desideri utilizzare la funzione di pubblicazione, aggiungi quanto segue allo script:


```
import { publish } from '@amazon-codecatalyst/blueprint-util.cli/lib/publish/publish';
```

- Se desideri utilizzare lo strumento di caricamento delle immagini, aggiungi quanto segue allo script:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/image-upload-tool/upload-image-to-aws';
```

5. Chiamate la funzione.

Esempi:

- Se desideri utilizzare la funzione di pubblicazione, aggiungi quanto segue allo script:

```
await publish(logger, config.publishEndpoint, {<your publishing options>});
```

- Se desideri utilizzare lo strumento di caricamento delle immagini, aggiungi quanto segue allo script:

```
const {imageUrl, imageName} = await uploadImagePublicly(logger, 'path/to/image');
```

Strumento di caricamento delle immagini

Lo strumento di caricamento delle immagini ti offre la possibilità di caricare la tua immagine in un bucket S3 nel tuo account AWS e poi distribuirla pubblicamente. CloudFront Lo strumento utilizza come input un percorso dell'immagine nello storage locale (e il nome del bucket opzionale) e restituisce l'URL dell'immagine che è disponibile pubblicamente. Per ulteriori informazioni, consulta [What is Amazon CloudFront?](#) e [cos'è Amazon S3?](#)

Per utilizzare lo strumento di caricamento delle immagini

1. Clona l' [GitHub archivio di blueprints open source](#) che fornisce l'accesso al blueprints SDK e al blueprint di esempio. In un terminale funzionante, esegui il seguente comando:

```
git clone https://github.com/aws/codecatalyst-blueprints.git
```

2. Esegui il comando seguente per accedere al GitHub repository dei blueprints:

```
cd codecatalyst-blueprints
```

3. Esegui il comando seguente per installare le dipendenze:

```
yarn && yarn build
```

4. Esegui il comando seguente per assicurarti che sia installata la versione CLI più recente del blueprint:

```
yarn upgrade @amazon-codecatalyst/blueprint-util.cli
```

5. Accedi all'account AWS con il bucket S3 in cui desideri caricare l'immagine. Per ulteriori informazioni, consulta [Configurare l'AWS CLI](#) e [accedere tramite l'interfaccia a riga di comando AWS](#).
6. Esegui il seguente comando dalla radice del tuo CodeCatalyst repository per accedere alla directory con la CLI blueprint:

```
cd packages/utils/blueprint-cli
```

7. Esegui il comando seguente per caricare l'immagine in un bucket S3:

```
yarn blueprint upload-image-public <./path/to/your/image>  
    <optional:optional-bucket-name>
```

Viene generato un URL per la tua immagine. L'URL non sarà disponibile immediatamente poiché la CloudFront distribuzione richiede del tempo per essere distribuita. Controlla lo stato della distribuzione per ottenere lo stato di distribuzione più recente. Per ulteriori informazioni, consulta [Lavorare con le distribuzioni](#).

Valutazione delle modifiche all'interfaccia con il test delle istantanee

Sono supportati i test di snapshot generati su più configurazioni del tuo blueprint.

I blueprint supportano il [test delle istantanee](#) sulle configurazioni fornite dall'utente in qualità di autore del blueprint. Le configurazioni sono sostituzioni parziali che vengono unite al file defaults.json alla radice di un blueprint. Quando il test delle istantanee è abilitato e configurato, il processo di compilazione e test sintetizza le configurazioni fornite e verifica che gli output sintetizzati non siano

cambiati rispetto allo snapshot di riferimento. [Per visualizzare il codice di test delle istantanee, consulta il repository dei blueprints. CodeCatalyst GitHub](#)

Per abilitare il test delle istantanee

1. Nel `.projenrc.ts` file, aggiorna l'oggetto di input `ProjenBlueprint` con i file di cui desideri creare un'istannea. Per esempio:

```
{
  ....
  blueprintSnapshotConfiguration: {
    snapshotGlobs: ['**', '!environments/**', '!aws-account-to-environment/**'],
  },
}
```

2. Risintetizzate il blueprint per creare TypeScript file nel vostro progetto blueprint. Non modificate i file sorgente poiché sono gestiti e rigenerati da Projen. Utilizza il seguente comando :

```
yarn projen
```

3. Vai alla `src/snapshot-configurations` directory per visualizzare il `default-config.json` file con un oggetto vuoto. Aggiorna o sostituisci il file con una o più configurazioni di test personalizzate. Ogni configurazione di test viene quindi unita al `defaults.json` file del progetto, sintetizzata e confrontata con le istantanee durante il test. Usa il seguente comando per testare:

```
yarn test
```

La prima volta che si utilizza un comando `test`, viene visualizzato il seguente messaggio: `Snapshot Summary > NN snapshots written from 1 test suite`. Le successive esecuzioni di test verificano che l'output sintetizzato non sia cambiato dalle istantanee e visualizzano il seguente messaggio: `Snapshots: NN passed, NN total`

Se modifichi intenzionalmente il tuo blueprint per produrre un output diverso, esegui il seguente comando per aggiornare le istantanee di riferimento:

```
yarn test:update
```

Le istantanee prevedono che gli output sintetizzati siano costanti tra ogni esecuzione. Se il blueprint genera file che variano, è necessario escludere tali file dal test delle istantanee. Aggiorna l'`blueprintSnapshotConfiguration` oggetto dell'oggetto di `ProjenBlueprint` input per aggiungere la `snapshotGlobs` proprietà. La `snapshotGlobs` proprietà è una matrice di [globs](#) che determina quali file sono inclusi o esclusi dall'istannea.

Note

Esiste un elenco predefinito di globs. Se si specifica un elenco personalizzato, potrebbe essere necessario ripristinare in modo esplicito le voci predefinite.

Pubblicazione di un blueprint personalizzato in uno spazio

Prima di poter aggiungere un blueprint personalizzato al catalogo dei blueprint del tuo spazio, devi pubblicarlo nello spazio. È inoltre possibile visualizzare il blueprint nella CodeCatalyst console prima della pubblicazione. È possibile pubblicare una versione di anteprima o una versione normale del blueprint.

Important

Se desideri utilizzare pacchetti blueprint provenienti da fonti esterne, considera i rischi che possono derivare da tali pacchetti. Sei responsabile dei blueprint personalizzati che aggiungi al tuo spazio e del codice che generano.

Important

Per pubblicare un blueprint personalizzato nel tuo CodeCatalyst spazio, devi accedere con un account che abbia il ruolo di amministratore dello spazio o Power user nello spazio.

Argomenti

- [Visualizzazione e pubblicazione di una versione di anteprima di un blueprint personalizzato](#)
- [Visualizzazione e pubblicazione di una versione normale di un blueprint personalizzato](#)
- [Pubblicazione e aggiunta di un blueprint personalizzato in spazi e progetti specifici](#)

Visualizzazione e pubblicazione di una versione di anteprima di un blueprint personalizzato

Puoi pubblicare una versione di anteprima del tuo blueprint personalizzato nel tuo spazio se desideri aggiungerlo al catalogo dei blueprint del tuo spazio. Ciò consente di visualizzare il blueprint come utente prima di aggiungere la versione non in anteprima al catalogo. La versione di anteprima consente di pubblicare senza utilizzare una versione effettiva. Ad esempio, se lavori su una 0.0.1 versione, puoi pubblicare e aggiungere una versione di anteprima, in modo da pubblicare e aggiungere nuovi aggiornamenti per una seconda versione 0.0.2.

Dopo aver apportato le modifiche, ricostruisci il pacchetto del tuo blueprint personalizzato eseguendo il `package.json` file e visualizza in anteprima le modifiche.

Per visualizzare e pubblicare una versione di anteprima di un blueprint personalizzato

1. Riprendi il tuo ambiente di sviluppo. Per ulteriori informazioni, consulta [Ripresa del lavoro in un ambiente di sviluppo](#)
2. Apri un terminale funzionante nel tuo ambiente di sviluppo.
3. (Facoltativo) In un terminale funzionante, installa le dipendenze necessarie per il tuo progetto se non le hai già installate. Utilizza il seguente comando :

```
yarn
```

4. (Facoltativo) Se hai apportato modifiche al `.projenrc.ts` file, rigenera la configurazione del progetto prima di creare e visualizzare l'anteprima del blueprint. Utilizza il seguente comando :

```
yarn projen
```

5. Ricostruisci e visualizza in anteprima il tuo blueprint personalizzato utilizzando il seguente comando. Utilizza il seguente comando :

```
yarn blueprint:preview
```

Vai al `See this blueprint at:` link fornito per visualizzare in anteprima il tuo blueprint personalizzato. Verifica che l'interfaccia utente, incluso il testo, appaia come previsto in base alla configurazione. Se desideri modificare il blueprint personalizzato, puoi modificare il `blueprint.ts` file, risintetizzare il blueprint e quindi pubblicare nuovamente una versione di anteprima. Per ulteriori informazioni, consulta [Risintesi](#).

6. (Facoltativo) Puoi pubblicare una versione di anteprima del tuo blueprint personalizzato, che può quindi essere aggiunta al catalogo dei blueprint del tuo spazio. Vai al `Enable version [preview version number] at`: link per pubblicare una versione di anteprima nel tuo spazio.

Puoi emulare la creazione di un progetto senza dover creare un progetto in CodeCatalyst. Per sintetizzare il progetto, utilizzate il seguente comando:

```
yarn blueprint:synth
```

Nella cartella viene generato un blueprint. `synth/synth.[options-name]/proposed-bundle/`
Per ulteriori informazioni, consulta [Sintesi](#).

Se stai aggiornando il tuo blueprint personalizzato, utilizza invece il seguente comando per risintetizzare il progetto:

```
yarn blueprint:resynth
```

Nella cartella viene generato un blueprint. `synth/synth.[options-name]/proposed-bundle/`
Per ulteriori informazioni, consulta [Risintesi](#).

Dopo aver pubblicato la versione di anteprima, è possibile aggiungere il blueprint in modo che i membri dello spazio possano utilizzarlo per creare nuovi progetti o aggiungere progetti esistenti. Per ulteriori informazioni, consulta [Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali](#).

Visualizzazione e pubblicazione di una versione normale di un blueprint personalizzato

Dopo aver completato lo sviluppo e la visualizzazione in anteprima del blueprint personalizzato, puoi visualizzare e pubblicare la nuova versione che desideri aggiungere al catalogo dei blueprint del tuo spazio. Il flusso di lavoro di rilascio generato durante la creazione di un progetto pubblica automaticamente le modifiche che vengono inviate. Se hai disattivato la generazione del flusso di lavoro durante la creazione del blueprint, il blueprint non viene automaticamente reso disponibile per essere aggiunto al catalogo dei blueprint del tuo spazio. È comunque possibile pubblicare il blueprint personalizzato nel proprio spazio dopo aver eseguito un comando. `yarn`

Per visualizzare e pubblicare un blueprint personalizzato

1. Riprendi il tuo ambiente di sviluppo. Per ulteriori informazioni, consulta [Ripresa del lavoro in un ambiente di sviluppo](#)
2. Apri un terminale funzionante nel tuo ambiente di sviluppo.
3.
 - Se hai disattivato la generazione del workflow di rilascio durante la creazione del blueprint, usa il seguente comando:

```
yarn blueprint:release
```

Puoi comunque accedere al `See this blueprint at:` link fornito per visualizzare il tuo blueprint personalizzato.

Pubblica la versione aggiornata del tuo blueprint personalizzato, che può quindi essere aggiunta al catalogo dei blueprint del tuo spazio. Vai al `Enable version [release version number] at:` link per pubblicare la versione più recente nel tuo spazio.

- Se hai scelto un workflow di rilascio durante la creazione del blueprint, la versione più recente del blueprint viene pubblicata automaticamente quando vengono inviate le modifiche. Utilizza il seguente comando:

```
git add .
```

```
git commit -m "commit message"
```

```
git push
```

Dopo aver pubblicato la versione normale, puoi aggiungere il blueprint in modo che i membri dello spazio possano utilizzarlo per creare nuovi progetti o aggiungere progetti esistenti. Per ulteriori informazioni, consulta [Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali](#).

Pubblicazione e aggiunta di un blueprint personalizzato in spazi e progetti specifici

Per impostazione predefinita, i `blueprint:release` comandi `blueprint:preview` and vengono pubblicati nello CodeCatalyst spazio in cui è stato creato il blueprint. Se disponi di più spazi Enterprise, puoi visualizzare in anteprima e pubblicare lo stesso blueprint anche in quegli spazi. Puoi anche aggiungere un blueprint a un progetto esistente di un altro spazio.

Per pubblicare o aggiungere un blueprint personalizzato in uno spazio specificato

1. Riprendi il tuo ambiente di sviluppo. Per ulteriori informazioni, consulta [Ripresa del lavoro in un ambiente di sviluppo](#).
2. Apri un terminale funzionante nel tuo ambiente di sviluppo.
3. (Opzionale) Installa le dipendenze necessarie per il tuo progetto se non le hai già installate. Utilizza il seguente comando :

```
yarn
```

4. Utilizzate il `--space` tag per pubblicare un'anteprima o una versione normale in uno spazio specificato. Per esempio:

- ```
yarn blueprint:preview --space my-awesome-space # publishes under a "preview" version tag to 'my-awesome-space'
```

Output di esempio:

```
Enable version 0.0.1-preview.0 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.0/projects/create
```

- ```
yarn blueprint:release --space my-awesome-space # publishes normal version to 'my-awesome-space'
```

Output di esempio:

```
Enable version 0.0.1 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1/projects/create
```

Utilizzate il `--project` per aggiungere una versione di anteprima di un blueprint personalizzato a un progetto esistente in uno spazio specificato. Per esempio:


```
yarn blueprint:preview --space my-awesome-space --project my-project # previews  
blueprint application to an existing project
```

Output di esempio:

```
Enable version 0.0.1-preview.1 at: https://codecatalyst.aws/spaces/my-awesome-  
space/blueprints  
Blueprint applied to [my-project]: https://codecatalyst.aws/spaces/my-awesome-  
space/projects/my-project/blueprints/%40amazon-codecatalyst%2FmySpace.my-blueprint/  
publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.1/add
```

Impostazione delle autorizzazioni di pubblicazione per un blueprint personalizzato

Per impostazione predefinita, l'autorizzazione di un blueprint personalizzato è abilitata se è stata generata una versione del flusso di lavoro durante la creazione del progetto. Quando le autorizzazioni di pubblicazione sono abilitate, il blueprint può essere pubblicato nello spazio. È possibile disabilitare l'autorizzazione in modo che il blueprint non possa essere pubblicato. Quando l'autorizzazione è disabilitata, il flusso di lavoro di rilascio generato durante la creazione del blueprint non viene eseguito. Le nuove modifiche a un blueprint non possono essere pubblicate a meno che le autorizzazioni del blueprint non siano abilitate.

Important

Per abilitare o disabilitare le autorizzazioni di pubblicazione di un progetto blueprint personalizzato, è necessario accedere con un account che abbia il ruolo di amministratore dello spazio o Power user nello spazio.

Per impostare le autorizzazioni di pubblicazione di un progetto blueprint

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri gestire le autorizzazioni di pubblicazione di un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.

4. Scegli la scheda Autorizzazioni di pubblicazione del progetto per visualizzare le autorizzazioni di pubblicazione per tutti i blueprint del tuo spazio.
5. Scegli il blueprint che desideri gestire, quindi scegli Abilita o Disabilita per modificare le autorizzazioni di pubblicazione. Se stai abilitando le autorizzazioni, esamina i dettagli della modifica delle autorizzazioni, quindi scegli Abilita la pubblicazione del blueprint per confermare la modifica.

Aggiungere un blueprint personalizzato a un catalogo di progetti spaziali

Dopo aver pubblicato un blueprint personalizzato nel tuo spazio, puoi aggiungerlo al catalogo dei blueprint del tuo spazio. Se aggiungi un blueprint personalizzato al catalogo dei blueprint del tuo CodeCatalyst spazio, il blueprint è disponibile per tutti i membri dello spazio e può essere utilizzato durante la creazione di un progetto o l'aggiunta a un progetto esistente. Prima di aggiungere un blueprint personalizzato al catalogo dei blueprint dello spazio, è necessario abilitare l'autorizzazione alla pubblicazione del blueprint. Se hai optato per la generazione di versioni del flusso di lavoro, le autorizzazioni di pubblicazione sono abilitate per impostazione predefinita. Per ulteriori informazioni, consulta [Impostazione delle autorizzazioni di pubblicazione per un blueprint personalizzato](#) e [Pubblicazione di un blueprint personalizzato in uno spazio](#).

Important

Per aggiungere un blueprint personalizzato al catalogo dei blueprint del tuo CodeCatalyst spazio, devi accedere con un account che abbia il ruolo di amministratore dello spazio o Power user nello spazio.

Per aggiungere un blueprint al catalogo dei blueprint dello spazio

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Il blueprint può essere aggiunto solo dal ramo predefinito del repository di origine. Se hai sviluppato il blueprint su un feature branch, unisci il feature branch con le modifiche apportate al feature branch predefinito. Crea una pull request per unire eventuali modifiche al ramo predefinito. Per ulteriori informazioni, consulta [Revisione del codice con richieste pull in Amazon CodeCatalyst](#).
3. Nella CodeCatalyst console, accedi alla dashboard dello spazio con il tuo blueprint personalizzato.

4. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
5. Scegli il nome del blueprint che desideri aggiungere, quindi scegli Aggiungi al catalogo. Se disponi di più di una versione, scegli una versione dal menu a discesa Versione del catalogo.
6. Seleziona Salva.

Modifica delle versioni del catalogo per un blueprint personalizzato

In qualità di autore di blueprint, puoi gestire la versione che desideri pubblicare nel catalogo dei blueprint dello spazio. La modifica della versione del catalogo di un blueprint non influisce sui progetti che utilizzano una versione di blueprint diversa.

Important

Per modificare le versioni del catalogo per un blueprint personalizzato nel catalogo dei blueprints del tuo CodeCatalyst spazio Amazon, devi accedere con un account con il ruolo di amministratore dello spazio o Power user nello spazio.

Per gestire una versione personalizzata del blueprint

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri modificare la versione di un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Nella tabella Space blueprint, scegli il pulsante di opzione relativo al blueprint personalizzato che desideri gestire.
5. Scegli Crea versione del catalogo, quindi scegli una versione dal menu a discesa Versione del catalogo.
6. Seleziona Salva.

Visualizzazione di dettagli, versioni e progetti di un blueprint personalizzato

Puoi visualizzare i blueprint personalizzati pubblicati dal tuo spazio, inclusi i dettagli del blueprint, le versioni e i progetti creati con o aggiungendo il blueprint.

Argomenti

- [Visualizzazione dei blueprint personalizzati di uno spazio](#)
- [Visualizzazione dei progetti creati o aggiunti con blueprint personalizzati](#)

Visualizzazione dei blueprint personalizzati di uno spazio

Per visualizzare i blueprint personalizzati di uno spazio

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri visualizzare un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprint per visualizzare i blueprint dello spazio. Nella tabella vengono visualizzati i seguenti dettagli:
 - Nome: nome del blueprint personalizzato.
 - Stato del catalogo: indica se il blueprint personalizzato è pubblicato nel catalogo dei blueprint dello spazio.
 - Versione più recente: la versione più recente del blueprint personalizzato.
 - Ultima modifica: la data dell'ultimo aggiornamento del progetto spaziale.

Visualizzazione dei progetti creati o aggiunti con blueprint personalizzati

Per visualizzare i progetti creati con o aggiunti blueprint personalizzati

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri visualizzare un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Dalla tabella Space blueprint, scegli il nome di un blueprint personalizzato per visualizzare i progetti che utilizzano i blueprint e i progetti che non utilizzano le tabelle blueprint.

Rimozione di un blueprint personalizzato da un catalogo di space blueprint

Un blueprint personalizzato può essere rimosso dal catalogo dei blueprint del tuo spazio se non desideri più che venga utilizzato per creare nuovi progetti o applicato a progetti esistenti.

 Note

Se rimuovi un blueprint personalizzato dal catalogo dei blueprint di uno spazio, ciò non influisce sul progetto creato dal blueprint o su un progetto che ha applicato il blueprint. Le risorse del blueprint non vengono rimosse dal progetto.

 Important

Per rimuovere un blueprint personalizzato dal catalogo dei blueprint del tuo CodeCatalyst spazio, devi accedere con un account che abbia il ruolo di amministratore dello spazio o Power user nello spazio.

Per rimuovere un blueprint personalizzato da un catalogo di progetti spaziali

1. [Aprire la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella CodeCatalyst console, accedi alla dashboard dello spazio con il tuo blueprint personalizzato
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Scegli il nome del progetto che desideri rimuovere, quindi scegli Rimuovi progetto dal catalogo.

Eliminazione di un blueprint o di una versione personalizzati pubblicati

Quando elimini la versione di un blueprint personalizzato o il blueprint stesso dal tuo CodeCatalyst spazio Amazon, tutti i tuoi accessi alle risorse del progetto o della versione del blueprint vengono rimossi. Dopo aver eliminato una versione del blueprint o il blueprint, i membri del progetto non saranno in grado di accedere alle risorse del progetto e tutti i flussi di lavoro richiesti dagli archivi di origine di terze parti verranno interrotti.

 Note

Se si elimina un blueprint, ciò non influisce sul progetto a cui è stato applicato il blueprint. Le risorse del blueprint non vengono rimosse dal progetto.

Se una versione del blueprint viene pubblicata nel catalogo dei blueprint dello spazio, scegli una nuova versione per il catalogo prima di eliminare la versione pubblicata.

 Important

Per eliminare un blueprint personalizzato pubblicato o una versione del catalogo di un blueprint personalizzato dal proprio spazio, è necessario accedere con un account con il ruolo di amministratore dello spazio o Power user nello spazio.

Per eliminare la versione del catalogo di un blueprint personalizzato

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri eliminare la versione del catalogo di un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Scegli il nome del blueprint con la versione del catalogo che desideri eliminare.
5. Scegli il pulsante di opzione per la versione del catalogo che desideri eliminare, quindi scegli Elimina versione.
6. Controlla i dettagli, quindi scegli un'altra versione del blueprint dal menu a discesa Scegli una nuova versione del catalogo blueprint.
7. Inserisci delete per confermare l'eliminazione della versione del catalogo dei blueprint.
8. Scegli Elimina.

Se una versione del blueprint non è presente nel catalogo dei blueprint dello spazio, puoi eliminare la versione senza sceglierne una nuova.

Per eliminare una versione personalizzata del blueprint

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri eliminare una versione del blueprint personalizzata.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Scegli il nome del blueprint con la versione che desideri eliminare.
5. Scegli il pulsante di opzione relativo alla versione che desideri eliminare, quindi scegli Elimina versione.

6. Inserisci `delete` per confermare l'eliminazione della versione del blueprint.
7. Scegli Elimina.

L'eliminazione di un blueprint dal catalogo dei blueprint dello spazio elimina tutte le versioni del blueprint. I progetti dello spazio che utilizzano il blueprint non sono interessati dall'eliminazione.

Per eliminare una versione personalizzata del blueprint

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella CodeCatalyst console, accedi allo spazio in cui desideri eliminare un blueprint personalizzato.
3. Nella dashboard dello spazio, scegli la scheda Impostazioni, quindi scegli Blueprints.
4. Nella tabella Space blueprint, scegli il pulsante di opzione relativo al blueprint personalizzato che desideri eliminare, quindi scegli Elimina blueprint.
5. Inserisci `delete` per confermare l'eliminazione del blueprint personalizzato.
6. Scegli Delete (Elimina).

Gestione delle dipendenze, delle mancate corrispondenze e degli strumenti

Una gestione errata delle dipendenze può portare a errori di compilazione e problemi di runtime per chi utilizza i tuoi blueprint personalizzati. Strumenti e componenti obsoleti possono impedire agli utenti del blueprint di accedere alle funzionalità e alle correzioni di bug più recenti. È possibile gestire le dipendenze, gestire le discrepanze tra le dipendenze e aggiornare strumenti e componenti per garantire che tutte le dipendenze si basino sulle stesse versioni dei componenti e sulla sincronizzazione degli stessi componenti.

Argomenti

- [Aggiungere dipendenze](#)
- [Gestione delle discrepanze tra i tipi di dipendenza](#)
- [Usando yarn e npm](#)
- [Aggiornamento di strumenti e componenti](#)

Aggiungere dipendenze

In qualità di autore del blueprint, potrebbe essere necessario aggiungere pacchetti al blueprint, ad esempio. `@amazon-codecatalyst/blueprint-component.environments` [È necessario aggiornare il `projen.ts` file con quel pacchetto e quindi rigenerare la configurazione del progetto con Projen.](#) Projen funge da modello di progetto per ogni codebase di blueprint, il che offre la possibilità di inviare aggiornamenti degli strumenti compatibili con le versioni precedenti modificando il modo in cui il modello rende i file di configurazione. Il `package.json` file è un file parzialmente di proprietà del modello Projen. Projen riconosce le versioni di dipendenza incluse nel file `package.json`, ma altre opzioni devono provenire dal modello.

Per aggiungere una dipendenza e aggiornare un file **`projenrc.ts`**

1. Nel `projen.ts` file, vai alla sezione `deps`.
2. Aggiungi la dipendenza che desideri utilizzare nel tuo blueprint.
3. Usa il seguente comando per rigenerare la configurazione del tuo progetto:

```
yarn projen && yarn
```

Gestione delle discrepanze tra i tipi di dipendenza

Dopo un aggiornamento di [Yarn](#), potresti ricevere il seguente errore relativo a un parametro del repository:

```
Type 'SourceRepository' is missing the following properties from type  
'SourceRepository': synthesisSteps, addSynthesisStep
```

L'errore è dovuto a una mancata corrispondenza delle dipendenze che si verifica quando un componente si basa su una versione più recente di un altro componente, ma il componente di riferimento è collegato a una versione precedente. L'errore può essere corretto facendo in modo che tutti i componenti si basino sulla stessa versione in modo che la versione sia sincronizzata tra di loro. È meglio mantenere tutti i pacchetti forniti da blueprint nella stessa versione più recente (`0.0.x`), a meno che tu non sia sicuro di come gestisci le versioni. L'esempio seguente mostra come il `package.json` file può essere configurato in modo che tutte le dipendenze si basino sulla stessa versione:

```
...  
"@caws-blueprint-component/caws-environments": "^0.1.12345",
```



```
"@caws-blueprint-component/caws-source-repositories": "^0.1.12345",  
"@caws-blueprint-component/caws-workflows": "^0.1.12345",  
"@caws-blueprint-component/caws-workspaces": "^0.1.12345",  
"@caws-blueprint-util/blueprint-utils": "^0.1.12345",  
...  
"@caws-blueprint/blueprints.blueprint": "*",
```

Dopo aver configurato le versioni per tutte le dipendenze, utilizzate il seguente comando:

```
yarn install
```

Usando yarn e npm

I progetti utilizzano [Yarn](#) per gli utensili. L'uso di [npm](#) e Yarn causerà problemi di strumenti perché il modo in cui gli alberi delle dipendenze vengono risolti da ciascuno è diverso. Per evitare tali problemi, è meglio usare solo Yarn.

Se hai installato accidentalmente delle dipendenze usando npm, puoi rimuovere il `package-lock.json` file generato e assicurarti che il `.projenrc.ts` file sia aggiornato con le dipendenze necessarie. Rigeneri la configurazione del tuo progetto con Projen.

Utilizzate quanto segue per rigenerare dal modello:

```
yarn projen
```

Dopo esserti assicurato che il file `.projenrc.ts` sia aggiornato con le dipendenze necessarie, usa il seguente comando:

```
yarn
```

Aggiornamento di strumenti e componenti

Occasionalmente, potresti voler aggiornare gli strumenti e i componenti per rendere disponibili nuove funzionalità. Ti consigliamo di mantenere tutti i componenti sulla stessa versione a meno che tu non sia sicuro di come gestisci le versioni. Le versioni sono sincronizzate tra i componenti, quindi le stesse versioni per tutti i componenti garantiscono la corretta dipendenza tra di essi.

Utilizzo di Yarn workspace monorepo

Usa il seguente comando per aggiornare utilità e componenti dalla radice del repository di un blueprint personalizzato:

```
yarn upgrade @amazon-codecatalyst/*
```

Usa il seguente comando se non stai usando un monorepo:

```
yarn upgrade --pattern @amazon-codecatalyst/*
```

Altre opzioni che puoi usare per aggiornare strumenti e componenti:

- Usa `npm view @caws-blueprint-component/<some-component>` per ottenere l'ultima versione.
- Passa manualmente alla versione più recente impostando la versione nel file `package.json` e utilizzando il seguente comando: `yarn`. Tutti i componenti e le utilità devono avere la stessa versione.

Contribuisci

Il Blueprints Software Development Kit (SDK) è una libreria open source a cui puoi contribuire. In qualità di collaboratore, considera le linee guida per il contributo, il feedback e i difetti. Per ulteriori informazioni, consulta il repository dei [blueprints GitHub](#).

Quote per i progetti in CodeCatalyst

La tabella seguente descrive le quote e i limiti per i blueprint in Amazon. CodeCatalyst Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Numero massimo di blueprint applicati per progetto CodeCatalyst	100

Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst

CodeCatalyst i repository di origine sono repository Git ospitati su Amazon. CodeCatalyst Puoi utilizzare i repository di origine per CodeCatalyst archiviare, modificare e gestire in modo sicuro le risorse di un progetto.

Le risorse in un CodeCatalyst repository possono includere:

- documenti
- codice sorgente
- file binari

CodeCatalyst utilizza inoltre l'archivio di origine di un progetto per archiviare le informazioni di configurazione del progetto, ad esempio i file di configurazione del flusso di lavoro.

È possibile avere più di un archivio di origine in un CodeCatalyst progetto. Ad esempio, potresti voler disporre di repository di sorgenti separati per il codice sorgente front-end, il codice sorgente di back-end, le utilità e la documentazione.

Ecco un possibile flusso di lavoro per lavorare con il codice nei repository di sorgenti, nelle richieste pull e negli ambienti di sviluppo in: CodeCatalyst

Mary Major crea un progetto di applicazione web CodeCatalyst utilizzando un blueprint, che crea un repository di sorgenti contenente codice di esempio. Invita i suoi amici Li Juan, Saanvi Sarkar e Jorge Souza a lavorare al progetto con lei. Li Juan esamina il codice di esempio nel repository dei sorgenti e decide di apportare alcune rapide modifiche per aggiungere un test al codice. Li crea un ambiente di sviluppo, sceglie AWS Cloud9 come IDE e specifica un nuovo ramo, `test-code`. Si apre l'ambiente di sviluppo. Li aggiunge rapidamente il codice, quindi esegue il commit e inserisce il ramo con le modifiche al repository dei sorgenti. CodeCatalyst Li crea quindi una pull request. Come parte della creazione di quella pull request, Li aggiunge Jorge Souza e Saanvi Sarkar come revisori per garantire che il codice venga esaminato.

Durante la revisione del codice, Jorge Souza ricorda di avere un archivio di progetti tutto suo contenente un prototipo dell'app su GitHub cui stanno lavorando. Chiede a Mary Major di installare e configurare l'estensione che gli permetterà di collegare il GitHub repository al progetto come archivio

di sorgenti aggiuntivo. Mary esamina il repository GitHub e collabora con Jorge per configurare l' GitHub estensione in modo che possa collegare il GitHub repository come archivio di sorgenti aggiuntivo per il progetto.

CodeCatalyst I repository di sorgenti supportano le funzionalità standard di Git e funzionano con gli strumenti basati su Git esistenti. È possibile creare e utilizzare token di accesso personali (PATs) come password specifica per l'applicazione durante la clonazione e l'utilizzo di repository di sorgenti da un client Git o da ambienti di sviluppo integrati (). IDEs Questi sono associati alla tua identità PATs utente. CodeCatalyst Per ulteriori informazioni, consulta [Concedi agli utenti l'accesso al repository con token di accesso personali](#).

CodeCatalyst i repository di origine supportano le richieste pull. Questo è un modo semplice per te e per gli altri membri del progetto di rivedere e commentare le modifiche al codice prima di unirle da un ramo all'altro. Puoi visualizzare le modifiche nella CodeCatalyst console e commentare le righe di codice.

Gli invii alle filiali in un repository di CodeCatalyst origine possono avviare automaticamente un'esecuzione in un flusso di lavoro, in cui le modifiche possono essere create, testate e distribuite. Se il tuo repository di origine è stato creato come parte di un progetto utilizzando un modello di progetto, uno o più flussi di lavoro sono configurati per te come parte del progetto. Puoi aggiungere flussi di lavoro aggiuntivi per i repository in qualsiasi momento. I file di configurazione YAML per i flussi di lavoro in un progetto vengono archiviati nei repository di origine configurati nell'azione di origine per tali flussi di lavoro. Per ulteriori informazioni, consulta [Guida introduttiva ai flussi di lavoro](#).

Argomenti

- [Concetti relativi al repository di origine](#)
- [Configurazione per l'utilizzo dei repository di origine](#)
- [Guida introduttiva agli archivi di CodeCatalyst origine e al modello di applicazione a pagina singola](#)
- [Memorizzazione del codice sorgente nei repository per un progetto in CodeCatalyst](#)
- [Organizzazione del codice sorgente, lavoro con le filiali di Amazon CodeCatalyst](#)
- [Gestione dei file di codice sorgente in Amazon CodeCatalyst](#)
- [Revisione del codice con richieste pull in Amazon CodeCatalyst](#)
- [Comprendere le modifiche al codice sorgente con i commit in Amazon CodeCatalyst](#)
- [Quote per i repository di sorgenti in CodeCatalyst](#)

Concetti relativi al repository di origine

Ecco alcuni concetti da conoscere quando lavori con i repository dei CodeCatalyst sorgenti.

Argomenti

- [Progetti](#)
- [Archivi di origine](#)
- [Ambienti di sviluppo](#)
- [Token di accesso personali \(\) PATs](#)
- [Rami](#)
- [Rami predefiniti](#)
- [Commit](#)
- [Richieste pull](#)
- [Revisioni](#)
- [Flussi di lavoro](#)

Progetti

Un progetto rappresenta uno sforzo collaborativo CodeCatalyst che supporta i team e le attività di sviluppo. Dopo aver creato un progetto, puoi aggiungere, aggiornare o rimuovere utenti e risorse, personalizzare la dashboard del progetto e monitorare l'avanzamento del lavoro del team. Puoi avere più progetti all'interno di uno spazio.

Gli archivi di origine sono specifici per i progetti in cui li crei o li colleghi in uno spazio. Non puoi condividere un repository tra progetti e non puoi collegare un repository a più di un progetto in uno spazio. Gli utenti con il ruolo di collaboratore o amministratore di progetto in un progetto possono interagire con gli archivi di origine associati a quel progetto in base alle autorizzazioni concesse a tali ruoli. Per ulteriori informazioni, consulta [Concessione dell'accesso con ruoli utente](#).

Archivi di origine

Un repository di sorgenti consente di archiviare in modo sicuro codice e file per il progetto. Memorizza anche la cronologia delle versioni dei file. Per impostazione predefinita, un archivio dei sorgenti viene condiviso con gli altri utenti del CodeCatalyst progetto. Puoi avere più di un archivio di sorgenti per un progetto. È possibile creare archivi di sorgenti per progetti in CodeCatalyst oppure scegliere di

collegare un archivio di sorgenti esistente ospitato da un altro servizio se tale servizio è supportato da un'estensione installata. Ad esempio, è possibile collegare un GitHub repository a un progetto dopo aver installato l'GitHub estensione Repositories. Per ulteriori informazioni, consultare [Memorizzazione del codice sorgente nei repository per un progetto in CodeCatalyst](#) e [Guida rapida: installazione di estensioni, connessione di provider e collegamento di risorse in CodeCatalyst](#).

Ambienti di sviluppo

Un ambiente di sviluppo è un ambiente di sviluppo basato su cloud che puoi utilizzare CodeCatalyst per lavorare rapidamente sul codice archiviato nei repository di origine del tuo progetto. Gli strumenti di progetto e le librerie di applicazioni inclusi nel tuo ambiente di sviluppo sono definiti da un devfile nel repository di origine del progetto. Se non avete un devfile nel vostro repository di origine, verrà applicato automaticamente un devfile predefinito. Il devfile predefinito include strumenti per i linguaggi e i framework di programmazione più utilizzati. Per impostazione predefinita, un ambiente di sviluppo è configurato per avere un processore a 2 core, 4 GB di RAM e 16 GiB di storage persistente.

Puoi scegliere di clonare un ramo esistente del tuo repository di origine nel tuo ambiente di sviluppo oppure puoi scegliere di creare un nuovo ramo come parte della creazione dell'ambiente di sviluppo.

Token di accesso personali () PATs

Un token di accesso personale (PAT) è simile a una password. È associato all'identità utente e può essere utilizzato in tutti gli spazi e i progetti di CodeCatalyst. Si utilizza PATs per accedere a CodeCatalyst risorse che includono ambienti di sviluppo integrati (IDEs) e repository di sorgenti basati su Git. PATs ti rappresentano CodeCatalyst e puoi gestirle nelle tue impostazioni utente. Un utente può avere più di un PAT. I token di accesso personali vengono visualizzati solo una volta. Come procedura consigliata, assicurati di archivarli in modo sicuro sul tuo computer locale. Per impostazione predefinita, PATs scadono dopo un anno.

Quando si lavora con ambienti di sviluppo integrati (IDEs), PATs sono l'equivalente di una password Git. Fornisci il PAT quando ti viene richiesta una password quando configuri il tuo IDE per funzionare con un repository Git. Per ulteriori informazioni su come connettere il tuo IDE con un repository basato su Git, consulta la documentazione del tuo IDE.

Rami

Un ramo è un puntatore o un riferimento a un commit in Git e in CodeCatalyst. Puoi usare le filiali per organizzare il tuo lavoro. Ad esempio, puoi utilizzare i rami per lavorare su una versione nuova

o diversa dei file senza influire sui file in altri rami. Puoi utilizzare i rami per sviluppare nuove funzionalità, archiviare una versione specifica del tuo progetto e altro ancora. Un repository di origine può avere uno o più rami. Quando si crea un progetto utilizzando un modello, il repository di origine creato per il progetto contiene file di esempio in un ramo chiamato main. Il ramo principale è il ramo predefinito per il repository.

Rami predefiniti

I repository di origine CodeCatalyst hanno un ramo predefinito indipendentemente da come li crei. Se scegliete di creare un progetto utilizzando un modello, l'archivio di origine creato per quel progetto include un file README.md oltre a codice di esempio, definizioni del flusso di lavoro e altre risorse. Se si crea un repository di origine senza utilizzare un modello, viene aggiunto automaticamente un file README.md come primo commit e viene creato automaticamente un ramo predefinito come parte della creazione del repository. Questo ramo predefinito è denominato main. Questo ramo predefinito è quello utilizzato come ramo di base o predefinito nei repository locali (repos) quando gli utenti clonano il repository. È possibile modificare il ramo utilizzato come ramo predefinito. Per ulteriori informazioni, consulta [Gestione del ramo predefinito per un repository](#).

Non è possibile eliminare il ramo predefinito per un repository di origine. I risultati della ricerca includono solo i risultati del ramo predefinito.

Commit

Un commit è una modifica a un file o a un set di file. Nella CodeCatalyst console Amazon, un commit salva le modifiche e le invia a un repository di origine. Il commit include informazioni sulla modifica, tra cui l'identità dell'utente che ha apportato la modifica, l'ora e la data della modifica, il titolo del commit e qualsiasi messaggio incluso sulla modifica. Per ulteriori informazioni, consulta [Comprendere le modifiche al codice sorgente con i commit in Amazon CodeCatalyst](#).

Nel contesto di un repository di origine in CodeCatalyst, i commit sono istantanee del contenuto e delle modifiche apportate al contenuto del repository. Puoi anche aggiungere tag Git ai commit, per identificare commit specifici.

Richieste pull

Una pull request è il modo principale in cui tu e gli altri utenti esaminate, commentate e unite le modifiche al codice da un ramo all'altro in un repository di sorgenti. È possibile utilizzare le richieste pull per esaminare in modo collaborativo le modifiche al codice in caso di modifiche o correzioni

minori, aggiunte di funzionalità importanti o nuove versioni del software rilasciato. In una pull request, puoi esaminare le modifiche tra i rami di origine e di destinazione o le differenze tra le revisioni di tali rami. È possibile aggiungere commenti alle singole righe di modifiche al codice e commenti sulla pull request in generale.

Tip

Durante la creazione di una pull request, la differenza visualizzata è la differenza tra la punta del ramo di origine e la punta del ramo di destinazione. Una volta creata la pull request, la differenza visualizzata sarà tra la revisione della pull request scelta e il commit che costituiva la punta del ramo di destinazione quando è stata creata la pull request. Per ulteriori informazioni sulle differenze e sulle basi di unione in Git, consulta [git-merge-base](#) la documentazione di Git.

Revisioni

Una revisione è una versione aggiornata di una pull request. Ogni push al ramo sorgente di una pull request crea una revisione che contiene le modifiche apportate nei commit inclusi in quel push. È possibile visualizzare le differenze tra le revisioni di una pull request oltre alle differenze tra i rami di origine e di destinazione. Per ulteriori informazioni, consulta [Revisione del codice con richieste pull in Amazon CodeCatalyst](#).

Flussi di lavoro

Un flusso di lavoro è una procedura automatizzata che descrive come creare, testare e distribuire il codice come parte di un sistema di integrazione e distribuzione continua (CI/CD). Un flusso di lavoro definisce una serie di passaggi, o azioni, da eseguire durante l'esecuzione di un flusso di lavoro. Un flusso di lavoro definisce anche gli eventi, o trigger, che causano l'avvio del flusso di lavoro. Per configurare un flusso di lavoro, è necessario creare un file di definizione del flusso di lavoro utilizzando l'editor [visivo o YAML](#) della CodeCatalyst console.

Tip

Per una rapida occhiata a come potresti utilizzare i flussi di lavoro in un progetto, [crea un progetto con un blueprint](#). Ogni blueprint implementa un flusso di lavoro funzionante che puoi rivedere, eseguire e sperimentare.

Un repository di origine può anche archiviare i file di configurazione e altre informazioni per flussi di lavoro, notifiche, problemi e altre informazioni di configurazione per il progetto. I file di configurazione vengono creati e archiviati nell'archivio di origine quando si creano risorse che richiedono file di configurazione o quando si specifica il repository come azione di origine per un flusso di lavoro. Se crei un progetto da un blueprint, i file di configurazione saranno già archiviati nel repository di origine creato per voi come parte del progetto. Queste informazioni di configurazione sono memorizzate in una cartella denominata `.codecatalyst` nel ramo predefinito del repository. Ogni volta che crei un ramo del ramo predefinito, crei una copia di questa cartella e della sua configurazione oltre a tutti gli altri file e cartelle del ramo.

Configurazione per l'utilizzo dei repository di origine

Quando lavori con i repository di sorgenti in Amazon CodeCatalyst sul tuo computer locale, puoi usare Git da solo o in un ambiente di sviluppo integrato (IDE) supportato per apportare modifiche al codice e inviare ed estrarre il codice. Come best practice, ti consigliamo di utilizzare le versioni più recenti di Git e altri software.

Note

Se usi Dev Environments, non devi installare Git. Una versione recente di Git è inclusa nel tuo ambiente di sviluppo.

Informazioni sulla compatibilità delle versioni per CodeCatalyst

Componente	Versione
Git	più recente

Installa Git

Per lavorare con file, commit, branch e altre informazioni nei repository di sorgenti da un client Git senza un IDE, installa Git sul tuo computer locale.

Per installare Git, consigliamo siti web come [Git Downloads](#).

Crea un token di accesso personale

Per clonare i repository di origine sul computer locale o sull'IDE preferito, è necessario creare un token di accesso personale (PAT).

Per creare un token di accesso personale (PAT)

1. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri relativa a un progetto o a uno spazio e scegliendo il tuo nome dall'elenco dei membri.

2. Nel nome PAT, inserisci un nome descrittivo per il tuo PAT.
3. In Data di scadenza, lascia la data predefinita o scegli l'icona del calendario per selezionare una data personalizzata. La data di scadenza predefinita è un anno dalla data corrente.
4. Scegli Create (Crea).

Puoi anche creare questo token quando scegli Clone repository come repository di origine.

5. Salva il segreto PAT in un luogo sicuro.

Important

Il segreto PAT viene visualizzato solo una volta. Non è possibile recuperarlo dopo aver chiuso la finestra.

Guida introduttiva agli archivi di CodeCatalyst origine e al modello di applicazione a pagina singola

Segui i passaggi di questo tutorial per imparare a lavorare con i repository di sorgenti in Amazon CodeCatalyst.

Il modo più rapido per iniziare a lavorare con i repository di sorgenti in Amazon CodeCatalyst è creare un progetto utilizzando un modello. Quando crei un progetto utilizzando un modello, le risorse vengono create automaticamente, incluso un archivio di sorgenti che include codice di esempio. Puoi usare questo repository e questo esempio di codice per imparare a:

- Visualizza gli archivi dei sorgenti di un progetto e sfogliale il contenuto
- Crea un ambiente di sviluppo con una nuova filiale in cui puoi lavorare sul codice
- Modifica un file, esegui il commit e invia le modifiche
- Crea una pull request e rivedi le modifiche al codice con altri membri del progetto
- Visualizza il flusso di lavoro del tuo progetto, crea e testa automaticamente le modifiche nel ramo sorgente della pull request
- Unisci le modifiche dal ramo di origine al ramo di destinazione e chiudi la pull request
- Guarda le modifiche unite create e distribuite automaticamente

Per ottenere il massimo da questo tutorial, invita altre persone al tuo progetto in modo da poter lavorare insieme su una pull request. Puoi anche esplorare le funzionalità aggiuntive di CodeCatalyst, come la creazione di problemi e la loro associazione a una pull request, o la configurazione di notifiche e la ricezione di avvisi quando viene eseguito il flusso di lavoro associato. Per un'esplorazione completa di, vedi. CodeCatalyst [Tutorial introduttivi](#)

Creazione di un progetto con un blueprint

La creazione di un progetto è il primo passo per poter lavorare insieme. Puoi utilizzare un blueprint per creare il tuo progetto, che creerà anche un repository di sorgenti con codice di esempio e un flusso di lavoro che genererà e distribuirà automaticamente il codice quando lo modifichi. In questo tutorial, ti illustreremo un progetto creato con il blueprint dell'applicazione a pagina singola, ma puoi seguire le procedure per qualsiasi progetto con un repository di sorgenti. Assicurati di scegliere un ruolo IAM o di aggiungere un ruolo IAM se non ne hai uno durante la creazione del progetto. Ti consigliamo di utilizzare il ruolo `CodeCatalystWorkflowDevelopmentRole-spaceName` di servizio per questo progetto.

Se hai già un progetto, puoi passare direttamente a [Visualizzazione degli archivi di un progetto](#).

Note

Solo gli utenti con il ruolo di amministratore dello spazio o utente esperto possono creare progetti in CodeCatalyst. Se non ricopri questo ruolo e hai bisogno di un progetto su cui lavorare per questo tutorial, chiedi a qualcuno con uno di questi ruoli di creare un progetto per te e aggiungerti al progetto creato. Per ulteriori informazioni, consulta [Concessione dell'accesso con ruoli utente](#).

Per creare un progetto con un blueprint

1. Nella CodeCatalyst console, accedi allo spazio in cui desideri creare un progetto.
2. Nella dashboard dello spazio, scegli Crea progetto.
3. Scegli Inizia con un progetto.

Tip

Puoi scegliere di aggiungere un blueprint fornendo ad Amazon Q i requisiti del tuo progetto in modo che Amazon Q ti suggerisca un blueprint. Per ulteriori informazioni, consultare [Utilizzo di Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di funzionalità](#) e [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#). Questa funzionalità è disponibile solo nella regione Stati Uniti occidentali (Oregon).

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

4. Dalla scheda CodeCatalyst Blueprint o Space blueprint, scegli un blueprint, quindi scegli Avanti.
5. In Assegna un nome al progetto, inserisci il nome che desideri assegnare al progetto e i nomi delle risorse associate. Il nome deve essere univoco all'interno del tuo spazio.
6. (Facoltativo) Per impostazione predefinita, il codice sorgente creato dal blueprint viene archiviato in un CodeCatalyst repository. In alternativa, è possibile scegliere di archiviare il codice sorgente del blueprint in un repository di terze parti. Per ulteriori informazioni, consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#).

Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo da CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Come best practice, assicuratevi sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Effettua una delle seguenti operazioni a seconda del provider di repository di terze parti che desideri utilizzare:

- GitHub repository: Connect a un GitHub account.

Scegli il menu a discesa Avanzato, scegli GitHub come provider del repository, quindi scegli l'GitHub account in cui desideri archiviare il codice sorgente creato dal blueprint.

 Note

Se stai collegando un GitHub account, devi creare una connessione personale per stabilire una mappatura delle identità tra la tua identità e la tua CodeCatalyst identità. GitHub Per ulteriori informazioni, consultare [Connessioni personali](#) e [Accesso alle GitHub risorse con connessioni personali](#).

- Archivi Bitbucket: Connect uno spazio di lavoro Bitbucket.

Scegli il menu a discesa Avanzato, scegli Bitbucket come provider di repository, quindi scegli l'area di lavoro Bitbucket in cui desideri archiviare il codice sorgente creato dal blueprint.

- GitLab repository: Connect a un GitLab utente.

Scegli il menu a discesa Avanzato, scegli GitLab come provider del repository, quindi scegli l'GitLab utente in cui desideri archiviare il codice sorgente creato dal blueprint.

7. In Risorse del progetto, configura i parametri del blueprint. A seconda del blueprint, potresti avere la possibilità di assegnare un nome al repository di origine.
8. (Facoltativo) Per visualizzare i file di definizione con aggiornamenti basati sulle selezioni dei parametri di progetto effettuate, scegliete Visualizza codice o Visualizza flusso di lavoro da Genera anteprima del progetto.
9. (Facoltativo) Scegli Visualizza dettagli dalla scheda del blueprint per visualizzare dettagli specifici sul blueprint, come una panoramica dell'architettura del blueprint, le connessioni e le autorizzazioni richieste e il tipo di risorse create dal blueprint.
10. Seleziona Crea progetto.

La pagina di panoramica del progetto viene visualizzata non appena si crea un progetto o si accetta un invito a partecipare a un progetto e si completa la procedura di accesso. La pagina di panoramica del progetto per un nuovo progetto non contiene problemi aperti o richieste pull. Facoltativamente,

puoi scegliere di creare un problema e assegnarlo a te stesso. Puoi anche scegliere di invitare qualcun altro al tuo progetto. Per ulteriori informazioni, consultare [Creazione di un problema in CodeCatalyst](#) e [Invitare un utente a un progetto](#).

Visualizzazione degli archivi di un progetto

Come membro di un progetto, puoi visualizzare i repository dei sorgenti del progetto. Puoi anche scegliere di creare repository aggiuntivi. Se qualcuno con il ruolo di amministratore di Space ha installato e configurato i GitHub repository, i repository Bitbucket o l'estensione dei GitLab repository, puoi anche aggiungere collegamenti a repository di terze parti negli GitHub account, nelle aree di lavoro Bitbucket o negli utenti configurati per l'estensione. GitLab Per ulteriori informazioni, consultare [Creazione di un repository di sorgenti](#) e [Guida rapida: installazione di estensioni, connessione di provider e collegamento di risorse in CodeCatalyst](#).

Note

Per i progetti creati con il blueprint dell'applicazione a pagina singola, il nome predefinito per l'archivio di origine che contiene il codice di esempio è. ***spa-app***

Per accedere agli archivi di origine di un progetto

1. Passate al progetto ed effettuate una delle seguenti operazioni:
 - Nella pagina di riepilogo del progetto, scegli il repository che desideri dall'elenco, quindi scegli Visualizza archivio.
 - Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Nei repository di origine, scegli il nome del repository dall'elenco. È possibile filtrare l'elenco dei repository digitando parte del nome del repository nella barra dei filtri.
2. Nella home page del repository, visualizza il contenuto del repository e le informazioni sulle risorse associate, come il numero di richieste pull e i flussi di lavoro. Per impostazione predefinita, vengono visualizzati i contenuti del ramo predefinito. È possibile modificare la visualizzazione scegliendo un ramo diverso dall'elenco a discesa.

La pagina di panoramica del repository include informazioni sui flussi di lavoro e le richieste pull configurate per i rami di questo repository e i relativi file. Se hai appena creato il progetto, i flussi di lavoro iniziali per creare, testare e distribuire il codice continueranno a essere in esecuzione, poiché il completamento richiede alcuni minuti. È possibile visualizzare i flussi di lavoro correlati e il relativo

stato scegliendo il numero sotto Flussi di lavoro correlati, ma in questo modo verrà aperta la pagina Flussi di lavoro in CI/CD. Per questo tutorial, rimani sulla pagina di panoramica ed esplora il codice nel repository. Il contenuto del README .md file viene visualizzato in questa pagina sotto i file del repository. In File, viene mostrato il contenuto del ramo predefinito. Puoi modificare la visualizzazione del file per mostrare il contenuto di un altro ramo, se ne hai uno. La `.codecatalyst` cartella contiene il codice utilizzato per altre parti del progetto, come i file YAML del flusso di lavoro.

Per visualizzare il contenuto delle cartelle, scegliete la freccia accanto al nome della cartella per espanderla. Ad esempio, scegliete la freccia accanto `src` a per visualizzare i file per l'applicazione Web a pagina singola contenuta in quella cartella. Per visualizzare il contenuto di un file, sceglierlo nell'elenco. Si aprirà Visualizza file, in cui è possibile sfogliare il contenuto di più file. Puoi modificare anche singoli file nella console, ma per modificare più file, ti consigliamo di creare un ambiente di sviluppo.

Creazione di un ambiente di sviluppo

Puoi aggiungere e modificare file in un repository di origine nella CodeCatalyst console Amazon. Tuttavia, per lavorare in modo efficace con più file e rami, ti consigliamo di utilizzare un ambiente di sviluppo o di clonare il repository sul tuo computer locale. In questo tutorial, creeremo un ambiente di AWS Cloud9 sviluppo con un ramo denominato. **develop** Puoi scegliere un nome di ramo diverso, ma assegnando un nome al ramo **develop**, verrà eseguito automaticamente un flusso di lavoro per creare e testare il codice quando creerai una pull request più avanti in questo tutorial.

Tip

Se decidi di clonare un repository localmente anziché o in aggiunta all'utilizzo di un ambiente di sviluppo, assicurati di avere Git sul tuo computer locale o che il tuo IDE includa Git. Per ulteriori informazioni, consulta [Configurazione per l'utilizzo dei repository di origine](#).

Per creare un ambiente di sviluppo con un nuovo ramo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri creare un ambiente di sviluppo.
3. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, scegli Archivi di origine e scegli il repository per il quale desideri creare un ambiente di sviluppo.

4. Nella home page del repository, scegli Crea ambiente di sviluppo.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Scegli il repository da clonare, scegli Lavora in un nuovo ramo, inserisci il nome di un ramo nel campo Nome del ramo e scegli un ramo dal quale creare il nuovo ramo dal menu a discesa Crea ramo da.
7. Facoltativamente, aggiungi un alias per Dev Environment.
8. Facoltativamente, scegli il pulsante di modifica della configurazione dell'ambiente di sviluppo per modificare la configurazione di calcolo, archiviazione o timeout dell'ambiente di sviluppo.
9. Scegli Create (Crea). Durante la creazione dell'ambiente di sviluppo, la colonna di stato dell'ambiente di sviluppo mostrerà Avvio e la colonna di stato verrà visualizzata in esecuzione una volta creato l'ambiente di sviluppo. Si aprirà una nuova scheda con il tuo ambiente di sviluppo nell'IDE di tua scelta. Puoi modificare il codice, eseguire il commit e inviare le modifiche.

Dopo aver creato l'ambiente di sviluppo, puoi modificare i file, eseguire il commit delle modifiche e inviarle al **test** ramo. Per questo tutorial, modifica il contenuto tra i `<p>` tag nel `App.tsx` file nella `src` cartella per modificare il testo visualizzato sulla pagina web. Conferma e invia la modifica, quindi torna alla CodeCatalyst scheda.

Per apportare e promuovere una modifica da un ambiente di AWS Cloud9 sviluppo

1. In AWS Cloud9, espandi il menu di navigazione laterale per sfogliare i file. `src` Espandi e apri `App.tsx`.
2. Modifica il testo all'interno dei `<p>` tag.
3. Salva il file, quindi esegui il commit e invia le modifiche utilizzando il menu Git. In alternativa, nella finestra del terminale, esegui il commit e invia le modifiche con i `git push` comandi `git commit and`.

```
git commit -am "Making an example change"
git push
```

 Tip

Potrebbe essere necessario modificare le directory del terminale nella directory del repository Git prima di poter eseguire correttamente i comandi Git.

Creazione di una richiesta di pull

Puoi utilizzare le pull request per esaminare in modo collaborativo le modifiche al codice per individuare modifiche o correzioni minori, aggiunte di funzionalità importanti o nuove versioni del software rilasciato. In questo tutorial, creerai una pull request per rivedere le modifiche apportate al **test** ramo rispetto al ramo principale. La creazione di una pull request in un progetto creato con un modello avvierà anche un'esecuzione degli eventuali flussi di lavoro associati.

Come creare una richiesta pull

1. Vai al tuo progetto.
2. Esegui una delle seguenti operazioni:
 - Nel riquadro di navigazione, scegli Codice, scegli Richieste pull, quindi scegli Crea richiesta pull.
 - Nella home page del repository, scegli Altro, quindi scegli Crea richiesta pull.
 - Nella pagina del progetto, scegli Crea richiesta pull.
3. Nel repository Source, assicurati che il repository di origine specificato sia quello che contiene il codice commesso. Questa opzione appare solo se non hai creato la pull request dalla pagina principale del repository.
4. Nel ramo di destinazione, scegli il ramo in cui unire il codice dopo averlo esaminato.
5. Nel ramo di origine, scegli il ramo che contiene il codice commesso.
6. In Pull request title, inserisci un titolo che aiuti gli altri utenti a capire cosa deve essere esaminato e perché.
7. (Facoltativo) Nella descrizione della richiesta Pull, fornisci informazioni come un link ai problemi o una descrizione delle modifiche.

Tip

Puoi scegliere Write description for me per generare CodeCatalyst automaticamente una descrizione delle modifiche contenute nella pull request. Puoi apportare modifiche alla descrizione generata automaticamente dopo averla aggiunta alla pull request. Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio e non è disponibile per le richieste pull negli archivi collegati. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

8. (Facoltativo) In Problemi, scegli Problemi di collegamento, quindi scegli un problema dall'elenco o inserisci il relativo ID. Per scollegare un problema, scegli l'icona di scollegamento.
9. (Facoltativo) In Revisori obbligatori, scegli Aggiungi revisori richiesti. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori obbligatori devono approvare le modifiche prima che la pull request possa essere unita al ramo di destinazione.

 Note

Non è possibile aggiungere un revisore sia come revisore obbligatorio che come revisore opzionale. Non puoi aggiungerti come revisore.

10. (Facoltativo) In Revisori opzionali, scegli Aggiungi revisori opzionali. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori opzionali non devono approvare le modifiche come requisito prima che la pull request possa essere unita al ramo di destinazione.
11. Esamina le differenze tra le filiali. La differenza mostrata in una pull request è rappresentata dalle modifiche tra la revisione nel ramo di origine e la base di unione, che è l'head commit del ramo di destinazione al momento della creazione della pull request. Se non viene visualizzata alcuna modifica, i rami potrebbero essere identici oppure potresti aver scelto lo stesso ramo sia per l'origine che per la destinazione.
12. Quando sei sicuro che la pull request contenga il codice e le modifiche che desideri rivedere, scegli Crea.

 Note

Dopo aver creato la pull request, puoi aggiungere commenti. I commenti possono essere aggiunti alla pull request o alle singole righe dei file, nonché alla pull request complessiva. È possibile aggiungere collegamenti a risorse, ad esempio file, utilizzando il segno @ seguito dal nome del file.

È possibile visualizzare le informazioni sui flussi di lavoro associati avviati dalla creazione di questa pull request scegliendo Overview e quindi esaminando le informazioni nell'area dei dettagli della richiesta Pull in Workflow runs. Per visualizzare l'esecuzione del workflow, scegliete l'esecuzione.

 Tip

Se alla filiale hai assegnato un nome diverso **develop**, non verrà eseguito automaticamente un flusso di lavoro per creare e testare le modifiche. Se desideri configurarlo, modifica il file YAML per il `onPullRequestBuildAndTest` flusso di lavoro. Per ulteriori informazioni, consulta [Creazione di un flusso di lavoro](#).

Puoi commentare questa pull request e chiedere agli altri membri del progetto di commentarla. Puoi anche scegliere di aggiungere o modificare revisori facoltativi o obbligatori. Puoi scegliere di apportare ulteriori modifiche al ramo di origine del repository e vedere come tali modifiche confermate creano revisioni per la pull request. Per ulteriori informazioni, vedere [Revisione di una richiesta di pull](#), [Aggiornamento di una richiesta di pull](#), [Revisione del codice con richieste pull in Amazon CodeCatalyst](#), e [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

Unione di una richiesta di pull

Una volta che una pull request è stata esaminata e ha ricevuto l'approvazione dei revisori richiesti, puoi unire il ramo di origine al ramo di destinazione nella console. CodeCatalyst L'unione di una pull request avvierà anche l'esecuzione delle modifiche in tutti i flussi di lavoro associati al ramo di destinazione. In questo tutorial, unirai il ramo di test a main, che avvierà l'esecuzione del flusso di lavoro. `onPushToMainDeployPipeline`

Per unire una pull request (console)

1. Nelle richieste Pull, scegli la pull request che hai creato nel passaggio precedente. Nella pull request, scegli Merge.
2. Scegli tra le strategie di unione disponibili per la pull request. Facoltativamente, seleziona o deselecta l'opzione per eliminare il ramo di origine dopo aver unito la pull request, quindi scegli Merge. Una volta completata l'unione, lo stato della pull request cambia in Merged e non viene più visualizzato nella visualizzazione predefinita delle pull request. La visualizzazione predefinita mostra le richieste pull con lo stato Aperto. È ancora possibile visualizzare una pull request unita, ma non è possibile approvarla o modificarne lo stato.

 Note

Se il pulsante Unisci non è attivo o viene visualizzata l'etichetta Non unibile, significa che un revisore richiesto non ha ancora approvato la pull request oppure la pull request

non può essere unita nella console. CodeCatalyst Un revisore che non ha approvato una pull request è indicato da un'icona a forma di orologio in Panoramica nell'area dei dettagli della pull request. Se tutti i revisori obbligatori hanno approvato la pull request ma il pulsante Merge non è ancora attivo, è possibile che si sia verificato un conflitto di unione o che sia stata superata la quota di archiviazione per lo spazio. È possibile risolvere i conflitti di unione per il ramo di destinazione in un ambiente di sviluppo, inviare le modifiche e quindi unire la pull request, oppure è possibile risolvere i conflitti e unirli localmente, quindi inviare il commit che contiene l'unione a CodeCatalyst Per ulteriori informazioni, consulta [Unire una richiesta pull \(Git\)](#) la documentazione di Git.

Visualizzazione del codice distribuito

Ora è il momento di visualizzare il codice originariamente distribuito che si trovava nel ramo predefinito e le modifiche unite una volta che sono state create, testate e distribuite automaticamente. A tale scopo, puoi tornare alla pagina di panoramica del repository e scegliere il numero accanto all'icona dei flussi di lavoro correlati oppure, nel pannello di navigazione, scegliere CI/CD e quindi scegliere Flussi di lavoro.

Per visualizzare il codice distribuito

1. In Flussi di lavoro, `inonPushToMainDeployPipeline`, espandi Esecuzioni recenti.

Note

Questo è il nome predefinito del flusso di lavoro per i progetti creati con il blueprint di applicazione a pagina singola.

2. L'esecuzione più recente è quella avviata dal conferimento della pull request unita al **main** branch e probabilmente mostrerà lo stato In corso. Scegli una corsa completata con successo dall'elenco per visualizzare i dettagli di quella corsa.
3. Scegli Variabili. Copia il valore per AppUrl. Questo è l'URL per l'applicazione Web a pagina singola distribuita. Apri una nuova scheda del browser e incolla il valore per visualizzare il codice creato e distribuito. Lascia aperta la scheda.
4. Torna all'elenco dei flussi di lavoro eseguiti e attendi il completamento dell'esecuzione più recente. Quando lo fa, torna alla scheda che hai aperto per visualizzare l'applicazione web e aggiornare il browser. Dovresti vedere le modifiche che hai apportato nella tua pull request unita.

Pulizia delle risorse

Dopo aver esplorato l'utilizzo di un repository di sorgenti e di una pull request, potresti voler rimuovere tutte le risorse che non ti servono. Non puoi eliminare le richieste pull, ma puoi chiuderle. Puoi eliminare tutti i rami che hai creato.

Se non hai più bisogno del repository di origine o del progetto, puoi anche eliminare quelle risorse. Per ulteriori informazioni, consultare [Eliminazione di un archivio di origine](#) e [Eliminazione di un progetto](#).

Memorizzazione del codice sorgente nei repository per un progetto in CodeCatalyst

Un repository di sorgenti consente di archiviare in modo sicuro codice e file per il progetto. Memorizza anche la cronologia delle fonti, dal primo commit fino alle ultime modifiche. Se si sceglie un blueprint che include un repository di origine, tale repository contiene anche i file di configurazione e altre informazioni per i flussi di lavoro e le notifiche per il progetto. Queste informazioni di configurazione sono archiviate in una cartella denominata `.codecatalyst`.

È possibile creare un repository di sorgenti CodeCatalyst sia creando un progetto con un blueprint che crea un repository di sorgenti come parte della creazione di un progetto, sia creando un repository di sorgenti in un progetto esistente. Gli utenti del progetto vedranno e potranno utilizzare automaticamente i repository creati per un progetto. Puoi anche scegliere di collegare un repository Git ospitato su GitHub, Bitbucket o GitLab al tuo progetto. In questo modo, gli utenti del progetto possono visualizzare e accedere a quel repository collegato nell'elenco degli archivi del progetto.

Note

Prima di poter collegare il repository, è necessario installare l'estensione per il servizio che lo ospita. Non è possibile collegare un repository archiviato. Sebbene sia possibile collegare un repository vuoto, non è possibile utilizzarlo CodeCatalyst finché non lo si è inizializzato con un commit iniziale che crea un ramo predefinito. Per ulteriori informazioni, consulta [Installazione di un'estensione in uno spazio](#).

Per impostazione predefinita, un archivio di sorgenti viene condiviso con altri membri del tuo CodeCatalyst progetto Amazon. Puoi creare repository di sorgenti aggiuntivi per un progetto o

collegare repository al progetto. Tutti i membri di un progetto possono visualizzare, aggiungere, modificare ed eliminare file e cartelle negli archivi di origine del progetto.

Per lavorare rapidamente sul codice in un repository di sorgenti, potete creare un ambiente di sviluppo che cloni un repository specifico e si ramifichi al suo interno, dove potete lavorare sul codice nell'ambiente di sviluppo integrato (IDE) che avete scelto per l'ambiente di sviluppo. Puoi clonare un repository di origine sul tuo computer locale e inserire e inserire le modifiche tra il repository locale e l'archivio remoto. CodeCatalyst Puoi anche lavorare con i repository di origine configurando l'accesso ad essi nel tuo IDE preferito, purché tale IDE supporti la gestione delle credenziali.

I nomi dei repository devono essere univoci all'interno di un progetto. CodeCatalyst

Argomenti

- [Creazione di un repository di sorgenti](#)
- [Clonazione di un repository Git esistente in un repository sorgente](#)
- [Collegamento di un repository di sorgenti](#)
- [Visualizzazione di un repository di origine](#)
- [Modifica delle impostazioni per un archivio di sorgenti](#)
- [Clonazione di un repository di sorgenti](#)
- [Eliminazione di un archivio di origine](#)

Creazione di un repository di sorgenti

Quando crei un progetto utilizzando un blueprint in Amazon CodeCatalyst, CodeCatalyst crea un repository di sorgenti per te. Tale archivio sorgente contiene codice di esempio oltre alle informazioni di configurazione per i flussi di lavoro e altre risorse create per te. Questo è il metodo consigliato per iniziare a utilizzare i repository in. CodeCatalyst Puoi scegliere di creare repository per un progetto. Questi repository conterranno un file README.md che puoi modificare o eliminare in qualsiasi momento. A seconda delle scelte effettuate al momento della creazione di un repository di origine, i repository potrebbero contenere anche un file. `.gitignore`

Se vuoi clonare un repository Git esistente in un repository CodeCatalyst sorgente, valuta invece la possibilità di creare un repository vuoto. Questo repository non sarà disponibile per l'uso CodeCatalyst finché non vi aggiungerai del contenuto, cosa che puoi fare con pochi semplici comandi Git. In alternativa, puoi aggiungere contenuti all'archivio vuoto direttamente dalla CodeCatalyst console. In alternativa, puoi collegare un repository di origine in un provider di repository Git supportato. Per ulteriori informazioni, consulta [Collegamento di un repository di sorgenti](#).

Per creare un repository di sorgenti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Crea repository.
5. In Nome archivio, fornisci un nome per il repository. In questa guida, lo usiamo *codecatalyst-source-repository*, ma puoi scegliere un nome diverso. I nomi dei repository devono essere univoci all'interno di un progetto. Per ulteriori informazioni sui requisiti per i nomi dei repository, vedere [Quote per i repository di sorgenti in CodeCatalyst](#).
6. (Facoltativo) In Descrizione, aggiungi una descrizione del repository che aiuterà gli altri utenti del progetto a capire a cosa serve il repository.
7. Scegli Crea repository (impostazione predefinita). Questa opzione crea un repository che include un ramo predefinito e un file README.md. A differenza di un repository vuoto, è possibile utilizzare questo repository non appena viene creato.
8. Nel ramo predefinito, lascia il nome come principale a meno che tu non abbia un motivo per scegliere un nome diverso. Gli esempi di questa guida utilizzano tutti il nome main per il ramo predefinito.
9. (Facoltativo) Aggiungi un .gitignore file per il tipo di codice che intendi inviare.
10. Scegli Create (Crea).

Note

CodeCatalyst aggiunge un README .md file al tuo repository quando lo crei. CodeCatalyst crea anche un commit iniziale per il repository in un ramo predefinito denominato main. È possibile modificare o eliminare il file README.md, ma non è possibile eliminare il ramo predefinito.

Per creare un archivio di sorgenti vuoto

1. Nella CodeCatalyst console, accedi al progetto in cui desideri creare un repository vuoto.
2. Nella pagina di riepilogo del progetto, in Archivi di origine, scegli Aggiungi archivio, quindi scegli Crea archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli Aggiungi repository, quindi scegli Crea repository.

3. In Nome archivio, fornisci un nome per il repository. In questa guida, lo usiamo *codecatalyst-source-repository*, ma puoi scegliere un nome diverso. I nomi dei repository devono essere univoci all'interno di un progetto. Per ulteriori informazioni sui requisiti per i nomi dei repository, vedere [Quote per i repository di sorgenti in CodeCatalyst](#)
4. (Facoltativo) In Descrizione, aggiungi una descrizione del repository che aiuterà gli altri utenti del progetto a capire a cosa serve il repository.
5. Scegli Crea un repository vuoto, quindi scegli Crea.

Clonazione di un repository Git esistente in un repository sorgente

Puoi clonare un repository Git esistente in un repository sorgente vuoto in Amazon. CodeCatalyst. Questo è un modo rapido per iniziare a usare codice precedentemente ospitato in un altro provider di repository Git. CodeCatalyst. È possibile clonare il contenuto del repository creando un clone mirror e quindi inviando il mirror a. CodeCatalyst. In alternativa, se si dispone di un repository locale del repository a cui si desidera aggiungere il contenuto CodeCatalyst, è possibile aggiungere il repository di CodeCatalyst origine come altro repository remoto al repository locale e quindi eseguire il push al repository di origine vuoto. Entrambi gli approcci sono ugualmente validi. L'uso di un clone mirror non solo mappa i rami, ma mappa tutti i riferimenti. È un modo semplice e pulito per creare una copia funzionante del repository in. CodeCatalyst. L'aggiunta di un repository remoto a un repository locale che punta a un repository CodeCatalyst sorgente vuoto aggiungerà il contenuto del repository CodeCatalyst, ma consentirà anche di effettuare push dal repository locale sia al repository di CodeCatalyst origine che al repository remoto Git originale. Questo può essere utile se desideri mantenere il codice in diversi repository remoti, ma può portare a conflitti se altri sviluppatori inseriscono il codice solo su uno dei telecomandi.

Le procedure seguenti utilizzano i comandi Git di base per eseguire questa operazione. Esistono molti modi per eseguire attività in Git, inclusa la clonazione. Per ulteriori informazioni, consulta la [documentazione di Git](#).

Important

È necessario creare un repository vuoto CodeCatalyst prima di potervi clonare il contenuto. È inoltre necessario disporre di un token di accesso personale. Per ulteriori informazioni, consultare [Per creare un archivio di sorgenti vuoto](#) e [Crea un token di accesso personale](#).

Da usare **git clone --mirror** per clonare un repository Git esistente in CodeCatalyst

1. Nella CodeCatalyst console, vai al progetto in cui hai creato un repository vuoto.
2. Nella pagina di riepilogo del progetto, scegli il repository vuoto dall'elenco, quindi scegli Visualizza archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli il nome del repository vuoto dall'elenco dei repository di origine per il progetto.
3. Copia l'URL del clone HTTPS del repository vuoto. Ti servirà per eseguire il push del clone mirror. Ad esempio, se hai chiamato il repository di origine MyExampleRepo nel MyExampleProject progetto nello ExampleCorp spazio e il tuo nome utente è LiJuan, l'URL del clone potrebbe essere simile al seguente:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

4. In una riga di comando o in una finestra di terminale, usa il **git clone --mirror** comando per creare un clone mirror del repository Git su cui vuoi clonare. CodeCatalyst Ad esempio, se desideri creare un clone mirror del repository codecatalyst-blueprints in, devi inserire il seguente comando: GitHub

```
git clone --mirror https://github.com/aws/codecatalyst-blueprints.git
```

5. Passare alla directory in cui è stato creato il clone.

```
cd codecatalyst-blueprints.git
```

6. Eseguite il git push comando, specificando l'URL e il nome del repository di origine di destinazione e l'opzione. CodeCatalyst --all (Questo è l'URL che hai copiato nel passaggio 3.) Esempio:

```
git push https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo --all
```

Per aggiungere un telecomando e inserire un repository locale CodeCatalyst

1. Nella CodeCatalyst console, vai al progetto in cui hai creato un repository vuoto.
2. Nella pagina di riepilogo del progetto, scegli il repository vuoto dall'elenco, quindi scegli Visualizza archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli il nome del repository vuoto dall'elenco dei repository di origine per il progetto.

3. Copia l'URL del clone HTTPS del repository vuoto. Ti servirà per eseguire il push del clone mirror. Ad esempio, se hai chiamato il repository di origine **MyExampleRepo** nel **MyExampleProject** progetto nello **ExampleCorp** spazio e il tuo nome utente è **LiJuan**, l'URL del clone potrebbe essere simile al seguente:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

4. Nella riga di comando o nella finestra del terminale, modifica le directory nel repository locale a cui desideri inviare il push. CodeCatalyst
5. Esegui il comando `git remote -v` per vedere i telecomandi esistenti per il repository locale. Ad esempio, se stai clonando un repository locale di un AWS CodeCommit repository denominato **MyDemoRepo** nella regione degli Stati Uniti orientali (Ohio), l'output del comando potrebbe essere simile al seguente:

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)  
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (push)
```

Copia l'URL remoto se desideri continuare a utilizzare il repository.

6. Usa il `git remote remove` comando per rimuovere il CodeCommit repository URLs for fetch e push for origin:

```
git remote remove origin
```

7. Usa il comando `git remote add` per aggiungere l'URL del repository di CodeCatalyst origine come telecomando fetch and push per il tuo repository locale. Esempio:

```
git remote add origin https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Questo sostituisce l'URL push del CodeCommit repository con l'URL del repository di CodeCatalyst origine, ma non modifica l'URL di recupero. Quindi, se esegui nuovamente il comando `git remote -v`, vedrai che ora stai estraendo (recuperando) il codice dal repository CodeCommit remoto, ma sei configurato per inviare le modifiche dal repository locale all'archivio di origine: CodeCatalyst

```
origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo (fetch)
```

```
origin https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/MyExampleProject/MyExampleRepo (push)
```

Puoi facoltativamente aggiungere nuovamente l'URL CodeCommit remoto se desideri inviarlo a entrambi i repository con il comando: `git remote set-url`

```
git remote set-url --add --push origin https://git-codecommit.us-east-2.amazonaws.com/v1/repos/MyDemoRepo
```

8. Esegui il `git push` comando per inviare il repository locale a tutti i telecomandi push configurati. In alternativa, esegui il `git push -u -origin` comando, specificando l'opzione per inviare il repository locale a entrambi i repository. Esempio:

```
git push -u -origin --all
```

Tip

A seconda della versione di Git in uso, `--all` potrebbe non funzionare per inviare tutti i rami del repository locale al repository vuoto. Potrebbe essere necessario effettuare il check-out e inviare ogni ramo separatamente.

Collegamento di un repository di sorgenti

Quando colleghi un repository di origine a un progetto, puoi includere repository con un' CodeCatalyst estensione per il servizio che ospita il repository, se tale estensione è installata per il tuo spazio. Solo gli utenti con il ruolo di amministratore dello spazio possono installare le estensioni. Una volta installata l'estensione, è possibile collegarsi ai repository configurati per l'accesso da tale estensione. Per ulteriori informazioni, consulta [Installazione di un'estensione in uno spazio](#) o segui [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Important

Dopo aver installato un'estensione del repository, il codice di tutti gli archivi a cui ti colleghi CodeCatalyst verrà indicizzato e archiviato. CodeCatalyst Ciò renderà il codice ricercabile in CodeCatalyst Per comprendere meglio la protezione dei dati per il tuo codice quando utilizzi

repository collegati in CodeCatalyst, consulta la sezione [Protezione dei dati](#) nella Amazon CodeCatalyst User Guide.

Puoi collegare un repository a un solo progetto in uno spazio. Non è possibile collegare un repository archiviato. Sebbene sia possibile collegare un repository vuoto, non è possibile utilizzarlo CodeCatalyst finché non lo si è inizializzato con un commit iniziale che crea un ramo predefinito.

Inoltre:

- Un GitHub repository, un repository Bitbucket o un repository di GitLab progetto possono essere collegati a un solo progetto in uno spazio. CodeCatalyst
- Non puoi utilizzare repository vuoti o archiviati, repository Bitbucket o GitHub repository di progetti con progetti. GitLab CodeCatalyst
- Non puoi collegare un GitHub repository, un repository Bitbucket o un repository di progetti con lo stesso nome di un repository in un GitLab progetto. CodeCatalyst
- L'estensione GitHub repositories non è compatibile con gli archivi di Enterprise Server. GitHub
- L'estensione Bitbucket repositories non è compatibile con i repository Bitbucket Data Center.
- L'estensione GitLab repositories non è compatibile con gli archivi di progetto autogestiti. GitLab
- Non puoi utilizzare le funzioni Scrivi descrizione per me o Riepiloga commenti con gli archivi collegati. Queste funzionalità sono disponibili solo nelle pull request in. CodeCatalyst

Sebbene sia possibile collegare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto come collaboratore, è possibile scollegare un repository di terze parti solo come amministratore dello spazio o amministratore del progetto. Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Come procedura consigliata, assicurati sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Per collegare un archivio di origine

1. Vai al progetto a cui desideri collegare un repository.

 Note

Prima di poter collegare un repository, un utente con il ruolo di amministratore dello spazio deve prima installare l'estensione per il provider che ospita il repository. Per ulteriori informazioni, consulta [Installazione di un'estensione in uno spazio](#).

2. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Scegli Aggiungi repository, quindi scegli Collega repository.
4. Dal menu a discesa Repository provider, scegli uno dei seguenti provider di repository di terze parti: o Bitbucket. GitHub
5. Esegui una delle seguenti operazioni a seconda del provider di repository di terze parti che hai scelto di collegare:
 - GitHub repository: collega un GitHub repository.
 1. Dal menu a discesa dell'GitHub account, scegli l' GitHub account che contiene il repository che desideri collegare.
 2. Dal menu a discesa del GitHub repository, scegli l' GitHub account a cui vuoi collegare il tuo progetto. CodeCatalyst
 3. (Facoltativo) Se non vedi un GitHub repository nell'elenco dei repository, è possibile che non sia stato configurato per l'accesso al repository nell'applicazione Amazon CodeCatalyst in. GitHub Puoi configurare quali GitHub repository possono essere utilizzati CodeCatalyst nell'account connesso.
 - a. Accedi al tuo [GitHub](#) account, scegli Impostazioni, quindi scegli Applicazioni.
 - b. Nella scheda GitHub App installate, scegli Configura per l' CodeCatalyst applicazione Amazon.
 - c. Effettua una delle seguenti operazioni per configurare l'accesso ai GitHub repository a cui desideri collegarti: CodeCatalyst

- Per fornire l'accesso a tutti i repository attuali e futuri, scegli Tutti i repository.
- Per consentire l'accesso a repository specifici, seleziona solo repository, scegli il menu a discesa Seleziona repository, quindi scegli un repository a cui desideri consentire il collegamento. CodeCatalyst
- Repository Bitbucket: collega un repository Bitbucket.
 1. Dal menu a discesa Bitbucket Workspace, scegli l'area di lavoro Bitbucket che contiene il repository che desideri collegare.
 2. Dal menu a discesa del repository Bitbucket, scegli il repository Bitbucket a cui desideri collegare il tuo progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto in Amazon. CodeCatalyst

6. Scegliere Link (Collegamento).

Se non desideri più utilizzare un GitHub repository, un repository Bitbucket o un repository di progetto in CodeCatalyst, puoi scollegarlo da un GitLab progetto. CodeCatalyst Quando un repository non è collegato, gli eventi in quel repository non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Visualizzazione di un repository di origine

Puoi visualizzare i repository di origine associati a un progetto in Amazon CodeCatalyst. Per i repository di origine in CodeCatalyst, la pagina di panoramica di un repository fornisce una rapida panoramica delle informazioni e delle attività in quel repository, tra cui:

- La descrizione del repository, se disponibile
- Il numero di filiali nel repository
- Il numero di richieste pull aperte per il repository
- Il numero di flussi di lavoro correlati per il repository
- I file e le cartelle nel ramo predefinito o nel ramo scelto

- Il titolo, l'autore e la data dell'ultimo commit nel ramo visualizzato
- Il contenuto del file README.md renderizzato in Markdown, se è incluso un file README.md

Questa pagina fornisce anche collegamenti ai commit, ai branch e alle richieste pull per il repository, oltre a un modo rapido per aprire, visualizzare e modificare singoli file.

Note

Non è possibile visualizzare queste informazioni sui repository collegati nella console. CodeCatalyst Per visualizzare le informazioni sui repository collegati, scegli il link nell'elenco dei repository per aprirlo nel servizio che lo ospita.

Per accedere ai repository di origine di un progetto

1. Passate al progetto ed effettuate una delle seguenti operazioni:
 - Nella pagina di riepilogo del progetto, scegli il repository desiderato dall'elenco, quindi scegli Visualizza archivio.
 - Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Nei repository di origine, scegli il nome del repository dall'elenco. Puoi filtrare l'elenco dei repository digitando parte del nome del repository nella barra dei filtri.
2. Nella home page del repository, visualizza il contenuto del repository e le informazioni sulle risorse associate, come il numero di richieste pull e i flussi di lavoro. Per impostazione predefinita, vengono visualizzati i contenuti del ramo predefinito. È possibile modificare la visualizzazione scegliendo un ramo diverso dall'elenco a discesa.

Tip

Puoi anche accedere rapidamente agli archivi del tuo progetto selezionando Visualizza il codice del progetto dalla pagina di riepilogo del progetto.

Modifica delle impostazioni per un archivio di sorgenti

Puoi gestire le impostazioni del tuo repository, inclusa la modifica della descrizione di un repository, la scelta del ramo predefinito, la creazione e la gestione delle regole del branch e la creazione e

la gestione delle regole di approvazione per le richieste pull in. CodeCatalyst Questo può aiutare i membri del progetto a capire a cosa serve il repository e a far rispettare le migliori pratiche e i processi utilizzati dal team.

Note

Non è possibile modificare il nome di un repository di origine.

Non è possibile modificare il nome, la descrizione o altre informazioni per un repository collegato in. CodeCatalyst Per modificare le informazioni su un repository collegato, è necessario modificarle nel provider che ospita il repository collegato. Per ulteriori informazioni, consulta la documentazione del servizio che ospita l'archivio collegato.

Per modificare le impostazioni di un repository

1. Nella CodeCatalyst console, accedi al progetto che contiene l'archivio di origine di cui desideri modificare le impostazioni.
2. Nella pagina di riepilogo del progetto, scegli il repository che desideri dall'elenco, quindi scegli Visualizza archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli il nome del repository dall'elenco dei repository di origine per il progetto.
3. Nella pagina di panoramica del repository, scegli Altro, quindi scegli Gestisci impostazioni.
4. Effettuare una o più delle seguenti operazioni:
 - Modifica la descrizione del repository, quindi scegli Salva.
 - Per modificare il ramo predefinito per il repository, in Ramo predefinito, scegli Modifica. Per ulteriori informazioni, consulta [Gestione del ramo predefinito per un repository](#).
 - Per aggiungere, rimuovere o modificare una regola in base ai ruoli del progetto autorizzati a eseguire determinate azioni in un ramo, in Regole del ramo, scegli Modifica. Per ulteriori informazioni, consulta [Gestisci le azioni consentite per una filiale con le regole della filiale](#).
 - Per aggiungere, rimuovere o modificare una regola di approvazione per l'unione delle richieste pull a un ramo, in Regole di approvazione, scegli Modifica. Per ulteriori informazioni, consulta [Gestione dei requisiti per l'unione di una pull request con le regole di approvazione](#).

Clonazione di un repository di sorgenti

Per lavorare in modo efficace con più file, branch e commit nei repository di origine, clona il repository di origine sul tuo computer locale e usa un client Git o un ambiente di sviluppo integrato (IDE) per apportare modifiche. Esegui il commit e invia le modifiche al repository di origine per poter utilizzare CodeCatalyst funzionalità come problemi e richieste pull. Puoi anche scegliere di creare un ambiente di sviluppo per lavorare sul codice. La creazione di un ambiente di sviluppo clona automaticamente il repository e il ramo specificati nell'ambiente di sviluppo.

Note

Non è possibile clonare i repository collegati nella CodeCatalyst console o creare ambienti di sviluppo per essi. Per clonare localmente un repository collegato, scegli il link nell'elenco dei repository per aprirlo nel servizio che lo ospita, quindi clonalo. Per ulteriori informazioni, consulta la documentazione del servizio che ospita l'archivio collegato.

Per creare un ambiente di sviluppo da un repository di origine

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Scegliete il repository di origine in cui desiderate lavorare sul codice.
4. Scegli Create Dev Environment.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Esegui una delle seguenti operazioni:
 - Scegli Lavora in un ramo esistente, quindi scegli un ramo dal menu a discesa Ramo esistente.
 - Scegli Lavora in un nuovo ramo, inserisci il nome di un ramo nel campo Nome ramo e scegli un ramo dal quale creare il nuovo ramo dal menu a discesa Crea ramo da.
7. Facoltativamente, aggiungi un nome per l'ambiente di sviluppo o modifica la sua configurazione.
8. Scegli Create (Crea).

Per clonare un repository di origine

1. Vai al tuo progetto.

2. Nella pagina di riepilogo del progetto, scegli il repository che desideri dall'elenco, quindi scegli Visualizza archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. È possibile filtrare l'elenco dei repository digitando parte del nome del repository nella barra dei filtri.
- 3.
4. Scegli Clone repository. Copia l'URL del clone per il repository.

Note

Se non disponi di un token di accesso personale (PAT), scegli Crea token. Copia il token e salvalo in una posizione sicura. Utilizzerai questo PAT quando ti verrà richiesta una password dal tuo client Git o dall'ambiente di sviluppo integrato (IDE).

5. Esegui una delle seguenti operazioni:
 - Per clonare un repository sul tuo computer locale, apri un terminale o una riga di comando ed esegui il git clone comando con l'URL clone dopo il comando. Esempio:

```
git clone https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Quando viene richiesta una password, incolla il PAT salvato in precedenza.

Note

Se il tuo sistema operativo fornisce la gestione delle credenziali o hai installato un sistema di gestione delle credenziali, devi fornire il PAT una sola volta. In caso contrario, potrebbe essere necessario fornire il PAT per ogni operazione Git. Come best practice, assicurati che il tuo sistema di gestione delle credenziali archivi in modo sicuro il tuo PAT. Non includete il PAT come parte della stringa URL del clone.

- Per clonare un repository utilizzando un IDE, segui la documentazione del tuo IDE. Scegli l'opzione per clonare un repository Git e fornisci l'URL. Quando viene richiesta una password, fornisci il PAT.

Eliminazione di un archivio di origine

Se un repository di sorgenti per un CodeCatalyst progetto Amazon non è più necessario, puoi eliminarlo. L'eliminazione di un repository di origine elimina anche tutte le informazioni sul progetto memorizzate nel repository. Se alcuni flussi di lavoro dipendono dall'archivio di origine, tali flussi di lavoro verranno eliminati dall'elenco dei flussi di lavoro del progetto dopo l'eliminazione del repository. I problemi che fanno riferimento all'archivio di origine non verranno eliminati o modificati, ma tutti i collegamenti all'archivio di origine aggiunti ai problemi falliranno una volta eliminato il repository.

Important

L'eliminazione di un repository di origine non può essere annullata. Dopo aver eliminato un repository di origine, non è più possibile clonarlo, estrarre dati da esso o inviargli dati. L'eliminazione di un repository di origine non elimina alcuna copia locale di tale repository (repository locali). Per eliminare un repository locale, utilizzate gli strumenti di gestione delle directory e dei file del computer locale.

Note

Non è possibile eliminare un repository collegato nella CodeCatalyst console. Per eliminare un repository collegato, scegliete il collegamento nell'elenco dei repository per aprirlo nel servizio che lo ospita, quindi eliminatelo. Per ulteriori informazioni, consulta la documentazione del servizio che ospita l'archivio collegato.

Per rimuovere un repository collegato da un progetto, vedere. [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#)

Per eliminare un repository di origine

1. Passa al progetto che contiene il repository di origine che desideri eliminare.
2. Nella pagina di riepilogo del progetto, scegli il repository che desideri dall'elenco, quindi scegli Visualizza archivio. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine. Scegli il nome del repository dall'elenco dei repository di origine per il progetto.
3. Nella home page del repository, scegli Altro, scegli Gestisci impostazioni, quindi scegli Elimina archivio.

4. Esamina le informazioni sul branch, sulla pull request e sul relativo flusso di lavoro per assicurarti di non eliminare un repository ancora in uso o con del lavoro incompiuto. Se vuoi continuare, digita `delete`, quindi scegli `Elimina`.

Organizzazione del codice sorgente, lavoro con le filiali di Amazon CodeCatalyst

In Git, i rami sono puntatori o riferimenti a un commit. Nello sviluppo, sono un modo utile di organizzare il lavoro. Puoi usare i rami per separare il lavoro su una versione nuova o diversa dei file senza influire sul lavoro in altri rami. Puoi usare branch per sviluppare nuove funzionalità, archiviare una versione specifica del tuo progetto e altro ancora. È possibile configurare regole per le filiali negli archivi di origine per limitare determinate azioni su un ramo a ruoli specifici in quel progetto.

I repository di origine in Amazon CodeCatalyst hanno contenuti e un ramo predefinito indipendentemente da come li crei. Gli archivi collegati potrebbero non avere un ramo o un contenuto predefinito, ma non sono utilizzabili CodeCatalyst finché non li iniziizzi e non crei un ramo predefinito. Quando crei un progetto utilizzando un blueprint, CodeCatalyst crea un repository di origine per quel progetto che include un file `README.md`, codice di esempio, definizioni del flusso di lavoro e altre risorse. Quando si crea un archivio di origine senza utilizzare un blueprint, viene aggiunto automaticamente un file `README.md` come primo commit e viene creato automaticamente un ramo predefinito. Questo ramo predefinito è denominato `main`. Questo ramo predefinito è quello utilizzato come ramo di base o predefinito nei repository locali (repos) quando gli utenti clonano il repository.

Note

Non è possibile eliminare il ramo predefinito. Il primo ramo creato per un repository di origine è il ramo predefinito per quel repository. Inoltre, la ricerca mostra solo i risultati del ramo predefinito. Non puoi cercare codice in altre filiali.

La creazione di un repository in crea CodeCatalyst anche un primo commit, che crea un ramo predefinito con un file `README.md` incluso. Il nome di quel ramo predefinito è `main`. Questo è il nome di ramo predefinito utilizzato negli esempi di questa guida.

Argomenti

- [Creare un ramo](#)

- [Gestione del ramo predefinito per un repository](#)
- [Gestisci le azioni consentite per una filiale con le regole della filiale](#)
- [Comandi Git per filiali](#)
- [Visualizzazione delle filiali e dei dettagli](#)
- [Eliminazione di un ramo](#)

Creare un ramo

Puoi usare la CodeCatalyst console per creare filiali in un CodeCatalyst repository. I rami che crei saranno visibili agli altri utenti la prossima volta che recupereranno le modifiche dal repository.

Tip

Puoi anche creare rami come parte della creazione di un ambiente di sviluppo per lavorare sul tuo codice. Per ulteriori informazioni, consulta [Creazione di un ambiente di sviluppo](#).

Puoi anche usare Git per creare rami. Per ulteriori informazioni, consulta [Comandi Git comuni per le filiali](#).

Per creare un ramo (console)

1. Nella CodeCatalyst console, accedi al progetto in cui risiede il tuo repository di origine.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Scegli il repository in cui desideri creare un ramo.
4. Nella pagina di panoramica del repository, scegli Altro, quindi scegli Crea ramo.
5. Inserisci un nome per il ramo.
6. Scegliete un ramo da cui creare il ramo, quindi scegliete Crea.

Gestione del ramo predefinito per un repository

Puoi specificare quale ramo utilizzare come ramo predefinito in un repository di origine in Amazon CodeCatalyst. Tutti i repository di origine CodeCatalyst hanno contenuti e un ramo predefinito, indipendentemente da come li crei. Se si utilizza un blueprint per creare un progetto, il ramo

predefinito nel repository di origine creato per quel progetto è denominato main. Il contenuto del ramo predefinito viene visualizzato automaticamente nella pagina di panoramica di quel repository.

Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per i repository collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo da CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente.

Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

È consigliabile assicurarsi sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Il ramo predefinito viene trattato in modo leggermente diverso rispetto a tutti gli altri rami in un repository di origine. Ha un'etichetta speciale accanto al suo nome, Default. Il ramo predefinito è quello usato come ramo base o predefinito nei repository locali (repository) quando gli utenti clonano il repository su computer locali con un client Git. È anche l'impostazione predefinita utilizzata durante la creazione di flussi di lavoro per l'archiviazione dei file YAML del flusso di lavoro e per l'archiviazione delle informazioni relative ai problemi. Quando si utilizza la ricerca in CodeCatalyst, viene eseguita la ricerca solo nel ramo predefinito di un repository. Poiché il ramo predefinito è fondamentale per tanti aspetti dei progetti, non è possibile eliminare un ramo se è specificato come ramo predefinito. Tuttavia, puoi scegliere di utilizzare un ramo diverso come ramo predefinito. In tal caso, tutte [le regole di filiale](#) applicate al precedente ramo predefinito verranno applicate automaticamente al ramo specificato come ramo predefinito.

Note

È necessario disporre del ruolo di amministratore del progetto per modificare il ramo predefinito per gli archivi di origine nei CodeCatalyst progetti. Questo non si applica ai repository collegati.

Per visualizzare e modificare il ramo predefinito per un repository

1. Vai al progetto in cui risiede il tuo repository.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

- Scegli il repository in cui desideri visualizzare le impostazioni, incluso il ramo predefinito.
3. Nella pagina di panoramica del repository, scegli Altro, quindi scegli Gestisci impostazioni.
 4. Nel ramo predefinito, il nome del ramo specificato come ramo predefinito viene visualizzato insieme a un'etichetta chiamata Default accanto al nome. Questa stessa etichetta appare accanto al nome del ramo nell'elenco dei rami in Branches.
 5. Per modificare il ramo predefinito, scegli Modifica.

 Note

È necessario avere il ruolo di amministratore del progetto nel progetto per modificare il ramo predefinito.

6. Scegli il nome del ramo che desideri impostare come ramo predefinito dall'elenco a discesa, quindi scegli Salva.

Gestisci le azioni consentite per una filiale con le regole della filiale

Quando si crea un ramo, sono consentite determinate azioni per quel ramo in base alle autorizzazioni per quel ruolo. È possibile modificare le azioni consentite per un ramo specifico configurando le regole del ramo. Le regole di filiale si basano sul ruolo che un utente ha nel progetto. Puoi scegliere di limitare alcune azioni predefinite, come l'invio di commit a una filiale, agli utenti con un ruolo particolare in un progetto. Questo può aiutarti a proteggere filiali specifiche di un progetto limitando i ruoli autorizzati a eseguire determinate azioni. Ad esempio, se configuri una regola di filiale per consentire solo agli utenti con il ruolo di amministratore di progetto di unirsi o inviare messaggi a quel ramo, gli utenti con altri ruoli nel progetto non saranno in grado di apportare modifiche al codice del ramo in questione.

È necessario considerare attentamente tutte le implicazioni della creazione di una regola per un ramo. Ad esempio, se scegli di limitare i push a una filiale agli utenti con il ruolo di amministratore di progetto, gli utenti con il ruolo di Collaboratore non saranno in grado di creare o modificare i flussi di lavoro in quel ramo, poiché il flusso di lavoro YAML è archiviato in quel ramo e tali utenti non possono eseguire il commit e inviare modifiche allo YAML. Come procedura consigliata, verifica tutte le regole di filiale dopo averle create per assicurarti che non abbiano impatti non previsti. Puoi anche utilizzare le regole di filiale insieme alle regole di approvazione per le pull request. Per ulteriori informazioni, consulta [Gestione dei requisiti per l'unione di una pull request con le regole di approvazione](#).

 Note

È necessario disporre del ruolo di amministratore di progetto per gestire le regole di filiale per gli archivi di origine nei CodeCatalyst progetti. Non è possibile creare regole di filiale per gli archivi collegati.

È possibile creare solo regole di filiale più restrittive rispetto alle autorizzazioni predefinite per il ruolo. Non è possibile creare regole di filiale più permissive di quelle consentite dal ruolo di un utente nel progetto. Ad esempio, non è possibile creare una regola di filiale che consenta agli utenti con il ruolo di revisore di accedere alla filiale.

Le regole di filiale applicate al ramo predefinito del repository di origine si comporteranno in modo leggermente diverso rispetto alle regole di filiale applicate ad altre filiali. Qualsiasi regola applicata al ramo predefinito verrà applicata automaticamente a qualsiasi ramo specificato come ramo predefinito. Il ramo precedentemente impostato come ramo predefinito manterrà le regole ad esso applicate, tranne per il fatto che non sarà più protetto contro l'eliminazione. Tale protezione viene applicata solo al ramo predefinito corrente.

Le regole di filiale hanno due stati, Standard e Personalizzato. Standard indica che le azioni consentite in una filiale sono quelle che corrispondono alle autorizzazioni per il ruolo ricoperto dall'utente CodeCatalyst per le azioni di filiale. Per ulteriori informazioni su quali ruoli dispongono di quali autorizzazioni, consulta [Concessione dell'accesso con ruoli utente](#). Personalizzato indica che una o più azioni di filiale prevedono azioni con un elenco specifico di ruoli autorizzati a eseguire quell'azione che differiscono dalle autorizzazioni predefinite concesse dal ruolo di un utente nel progetto.

 Note

Se crei una regola di filiale per limitare una o più azioni per un ramo, l'azione Elimina il ramo viene impostata automaticamente per consentire solo agli utenti con il ruolo di amministratore del progetto di eliminare quel ramo.

La tabella seguente elenca le azioni e le impostazioni predefinite per i ruoli autorizzati a eseguire queste azioni su un ramo.

Azioni e ruoli della filiale

Azione di filiale	Ruoli autorizzati a eseguire questa azione quando non vengono applicate regole di filiale
Unisci al ramo (questo include l'unione di una pull request al ramo)	Amministratore del progetto, collaboratore
Spingi verso la filiale	Amministratore del progetto, collaboratore
Eliminare il ramo	Amministratore del progetto, collaboratore
Elimina il ramo (ramo predefinito)	Non consentito

Non è possibile eliminare le regole del ramo, ma è possibile aggiornarle per consentire le azioni di tutti i ruoli che sarebbero autorizzati a eseguire questa azione su un ramo, il che rimuove effettivamente la regola.

 Note

È necessario disporre del ruolo di amministratore di progetto per configurare le regole di filiale per gli archivi di origine nei CodeCatalyst progetti. Questo non si applica ai repository collegati. Gli archivi collegati non supportano le regole delle filiali in CodeCatalyst

Per visualizzare e modificare le regole delle filiali per un repository

1. Vai al progetto in cui si trova il tuo repository.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri visualizzare le regole delle filiali.
3. Nella pagina di panoramica del repository, scegli Branches.
4. Nella colonna Regole del ramo, visualizza lo stato delle regole per ogni ramo del repository. Standard indica che le regole per l'azione delle filiali sono quelle predefinite per ogni ramo creato in un repository di origine e corrispondono alle autorizzazioni concesse a tali ruoli in un progetto. Personalizzato indica che una o più azioni di filiale dispongono di regole che limitano una o più azioni consentite per quel ramo a un diverso insieme di ruoli.

Per visualizzare le specifiche delle regole di filiale per una filiale, scegli la parola Standard o Personalizzato accanto alla filiale che desideri esaminare.

5. Per creare o modificare una regola di filiale, scegli Gestisci impostazioni. Nella pagina delle impostazioni per il repository di origine, in Regole di filiale, scegli Modifica.
6. In Branch, scegli il nome del ramo per il quale desideri configurare una regola dall'elenco a discesa. Per ciascuno dei tipi di azione consentiti, scegli i ruoli a cui desideri consentire l'esecuzione di quell'azione dall'elenco a discesa, quindi scegli Salva.

Comandi Git per filiali

Puoi usare Git per creare, gestire ed eliminare rami nel clone del repository di origine che hai sul tuo computer (il tuo repository locale) o nei tuoi ambienti di sviluppo, quindi eseguire il commit e inviare le modifiche al tuo repository di CodeCatalyst origine (il repository remoto). Esempio:

Comandi Git comuni per le filiali

Elenca tutti i rami del repository locale con un asterisco (*) visualizzato accanto al ramo corrente.	<code>git branch</code>
Recupera le informazioni su tutti i rami esistenti nel repository remoto nel repository locale.	<code>git fetch</code>
Elenca tutte le filiali nel repository locale e le filiali di tracciamento remoto nel repository locale.	<code>git branch -a</code>
Elenca solo le filiali di tracciamento remoto nel repository locale.	<code>git branch -r</code>
Crea un ramo nel repository locale utilizzando il nome del ramo specificato. Questo ramo non verrà visualizzato nell'archivio remoto finché non si esegue il commit e si invia la modifica.	<code>git branch <i>branch-name</i></code>

Crea un ramo nel repository locale utilizzando il nome del ramo specificato, quindi passa ad esso.	<code>git checkout -b <i>branch-name</i></code>
Passa a un altro ramo nel repository locale utilizzando il nome del ramo specificato.	<code>git checkout <i>other-branch-name</i></code>
Sposta un ramo dal repository locale all'archivio remoto utilizzando il nickname specificato dal repository locale per l'archivio remoto e il nome del ramo specificato. Imposta inoltre le informazioni di tracciamento a monte per la filiale nel repository locale.	<code>git push -u <i>remote-name</i> <i>branch-name</i></code>
Unisce le modifiche da un altro ramo nel repository locale al ramo corrente nel repository locale.	<code>git merge <i>from-other-branch-name</i></code>
Elimina un ramo nel repository locale a meno che non contenga lavori che non sono stati uniti.	<code>git branch -d <i>branch-name</i></code>
Elimina un ramo nell'archivio remoto utilizzando il soprannome specificato che il repository locale ha assegnato all'archivio remoto e il nome del ramo specificato. Da notare l'utilizzo dei due punti (:). In alternativa, specifica <code>--delete</code> come parte del comando.	<code>git push <i>remote-name</i> :<i>branch-name</i></code> <code>git push <i>remote-name</i> --delete <i>branch-name</i></code>

Per ulteriori informazioni, consulta la documentazione di Git.

Visualizzazione delle filiali e dei dettagli

Puoi visualizzare le informazioni sulle filiali remote in Amazon CodeCatalyst, incluse le specifiche di file, cartelle e il commit più recente per una filiale specifica, nella CodeCatalyst console Amazon. Puoi anche usare i comandi Git e il tuo sistema operativo locale per visualizzare queste informazioni per le filiali remote e locali.

Per visualizzare le filiali (console)

1. Nella CodeCatalyst console, accedi al progetto che contiene l'archivio dei sorgenti in cui desideri visualizzare i rami. Scegliete Codice, scegliete Archivi di origine, quindi scegliete il repository di origine.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri visualizzare un ramo.

3. Viene visualizzato il ramo predefinito del repository. È possibile visualizzare un elenco di file e cartelle nel ramo, informazioni sul commit più recente e il contenuto del file README.md, se presente nel ramo. Per visualizzare le informazioni relative a un ramo diverso, selezionalo dall'elenco a discesa dei rami del repository.
4. Per visualizzare tutti i rami di un repository, scegli Visualizza tutto. La pagina Branches mostra informazioni sul nome, sul commit più recente e sulle regole per ogni ramo.

Per informazioni su come usare Git e il sistema operativo per visualizzare rami e dettagli, consulta [Comandi Git comuni per le filiali](#), la documentazione Git e la documentazione del sistema operativo.

Eliminazione di un ramo

Se non hai più bisogno di un ramo, puoi eliminarlo. Ad esempio, se hai unito un ramo con una modifica alla funzionalità nel ramo predefinito e tale funzionalità è stata rilasciata, potresti voler eliminare il ramo di funzionalità originale, poiché le modifiche fanno già parte del ramo predefinito. Mantenere basso il numero di filiali può aiutare gli utenti a trovare il ramo che contiene le modifiche su cui desiderano lavorare. Quando si elimina un ramo, le copie di quel ramo rimangono nei cloni del repository sui computer locali finché gli utenti non recuperano e sincronizzano le modifiche.

Per eliminare un ramo (console)

1. Vai al progetto in cui risiede il tuo repository.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri eliminare un ramo.

3. Nella pagina di panoramica del repository, scegli il selettore a discesa accanto al nome del ramo, quindi scegli Visualizza tutto.

4. Scegli il ramo che desideri eliminare, quindi scegli Elimina ramo.

 Note

Non è possibile eliminare il ramo predefinito per un repository.

5. Viene visualizzata una finestra di dialogo di conferma. Mostra il repository, il numero di richieste pull aperte e il numero di flussi di lavoro associati al ramo.
6. Per confermare l'eliminazione del ramo, digita delete nella casella di testo, quindi scegli Elimina.

Puoi anche usare Git per eliminare i rami. Per ulteriori informazioni, consulta [Comandi Git comuni per le filiali](#).

Gestione dei file di codice sorgente in Amazon CodeCatalyst

In Amazon CodeCatalyst, un file è un'informazione autonoma e controllata dalla versione disponibile per te e per gli altri utenti del repository e del ramo di origine in cui è archiviato il file. Puoi organizzare i file del repository con una struttura di directory. CodeCatalyst tiene traccia automaticamente di ogni modifica apportata a un file. È possibile archiviare diverse versioni di un file in diversi rami del repository.

Per aggiungere o modificare più file in un repository di origine, puoi usare un client Git, un ambiente di sviluppo o un ambiente di sviluppo integrato (IDE). Per aggiungere o modificare un singolo file, puoi usare la CodeCatalyst console.

Argomenti

- [Creare o aggiungere un file](#)
- [Visualizzazione di un file](#)
- [Visualizzazione della cronologia delle modifiche apportate a un file](#)
- [Modifica di un file](#)
- [Ridenominazione o eliminazione di un file](#)

Creare o aggiungere un file

Per creare e aggiungere file a un repository sorgente, puoi utilizzare la CodeCatalyst console Amazon, un ambiente di sviluppo, un ambiente di sviluppo integrato (IDE) connesso o un client Git.

La CodeCatalyst console include un editor di codice per la creazione di file. Questo editor è un modo pratico per creare o modificare un file semplice, ad esempio un file README.md, in un ramo di un repository. Quando lavori su più di un file, valuta la possibilità di [creare](#) un ambiente di sviluppo.

Per creare un ambiente di sviluppo da un repository di origine

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Scegliete il repository di origine in cui desiderate lavorare sul codice.
4. Scegli Create Dev Environment.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Esegui una delle seguenti operazioni:
 - Scegli Lavora in un ramo esistente, quindi scegli un ramo dal menu a discesa Ramo esistente.
 - Scegli Lavora in un nuovo ramo, inserisci il nome di un ramo nel campo Nome ramo e scegli un ramo dal quale creare il nuovo ramo dal menu a discesa Crea ramo da.
7. Facoltativamente, aggiungi un nome per l'ambiente di sviluppo o modifica la sua configurazione.
8. Scegli Create (Crea).

Per creare un file nella console CodeCatalyst

1. Vai al progetto in cui desideri creare un file. Per ulteriori informazioni su come accedere a un repository, consulta [Visualizzazione di un repository di origine](#).
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegliete il repository in cui desiderate creare il file.

3. (Facoltativo) Scegliete il ramo in cui desiderate creare il file, se desiderate creare il file in un ramo diverso da quello predefinito.
4. Scegliete Crea file.
5. Inserisci il nome del file in Nome file. Aggiungi il contenuto del file nell'editor.

 Tip

Se desideri creare il file in una sottocartella o sottodirectory della radice del ramo, includi quella struttura come parte del nome del file.

Quando siete soddisfatti delle modifiche, scegliete Commit.

6. In Nome file, rivedi il nome del file e apporta le modifiche che desideri. Facoltativamente, scegli il ramo in cui desideri creare il file dall'elenco dei rami disponibili in Branch. In Commit message, inserisci facoltativamente una breve ma informativa descrizione del motivo per cui hai apportato questa modifica. Queste verranno visualizzate come informazioni di commit di base per il commit che aggiunge il file al repository di origine.
7. Scegli Commit per eseguire il commit e invia il file al repository di origine.

Puoi anche aggiungere file a un repository di origine clonandolo sul tuo computer locale e usando un client Git o un ambiente di sviluppo integrato (IDE) connesso per inviare file e modifiche.

 Note

Se vuoi aggiungere un sottomodulo Git, devi usare un client Git o un ambiente Dev ed eseguire il `git submodule add` comando. Non è possibile aggiungere o visualizzare i sottomoduli Git nella CodeCatalyst console o visualizzare le differenze nei sottomoduli Git nelle richieste pull. Per ulteriori informazioni sui sottomoduli Git, consulta la [documentazione di Git](#).

Per aggiungere un file utilizzando un client Git o un ambiente di sviluppo integrato (IDE) connesso

1. Clona il tuo repository di origine sul tuo computer locale. Per ulteriori informazioni, consulta [Clonazione di un repository di sorgenti](#).
2. Crea file nel repository locale o copia i file nel repository locale.
3. Crea e invia un commit effettuando una delle seguenti operazioni:
 - Se stai usando un client Git, dal terminale o dalla riga di comando, esegui il `git add` comando, specificando i nomi dei file che desideri aggiungere. In alternativa, per aggiungere tutti i file aggiunti o modificati, esegui il `git add` comando seguito da un punto singolo o doppio per

indicare se desideri includere tutte le modifiche a livello di directory corrente (periodo singolo) o tutte le modifiche nella directory corrente e in tutte le sottodirectory (periodo doppio). Per confermare le modifiche, esegui il `git commit -m` comando e fornisci un messaggio di commit. Per inserire le modifiche nell'archivio dei sorgenti CodeCatalyst, `git push` esegui. Per ulteriori informazioni sui comandi Git, consulta la documentazione Git e [Comandi Git per filiali](#).

- Se utilizzi un ambiente di sviluppo o un IDE, crea file e aggiungi file nell'IDE, quindi esegui il commit e invia le modifiche. Per ulteriori informazioni, consultate [Scrivi e modifica il codice con Dev Environments in CodeCatalyst](#) o consultate la documentazione IDE.

Visualizzazione di un file

Puoi visualizzare i file nel tuo repository di origine nella CodeCatalyst console Amazon. Puoi visualizzare i file nel ramo predefinito e in qualsiasi altro ramo. Il contenuto dei file può variare a seconda del ramo che scegli di visualizzare.

Per visualizzare i file nella CodeCatalyst console

1. Vai al progetto in cui desideri visualizzare i file. Per ulteriori informazioni, consulta [Visualizzazione di un repository di origine](#).
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri visualizzare i file.
3. Viene visualizzato un elenco di file e cartelle per il ramo predefinito. I file sono indicati da un'icona cartacea, mentre le cartelle sono indicate da un'icona a forma di cartella.
4. Effettua una delle seguenti operazioni:
 - Per visualizzare file e cartelle in un ramo diverso, selezionatelo dall'elenco dei rami.
 - Per espandere una cartella, selezionala dall'elenco.
5. Per visualizzare il contenuto di un file specifico, selezionatelo dall'elenco. Il contenuto del file verrà visualizzato nel ramo. Per visualizzare il contenuto del file in un ramo diverso, scegli il ramo che desideri dal selettore del ramo.

 Tip

Quando visualizzi il contenuto di un file, puoi scegliere altri file da visualizzare in Visualizza file. Per modificare un file, scegli Modifica.

Puoi visualizzare più file nella console. Puoi anche visualizzare i file che hai clonato sul tuo computer locale usando un client Git o un ambiente di sviluppo integrato (IDE). Per ulteriori informazioni, consulta la documentazione del tuo client Git o IDE.

 Note

Non è possibile visualizzare i sottomoduli Git nella CodeCatalyst console. Per ulteriori informazioni sui sottomoduli Git, consulta la [documentazione di Git](#).

Visualizzazione della cronologia delle modifiche apportate a un file

Puoi visualizzare la cronologia delle modifiche a un file nel tuo repository di origine nella CodeCatalyst console Amazon. Questo può aiutarti a comprendere le modifiche apportate al file mediante vari commit nel ramo in cui scegli di visualizzare la cronologia del file. Ad esempio, se visualizzi la cronologia delle modifiche al **readme.md** file nel **main** ramo del repository di origine, vedrai un elenco di commit che includevano le modifiche a quel file in quel ramo.

 Note

Non è possibile visualizzare la cronologia di un file in un repository collegato nella console CodeCatalyst

Per visualizzare la cronologia di un file nella console CodeCatalyst

1. Vai al progetto in cui desideri visualizzare la cronologia di un file. Per ulteriori informazioni, consulta [Visualizzazione di un repository di origine](#).
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

3. Scegli il repository in cui desideri visualizzare la cronologia di un file. Scegli il ramo in cui desideri visualizzare la cronologia del file, quindi scegli il file dall'elenco. Scegli View history (Visualizza cronologia).
4. Esamina l'elenco dei commit che includevano le modifiche a questo file nel ramo specificato. Per visualizzare i dettagli delle modifiche incluse in un particolare commit, scegli il messaggio di commit per quel commit nell'elenco. Vengono visualizzate le differenze tra quel commit e il commit principale.
5. Per rivedere la cronologia delle modifiche al file in un altro ramo, usa il selettore del ramo per modificare le visualizzazioni di quel ramo, scegli il file dall'elenco dei file, quindi scegli Visualizza cronologia.

Note

Non è possibile visualizzare la cronologia delle modifiche ai sottomoduli Git nella CodeCatalyst console. Per ulteriori informazioni sui sottomoduli Git, consulta la [documentazione di Git](#).

Modifica di un file

Puoi modificare singoli file nella CodeCatalyst console Amazon. Per modificare più file contemporaneamente, crea un ambiente di sviluppo o clona il repository e apporta le modifiche utilizzando un client Git o un ambiente di sviluppo integrato (IDE). Per ulteriori informazioni, consulta [Scrivi e modifica il codice con Dev Environments in CodeCatalyst](#) o [Clonazione di un repository di sorgenti](#).

Per modificare un file nella console CodeCatalyst

1. Vai al progetto in cui desideri modificare un file. Per ulteriori informazioni su come accedere a un repository, consulta [Visualizzazione di un repository di origine](#).
2. Scegliete il repository in cui desiderate modificare il file. Scegli Visualizza filiali, quindi scegli il ramo in cui vuoi lavorare. Scegli il file dall'elenco di file e cartelle in quel ramo.

Viene visualizzato il contenuto del file.

3. Scegli Modifica.

4. Nell'editor, modifica il contenuto del file, quindi scegli Commit. Facoltativamente, in Conferma modifiche, aggiungi ulteriori informazioni sulla modifica nel messaggio Commit. Quando sei soddisfatto delle modifiche, scegli Commit.

Ridenominazione o eliminazione di un file

Puoi rinominare o eliminare i file in un ambiente di sviluppo, localmente sul tuo computer o in un ambiente di sviluppo integrato (IDE). Dopo aver rinominato o eliminato i file, esegui il commit e inserisci le modifiche nell'archivio di origine. Non puoi rinominare o eliminare file nella CodeCatalyst console Amazon.

Revisione del codice con richieste pull in Amazon CodeCatalyst

Una pull request è il modo principale in cui tu e gli altri membri del progetto potete rivedere, commentare e unire le modifiche al codice da un ramo all'altro. Puoi utilizzare le pull request per esaminare le modifiche al codice in modo collaborativo per apportare modifiche o correzioni minori, aggiunte di funzionalità importanti o nuove versioni del software rilasciato. Se utilizzi i problemi per tenere traccia del lavoro sul tuo progetto, puoi collegare problemi specifici alle richieste pull per aiutarti a tenere traccia dei problemi risolti dalle modifiche al codice nella pull request. Quando crei, aggiorni, commenti, unisci o chiudi una pull request, viene automaticamente inviata un'email all'autore della pull request e a tutti i revisori necessari o facoltativi per la pull request.

Tip

Puoi configurare quali eventi di pull request riceverai e-mail come parte del tuo profilo. Per ulteriori informazioni, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).

Le richieste pull richiedono due rami in un repository di origine: un ramo di origine che contiene il codice che desideri rivedere e un ramo di destinazione, in cui desideri unire il codice recensito. Il ramo di origine contiene il commit AFTER, ovvero il commit con le modifiche da unire nel ramo di destinazione. Il ramo di destinazione contiene il commit BEFORE, che rappresenta lo stato del codice prima che il ramo della richiesta di pull venisse unito nel ramo di destinazione.

Note

Durante la creazione di una richiesta pull, la differenza visualizzata è la differenza tra la punta del ramo di origine e la punta del ramo di destinazione. Una volta creata la pull

request, la differenza visualizzata sarà tra la revisione della pull request scelta e il commit che costituiva la punta del ramo di destinazione quando è stata creata la pull request. Per ulteriori informazioni sulle differenze e sulle basi di unione in Git, consulta [git-merge-base](#) la documentazione di Git.

Sebbene venga creata una pull request per un repository di sorgenti e rami specifici, puoi crearli, visualizzarli, esaminarli e chiuderli come parte del tuo progetto. Non è necessario visualizzare il repository dei sorgenti per visualizzare e utilizzare le richieste pull. Lo stato di una pull request è impostato su Open al momento della creazione. La pull request rimane aperta finché non la si unisce nella CodeCatalyst console, che modifica lo stato in Unito, o la si chiude, il che cambia lo stato in Chiuso.

Una volta che il codice è stato esaminato, puoi modificare lo stato della pull request in diversi modi:

- Unisci la pull request nella CodeCatalyst console. Il codice nel ramo sorgente della pull request verrà unito al ramo di destinazione. Lo stato della pull request cambierà in Merged. Non può essere ripristinato in Aperto.
- Unisci i rami localmente e invia le modifiche, quindi chiudi la pull request nella CodeCatalyst console.
- Usa la CodeCatalyst console per chiudere la pull request senza unirla. Questo cambierà lo stato in Chiuso e non unirà il codice dal ramo di origine al ramo di destinazione.

Prima di creare una richiesta di pull:

- Esegui il commit e invia le modifiche al codice che desideri rivedere a un ramo (il ramo di origine).
- Imposta le notifiche per il tuo progetto, in modo che gli altri utenti possano ricevere notifiche su eventuali flussi di lavoro eseguiti quando crei una pull request. (Questo passaggio è facoltativo ma consigliato).

Argomenti

- [Creazione di una richiesta di pull](#)
- [Visualizzazione delle richieste pull](#)
- [Gestione dei requisiti per l'unione di una pull request con le regole di approvazione](#)
- [Revisione di una richiesta di pull](#)

- [Aggiornamento di una richiesta di pull](#)
- [Unione di una richiesta di pull](#)
- [Chiusura di una pull request](#)

Creazione di una richiesta di pull

La creazione di richieste pull aiuta altri utenti a vedere ed esaminare le modifiche del codice prima di unirle in un altro ramo. In primo luogo, è necessario creare un ramo per le modifiche del codice che viene definito come il ramo di origine della richiesta pull. Dopo aver eseguito il commit e aver inviato le modifiche al repository, puoi creare una richiesta pull che confronta il contenuto del ramo di origine con il contenuto del ramo di destinazione.

Puoi creare una pull request nella CodeCatalyst console Amazon da una filiale specifica, dalla pagina delle pull request o dalla panoramica del progetto. La creazione di una richiesta pull da un ramo specifico fornisce automaticamente il nome del repository e il ramo di origine nella pagina di creazione della pull request. Quando crei una pull request, riceverai automaticamente e-mail su eventuali aggiornamenti alla pull request, nonché quando la pull request viene unita o chiusa.

Note

Durante la creazione di una pull request, la differenza visualizzata è la differenza tra la punta del ramo di origine e la punta del ramo di destinazione. Una volta creata la pull request, la differenza visualizzata sarà tra la revisione della pull request scelta e il commit che costituiva la punta del ramo di destinazione quando è stata creata la pull request. Per ulteriori informazioni sulle differenze e sulle basi di unione in Git, consulta [git-merge-base](#) documentazione di Git.

Puoi utilizzare la funzione Write description for me durante la creazione di richieste pull per fare in modo che Amazon Q crei automaticamente una descrizione delle modifiche contenute in una richiesta pull. Quando scegli questa opzione, Amazon Q analizza le differenze tra il ramo di origine che contiene le modifiche al codice e il ramo di destinazione in cui desideri unire queste modifiche. Quindi crea un riepilogo di quali sono tali modifiche, nonché la migliore interpretazione dell'intento e dell'effetto di tali modifiche. Questa funzionalità è disponibile solo nella regione degli Stati Uniti occidentali (Oregon) per le CodeCatalyst pull request. La funzione Scrivi descrizione per me non è disponibile per le richieste pull negli archivi collegati.

 Note Note

Realizzato da Amazon Bedrock: AWS implementa il rilevamento [automatico degli abusi](#). Poiché le funzionalità *Scrivi una descrizione per me*, *Crea un riepilogo dei commenti*, *Attività consigliate*, *Usa Amazon Q per creare o aggiungere funzionalità a un progetto* e *Assegna i problemi ad Amazon Q con Amazon Q Developer Agent per lo sviluppo software* sono supportate da Amazon Bedrock, gli utenti possono utilizzare al meglio i controlli implementati in Amazon Bedrock per rafforzare la protezione, la sicurezza e l'uso responsabile dell'intelligenza artificiale (IA).

Come creare una richiesta pull

1. Vai al tuo progetto.
2. Esegui una delle seguenti operazioni:
 - Nel riquadro di navigazione, scegli Codice, scegli Richieste pull, quindi scegli Crea richiesta pull.
 - Nella home page del repository, scegli Altro, quindi scegli Crea richiesta pull.
 - Nella pagina del progetto, scegli Crea richiesta pull.
3. Nel repository Source, assicurati che il repository di origine specificato sia quello che contiene il codice commesso. Questa opzione appare solo se non hai creato la pull request dalla pagina principale del repository.
4. Nel ramo di destinazione, scegli il ramo in cui unire il codice dopo averlo esaminato.
5. Nel ramo di origine, scegli il ramo che contiene il codice commesso.
6. In Pull request title, inserisci un titolo che aiuti gli altri utenti a capire cosa deve essere esaminato e perché.
7. (Facoltativo) Nella descrizione della richiesta Pull, fornisci informazioni come un link ai problemi o una descrizione delle modifiche.

 Tip

Puoi scegliere Write description for me per generare CodeCatalyst automaticamente una descrizione delle modifiche contenute nella pull request. Puoi apportare modifiche alla descrizione generata automaticamente dopo averla aggiunta alla pull request. Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio e non è disponibile per le richieste pull negli archivi collegati. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

8. (Facoltativo) In Problemi, scegli Problemi di collegamento, quindi scegli un problema dall'elenco o inserisci il relativo ID. Per scollegare un problema, scegli l'icona di scollegamento.
9. (Facoltativo) In Revisori obbligatori, scegli Aggiungi revisori richiesti. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori obbligatori devono approvare le modifiche prima che la pull request possa essere unita al ramo di destinazione.

 Note

Non è possibile aggiungere un revisore sia come revisore obbligatorio che come revisore opzionale. Non puoi aggiungerti come revisore.

10. (Facoltativo) In Revisori opzionali, scegli Aggiungi revisori opzionali. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori opzionali non devono approvare le modifiche come requisito prima che la pull request possa essere unita al ramo di destinazione.
11. Esamina le differenze tra le filiali. La differenza mostrata in una pull request è rappresentata dalle modifiche tra la revisione nel ramo di origine e la base di unione, che è l'head commit del ramo di destinazione al momento della creazione della pull request. Se non viene visualizzata alcuna modifica, i rami potrebbero essere identici oppure potresti aver scelto lo stesso ramo sia per l'origine che per la destinazione.
12. Quando sei sicuro che la pull request contenga il codice e le modifiche che desideri rivedere, scegli Crea.

 Note

Dopo aver creato la pull request, puoi aggiungere commenti. I commenti possono essere aggiunti alla pull request o alle singole righe dei file, nonché alla pull request

complessiva. È possibile aggiungere collegamenti a risorse, ad esempio file, utilizzando il segno @ seguito dal nome del file.

Per creare una pull request da una filiale

1. Vai al progetto in cui desideri creare una pull request.
2. Nel pannello di navigazione, scegli Archivi di origine, quindi scegli il repository che contiene il ramo in cui devi rivedere le modifiche al codice.
3. Scegli la freccia a discesa accanto al nome del ramo predefinito, quindi scegli il ramo che desideri dall'elenco. Per visualizzare tutti i rami di un repository, scegli Visualizza tutto.
4. Scegli Altro, quindi scegli Crea pull request.
5. Il repository e il ramo sorgente sono preselezionati automaticamente. Nel ramo di destinazione, scegli il ramo in cui unirai il codice una volta esaminato. In Pull request title, inserisci un titolo che aiuti gli altri utenti del progetto a capire cosa deve essere esaminato e perché. Facoltativamente, fornisci ulteriori informazioni nella descrizione della richiesta Pull, ad esempio incollando un link a problemi correlati in CodeCatalyst o aggiungendo una descrizione delle modifiche apportate.

 Note

I flussi di lavoro configurati per l'esecuzione di eventi pull request create verranno eseguiti dopo la creazione della pull request, se il ramo di destinazione per la pull request corrisponde a uno dei rami specificati nel flusso di lavoro.

6. Esamina le differenze tra i rami. Se non viene visualizzata alcuna modifica, i rami potrebbero essere identici oppure potresti aver scelto lo stesso ramo sia per l'origine che per la destinazione.
7. (Facoltativo) In Problemi, scegli Problemi di collegamento, quindi scegli un problema dall'elenco o inserisci il relativo ID. Per scollegare un problema, scegli l'icona di scollegamento.
8. (Facoltativo) In Revisori obbligatori, scegli Aggiungi revisori richiesti. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori obbligatori devono approvare le modifiche prima che la pull request possa essere unita al ramo di destinazione.

 Note

Non puoi aggiungere un revisore sia come obbligatorio che come facoltativo. Non puoi aggiungerti come revisore.

9. (Facoltativo) In Revisori opzionali, scegli Aggiungi revisori opzionali. Scegli dall'elenco dei membri del progetto per aggiungerli. I revisori opzionali non devono approvare le modifiche prima che la pull request possa essere unita al ramo di destinazione.
10. Quando ritieni che la pull request contenga le modifiche che desideri rivedere e includa i revisori richiesti, scegli Crea.

Se hai dei flussi di lavoro configurati per essere eseguiti laddove il ramo corrisponde al ramo di destinazione nella pull request, vedrai le informazioni su tali flussi di lavoro in Panoramica nell'area dei dettagli della pull request dopo la creazione della pull request. Per ulteriori informazioni, consulta [Aggiungere trigger ai flussi di lavoro](#).

Visualizzazione delle richieste pull

Puoi visualizzare le pull request per un progetto nella CodeCatalyst console Amazon. La pagina di riepilogo del progetto mostra tutte le richieste pull aperte per un progetto. Per visualizzare tutte le pull request indipendentemente dallo stato, vai alla pagina delle pull request relativa al tuo progetto. Quando visualizzi una pull request, puoi scegliere di avere un riepilogo di tutti i commenti lasciati sulle modifiche alla pull request creata per te.

 Note Note

Realizzato da Amazon Bedrock: AWS implementa il rilevamento [automatico degli abusi](#). Poiché le funzionalità *Scrivi una descrizione per me*, *Crea un riepilogo dei commenti*, *Attività consigliate*, *Usa Amazon Q per creare o aggiungere funzionalità a un progetto* e *Assegna i problemi ad Amazon Q con Amazon Q Developer Agent* per lo sviluppo software sono supportate da Amazon Bedrock, gli utenti possono utilizzare al meglio i controlli implementati in Amazon Bedrock per rafforzare la protezione, la sicurezza e l'uso responsabile dell'intelligenza artificiale (IA).

Per visualizzare le pull request aperte

1. Vai al progetto in cui desideri visualizzare le pull request.
2. Nella pagina del progetto, vengono visualizzate le richieste pull aperte, incluse informazioni su chi ha creato la pull request, quale repository contiene i rami per la pull request e la data di creazione della pull request. È possibile filtrare la visualizzazione delle pull request aperte per repository di origine.
3. Per visualizzare tutte le richieste pull, scegli Visualizza tutto. Puoi usare i selettori per scegliere tra le opzioni. Ad esempio, per visualizzare tutte le pull request, scegliete Qualsiasi status e Qualsiasi autore.

In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Richieste Pull, quindi utilizza i selettori per affinare la visualizzazione.

4. Nella pagina delle richieste Pull, puoi ordinare le richieste pull per ID, titolo, stato e altro. Per personalizzare quali informazioni e quante informazioni vengono visualizzate nella pagina delle pull request, scegli l'icona a forma di ingranaggio.
5. Per visualizzare una pull request specifica, selezionala dall'elenco.
6. Per visualizzare lo stato delle esecuzioni dei flussi di lavoro associati a questa richiesta pull, se presenti, scegli Panoramica ed esamina le informazioni nell'area dei dettagli della richiesta pull della pull request in Workflow run.

L'esecuzione del flusso di lavoro verrà eseguita se il flusso di lavoro è configurato con eventi di creazione o revisione della pull request e se i requisiti del ramo di destinazione nel flusso di lavoro corrispondono al ramo di destinazione specificato nella pull request. Per ulteriori informazioni, consulta [Aggiungere trigger ai flussi di lavoro](#).

7. Per visualizzare gli eventuali problemi collegati, scegli Panoramica e consulta le informazioni nei dettagli della richiesta Pull in Problemi. Se desideri visualizzare un problema collegato, scegli il relativo ID dall'elenco.
8. (Facoltativo) Per creare un riepilogo dei commenti lasciati sulle modifiche al codice nelle revisioni della pull request, scegli Crea riepilogo dei contenuti. Il riepilogo non includerà alcun commento lasciato sulla pull request complessiva.

Note

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio, non è disponibile per gli archivi collegati ed è disponibile solo nella

regione Stati Uniti occidentali (Oregon). Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

9. Per visualizzare le modifiche al codice nella pull request, scegli Modifiche. Puoi visualizzare rapidamente quanti file hanno modifiche nella pull request e quali file della pull request contengono commenti, in Files changed. Il numero di commenti visualizzato accanto a una cartella indica il numero di commenti sui file in quella cartella. Espandi la cartella per visualizzare il numero di commenti per ogni file contenuto nella cartella. Puoi anche visualizzare eventuali commenti lasciati su righe di codice specifiche.

Note

Non tutte le modifiche in una pull request possono essere visualizzate nella console. Ad esempio, non è possibile visualizzare i sottomoduli Git nella console, quindi non è possibile visualizzare le differenze in un sottomodulo in una pull request. Alcune differenze potrebbero essere troppo grandi per essere visualizzate. Per ulteriori informazioni, consultare [Quote per i repository di sorgenti in CodeCatalyst](#) e [Visualizzazione di un file](#).

10. Per visualizzare i report sulla qualità per questa pull request, scegli Report.

Note

È necessario configurare un flusso di lavoro per generare report in modo che possano essere visualizzati nelle richieste pull. Per ulteriori informazioni, consulta [Test con flussi di lavoro](#).

Gestione dei requisiti per l'unione di una pull request con le regole di approvazione

Quando crei una pull request, puoi scegliere di aggiungere revisori obbligatori o facoltativi a quella richiesta pull individuale. Tuttavia, puoi anche creare requisiti che tutte le pull request devono soddisfare quando si uniscono a un ramo di destinazione specifico. Questi requisiti sono denominati regole di approvazione. Le regole di approvazione sono configurate per le filiali di un repository. Quando si crea una richiesta pull il cui ramo di destinazione dispone di una regola di approvazione

configurata per tale ramo, è necessario soddisfare i requisiti di tale regola oltre alle approvazioni di tutti i revisori necessari prima di poter unire la pull request a quel ramo. La creazione di regole di approvazione può aiutarvi a mantenere gli standard di qualità per le fusioni in filiali, ad esempio il ramo predefinito.

Le regole di approvazione applicate al ramo predefinito del repository di origine si comporteranno in modo leggermente diverso rispetto alle regole di approvazione applicate ad altre filiali. Qualsiasi regola applicata al ramo predefinito verrà applicata automaticamente a qualsiasi ramo specificato come ramo predefinito. Il ramo precedentemente impostato come ramo predefinito manterrà le regole ad esso applicate.

Quando si creano regole di approvazione, è necessario considerare in che modo tali regole verranno rispettate dagli utenti del progetto sia nel presente che in futuro. Ad esempio, se nel progetto sono presenti sei utenti e crei una regola di approvazione che richiede cinque approvazioni prima di poter essere unita al ramo di destinazione, hai effettivamente creato una regola che richiede a tutti tranne la persona che ha creato le pull request di approvare quella pull request prima che possa essere unita.

Note

È necessario avere il ruolo di amministratore del progetto per creare e gestire le regole di approvazione nei progetti. CodeCatalyst Non è possibile creare regole di approvazione per gli archivi collegati.

Non è possibile eliminare le regole di approvazione, ma è possibile aggiornarle in modo che richiedano zero approvazioni, il che rimuove di fatto la regola.

Per visualizzare e modificare le regole di approvazione per le filiali di destinazione per le richieste pull

1. Vai al progetto in cui risiede il tuo repository.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri visualizzare le regole di approvazione.

3. Nella pagina di panoramica del repository, scegli Branches.
4. Nella colonna Regole di approvazione, scegli Visualizza per visualizzare lo stato di tutte le regole per ogni ramo del repository.

In Numero minimo di approvazioni, il numero corrisponde al numero di approvazioni necessarie prima che una pull request possa essere unita a quel ramo.

5. Per creare o modificare una regola di approvazione, scegli Gestisci impostazioni. Nella pagina delle impostazioni per l'archivio di origine, in Regole di approvazione, scegli Modifica.

Note

È necessario disporre del ruolo di amministratore del progetto per modificare le regole di approvazione.

6. In Branch, scegli il nome del ramo per il quale desideri configurare una regola di approvazione dall'elenco a discesa. In Numero minimo di approvazioni, inserisci un numero, quindi scegli Salva.

Revisione di una richiesta di pull

Puoi utilizzare la CodeCatalyst console Amazon per rivedere e commentare in modo collaborativo le modifiche incluse in una pull request. Puoi aggiungere commenti a singole righe di codice nella differenza tra i rami di origine e di destinazione o nella differenza tra le revisioni della pull request. Puoi scegliere di creare un riepilogo dei commenti lasciati sulle modifiche al codice nella pull request per aiutarti a comprendere rapidamente il feedback lasciato dagli altri utenti. Puoi anche scegliere di creare un ambiente di sviluppo per lavorare sul codice.

Note

Note

Realizzato da Amazon Bedrock: AWS implementa il rilevamento [automatico degli abusi](#). Poiché le funzionalità [Scrivi una descrizione per me](#), [Crea un riepilogo dei commenti](#), [Attività consigliate](#), [Usa Amazon Q per creare o aggiungere funzionalità a un progetto](#) e [Assegna i problemi ad Amazon Q con Amazon Q Developer Agent](#) per lo sviluppo software sono supportate da Amazon Bedrock, gli utenti possono utilizzare al meglio i controlli implementati in Amazon Bedrock per rafforzare la protezione, la sicurezza e l'uso responsabile dell'intelligenza artificiale (IA).

 Tip

Puoi configurare gli eventi di pull request relativi ai quali ricevere e-mail come parte del tuo profilo. Per ulteriori informazioni, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).

Le pull request mostrano quale sarà la differenza tra la revisione della pull request e il commit che era alla base del branch di destinazione al momento della creazione della pull request. Questa è chiamata merge base. Per ulteriori informazioni sulle differenze e sulle basi di unione in Git, consulta [git-merge-base](#) la documentazione di Git.

 Tip

Quando lavori sulla console, in particolare se hai una pull request aperta da un po' di tempo, valuta la possibilità di aggiornare il browser per assicurarti di avere a disposizione l'ultima revisione di una pull request prima di iniziare a esaminarla.

Per esaminare una pull request nella console CodeCatalyst

1. Vai al tuo progetto.
2. Passa alle pull request effettuando una delle seguenti operazioni:
 - Se la pull request è elencata nella pagina del progetto, sceglila dall'elenco.
 - Se la pull request non è elencata nella pagina del progetto, scegli Visualizza tutto. Usa i filtri e ordina per trovare la pull request, quindi sceglila dall'elenco.
 - Nel riquadro di navigazione, scegli Code, quindi scegli Pull request.
3. Scegli la pull request che desideri esaminare dall'elenco. Puoi filtrare l'elenco delle pull request digitando parte del relativo nome nella barra dei filtri.
4. In Panoramica, puoi rivedere il nome e il titolo della pull request. Puoi creare e visualizzare i commenti lasciati sulla pull request stessa. Puoi anche visualizzare i dettagli della pull request, comprese le informazioni sulle esecuzioni del flusso di lavoro, i problemi collegati, i revisori, l'autore della pull request e le possibili strategie di fusione.

 Note

I commenti lasciati su righe di codice specifiche vengono visualizzati in Modifiche.

5. (Facoltativo) Per aggiungere un commento che si applichi all'intera pull request, espandi Commenti sulla pull request, quindi scegli Crea commento.
6. (Facoltativo) Per visualizzare un riepilogo di tutti i commenti lasciati sulle modifiche nelle revisioni di questa pull request, scegli Crea riepilogo dei commenti.

 Note

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio ed è disponibile solo nella regione degli Stati Uniti occidentali (Oregon). Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

7. In Modifiche, puoi vedere le differenze tra il ramo di destinazione e la revisione più recente della pull request. Se esiste più di una revisione, puoi modificare le revisioni confrontate nella differenza tra di esse. Per ulteriori informazioni sulle revisioni, vedere. [Revisioni](#)

 Tip

Puoi visualizzare rapidamente quanti file hanno subito modifiche nella richiesta pull e quali file della pull request contengono commenti, in File modificati. Il numero di commenti visualizzato accanto a una cartella indica il numero di commenti sui file in quella cartella. Espandi la cartella per visualizzare il numero di commenti per ogni file contenuto nella cartella.

8. Per modificare il modo in cui vengono visualizzate le differenze, scegli tra Unified e Split.
9. Per aggiungere un commento a una riga della pull request, vai alla riga che desideri commentare. Scegli l'icona del commento che appare per quella riga, inserisci un commento, quindi scegli Salva.
10. Per visualizzare le modifiche tra le revisioni in una pull request o tra i rami di origine e di destinazione, scegli una delle opzioni disponibili in Confronto. I commenti sulle righe delle revisioni vengono conservati in tali revisioni.

11. Se hai configurato il flusso di lavoro per generare un rapporto sulla copertura del codice sui trigger delle pull request, puoi visualizzare i risultati della copertura delle linee e delle filiali nella pull request pertinente. Per nascondere i risultati della copertura del codice, scegli **Nascondi copertura del codice**. Per ulteriori informazioni, consulta [Report di copertura del codice](#).
12. Se desideri apportare modifiche al codice della pull request, puoi creare un ambiente di sviluppo dalla pull request. Scegli **Create Dev Environment**. Facoltativamente, aggiungi un nome per l'ambiente di sviluppo o modifica la sua configurazione, quindi scegli **Crea**.
13. In **Reports**, puoi visualizzare i report sulla qualità contenuti in questa pull request. Se esiste più di una revisione, è possibile modificare le revisioni confrontate in base alla differenza tra di esse. È possibile filtrare i report per nome, stato, flusso di lavoro, azione e tipo.

 Note

È necessario configurare un flusso di lavoro per generare report in modo che possano essere visualizzati nelle richieste pull. Per ulteriori informazioni, consulta [Configurazione dei report sulla qualità in un'azione](#).

14. Per visualizzare un report specifico, selezionalo dall'elenco. Per ulteriori informazioni, consulta [Test con flussi di lavoro](#).
15. Se sei elencato come revisore di questa pull request e desideri approvare le modifiche, assicurati di visualizzare la revisione più recente, quindi scegli **Approva**.

 Note

Tutti i revisori obbligatori devono approvare una pull request prima che possa essere unita.

Aggiornamento di una richiesta di pull

Puoi semplificare la revisione del codice da parte degli altri membri del progetto aggiornando la pull request. Puoi aggiornare una pull request per modificarne i revisori, i link ai problemi, il titolo della pull request o la sua descrizione. Ad esempio, potresti voler modificare i revisori richiesti per una pull request per rimuovere qualcuno che è in vacanza e aggiungere qualcun altro. Puoi anche aggiornare una pull request con ulteriori modifiche al codice inviando i commit al ramo sorgente di una pull request aperta. Ogni push al ramo sorgente di una pull request nel repository dei CodeCatalyst

sorgenti crea una revisione. I membri del progetto possono visualizzare le differenze tra le revisioni in una pull request.

Per aggiornare i revisori per una pull request

1. Vai al progetto in cui desideri aggiornare i revisori di una pull request.
2. Nella pagina del progetto, in Apri richieste pull, scegli la pull request in cui desideri aggiornare i revisori. In alternativa, nel pannello di navigazione, scegli Code, scegli Pull request, quindi scegli la pull request che desideri aggiornare.
3. (Facoltativo) In Panoramica, nell'area dei dettagli della richiesta Pull, scegli il segno più per aggiungere revisori obbligatori o facoltativi. Scegli la X accanto a un revisore per rimuoverlo come revisore facoltativo o obbligatorio.
4. (Facoltativo) In Panoramica, nell'area dei dettagli della richiesta pull, scegli Collega problemi per collegare un problema alla pull request, quindi scegli un problema dall'elenco o inserisci il relativo ID. Per scollegare un problema, scegli l'icona di scollegamento accanto al problema che desideri scollegare.

Per aggiornare file e codice nel ramo sorgente di una pull request

1. Per aggiornare più file, [crea un ambiente di sviluppo](#) o clona il repository e il suo ramo sorgente e usa un client Git o un ambiente di sviluppo integrato (IDE) per apportare modifiche ai file nel ramo di origine. Conferma e invia le modifiche al ramo di origine nel repository di CodeCatalyst origine per aggiornare automaticamente la pull request con le modifiche. Per ulteriori informazioni, consultare [Clonazione di un repository di sorgenti](#) e [Comprendere le modifiche al codice sorgente con i commit in Amazon CodeCatalyst](#).
2. Per aggiornare un singolo file in un ramo sorgente, puoi usare un client Git o un IDE come faresti per più file. Puoi anche modificarlo direttamente nella CodeCatalyst console. Per ulteriori informazioni, consulta [Modifica di un file](#).

Per aggiornare il titolo e la descrizione di una pull request

1. Vai al progetto in cui desideri aggiornare il titolo o la descrizione di una pull request.
2. La pagina del progetto mostra le pull request aperte, incluse informazioni su chi ha creato la pull request, quale repository contiene i rami per la pull request e quando è stata creata la pull

request. È possibile filtrare la visualizzazione delle pull request aperte per repository di origine. Scegli la pull request che desideri modificare dall'elenco.

3. Per visualizzare tutte le pull request, scegli Visualizza tutto. In alternativa, nel riquadro di navigazione, scegli Code, quindi scegli Pull request. Usa la casella di filtro o le funzioni di ordinamento per trovare la pull request che desideri modificare, quindi selezionala.
4. In Panoramica, scegli Modifica.
5. Cambia il titolo o la descrizione, quindi scegli Salva.

Unione di una richiesta di pull

Dopo che il codice è stato esaminato e tutti i revisori richiesti lo hanno approvato, puoi unire una pull request nella CodeCatalyst console utilizzando una strategia di unione supportata, ad esempio fast-forward. Non tutte le strategie di unione supportate nella CodeCatalyst console sono disponibili come scelte per tutte le richieste pull. CodeCatalyst valuta l'unione e consente di scegliere solo tra strategie di unione disponibili nella console e in grado di unire il ramo di origine con il ramo di destinazione. Puoi anche unire una richiesta pull con la tua scelta di strategie di fusione Git eseguendo il git merge comando sul tuo computer locale o in un ambiente di sviluppo per unire il ramo di origine nel ramo di destinazione. È quindi possibile inviare tali modifiche nel ramo di destinazione all'archivio di origine. CodeCatalyst

Note

L'unione del ramo e l'inserimento delle modifiche in Git non chiude automaticamente la pull request.

Se hai il ruolo di amministratore di progetto, puoi anche scegliere di unire una pull request che non ha ancora soddisfatto tutti i requisiti per le approvazioni e le regole di approvazione.

Unione di una pull request (console)

È possibile unire una richiesta pull nella CodeCatalyst console se non ci sono conflitti di unione tra i rami di origine e di destinazione e se tutti i revisori richiesti hanno approvato la pull request. Se ci sono conflitti o se l'unione non può essere completata, il pulsante di unione è inattivo e viene visualizzata l'etichetta Non unibile. In tal caso, è necessario ottenere l'approvazione di tutti gli approvatori necessari, risolvere i conflitti a livello locale se necessario e apportare le modifiche

prima di procedere all'unione. L'unione di una pull request invierà automaticamente un'e-mail al creatore della pull request e a tutti i revisori obbligatori o facoltativi. Non chiuderà o modificherà automaticamente lo stato di eventuali problemi collegati alla pull request.

Tip

Puoi configurare quali eventi di pull request ricevere e-mail come parte del tuo profilo. Per ulteriori informazioni, consulta [Invio di notifiche via email e su Slack da CodeCatalyst](#).

Per unire una pull request

1. Vai al progetto in cui desideri unire una richiesta pull.
2. Nella pagina del progetto, in Apri richieste pull, scegli la pull request che desideri unire. Se non vedi la pull request, scegli Visualizza tutte le pull request e poi scegli la dall'elenco. In alternativa, nel pannello di navigazione, scegli Code, scegli Pull request, quindi scegli la pull request che desideri unire. Selezionare Merge (Unisci).
3. Scegli tra le strategie di unione disponibili per la pull request. Facoltativamente, seleziona o deseleziona l'opzione per eliminare il ramo di origine dopo aver unito la pull request, quindi scegli Merge.

Note

Se il pulsante Unisci è inattivo o viene visualizzata l'etichetta Non unibile, significa che i revisori obbligatori non hanno ancora approvato la pull request oppure la pull request non può essere unita nella console. CodeCatalyst Un revisore che non ha approvato una pull request è indicato da un'icona a forma di orologio nell'area dei dettagli della pull request in Panoramica. Se tutti i revisori obbligatori hanno approvato la pull request ma il pulsante Merge è ancora inattivo, potresti avere un conflitto di unione. Scegli l'etichetta sottolineata Non unibile per visualizzare maggiori dettagli sul motivo per cui la pull request non può essere unita. Puoi risolvere i conflitti di unione per il ramo di destinazione in un ambiente di sviluppo o nella CodeCatalyst console e quindi unire la richiesta pull, oppure puoi risolvere i conflitti e unirli localmente, quindi inviare il commit che contiene l'unione al ramo di origine. CodeCatalyst Per ulteriori informazioni, consulta [Unire una richiesta pull \(Git\)](#) la documentazione di Git.

Sostituisci i requisiti di unione

Se hai il ruolo di amministratore di progetto, puoi scegliere di unire una pull request che non ha ancora soddisfatto tutti i requisiti per le approvazioni e le regole di approvazione richieste. Questa operazione viene definita come l'annullamento dei requisiti per una pull request. Potresti scegliere di farlo se un revisore richiesto non è disponibile o se hai urgente necessità di unire una pull request specifica in una filiale con regole di approvazione che non possono essere soddisfatte rapidamente.

Per unire una pull request

1. Nella pull request in cui desideri sovrascrivere i requisiti e unirli, scegli la freccia a discesa accanto al pulsante Unisci. Scegli Ignora i requisiti di approvazione.
2. In Override reason, fornisci i dettagli del motivo per cui stai unendo questa pull request senza che soddisfi le regole di approvazione e i requisiti richiesti per i revisori. Sebbene sia facoltativo, è altamente consigliato.
3. Facoltativamente, scegli una strategia di fusione o accetta quella predefinita. Puoi anche scegliere di aggiornare il messaggio di commit generato automaticamente con maggiori dettagli.
4. Seleziona o deseleziona l'opzione per eliminare il ramo di origine al momento dell'unione. Ti consigliamo di mantenere il ramo di origine quando sostituisci i requisiti per l'unione di una pull request fino a quando non avrai avuto la possibilità di rivedere la decisione con gli altri membri del team.
5. Selezionare Merge (Unisci).

Unire una richiesta pull (Git)

Git supporta molte opzioni per unire e gestire le filiali. I seguenti comandi sono alcune delle opzioni che puoi usare. Per ulteriori informazioni, consulta la documentazione disponibile sul [sito Web Git](#). Dopo aver unito e inserito le modifiche, chiudi manualmente la pull request. Per ulteriori informazioni, consulta [Chiusura di una pull request](#).

Comandi Git comuni per unire rami

Unisce le modifiche dal ramo di origine nel repository locale al ramo di destinazione nel repository locale.

```
git checkout destination-branch-name
```

```
git merge source-branch-name
```

Unisce il ramo di origine al ramo di destinazione, specificando un'unione rapida. In questo modo i rami vengono uniti e il puntatore del ramo di destinazione viene spostato sulla punta del ramo di origine.

```
git checkout destination-branch-name
```

```
git merge --ff-only source-branch-name
```

Unisce il ramo di origine al ramo di destinazione, specificando un'unione squash. Questo combina tutti i commit del ramo di origine in un unico commit di unione nel ramo di destinazione.

```
git checkout destination-branch-name
```

```
git merge --squash source-branch-name
```

Unisce il ramo di origine al ramo di destinazione, specificando un'unione a tre vie. Questo crea un commit di unione e aggiunge i singoli commit dal ramo di origine al ramo di destinazione.

```
git checkout destination-branch-name
```

```
git merge --no-ff source-branch-name
```

Elimina il ramo di origine nel repository locale. Ciò è utile per ripulire il repository locale dopo l'unione al ramo di destinazione e aver inviato le modifiche all'archivio di origine.

```
git branch -d source-branch-name
```

Elimina il ramo di origine nel repository remoto (il repository di origine in CodeCatalyst) utilizzando il nickname specificato dal repository locale per il repository remoto. Da notare l'utilizzo dei due punti (:). In alternativa, specifica `--delete` come parte del comando.

```
git push remote-name :source-branch-name
```

```
git push remote-name --delete source-branch-name
```

Chiusura di una pull request

Puoi contrassegnare una pull request come chiusa. Ciò non unisce la pull request, ma può aiutarti a determinare quali pull request richiedono un'azione e quali non sono più pertinenti. Ti consigliamo di chiudere una pull request se non hai più intenzione di unire tali modifiche o se le modifiche sono state unite da un'altra pull request.

La chiusura di una pull request invierà automaticamente un'email all'autore della pull request e a tutti i revisori obbligatori o facoltativi. Non cambierà automaticamente lo stato di eventuali problemi legati alla pull request.

Note

Non è possibile riaprire una pull request dopo che è stata chiusa.

Per chiudere una pull request

1. Vai al progetto in cui desideri chiudere una pull request.
2. Nella pagina del progetto, vengono visualizzate le pull request aperte. Scegli la pull request che vuoi chiudere.
3. Scegli Chiudi.
4. Esamina le informazioni, quindi scegli Close pull request.

Comprendere le modifiche al codice sorgente con i commit in Amazon CodeCatalyst

I commit sono snapshot dei contenuti e modifiche apportate ai contenuti di un repository. Ogni volta che un utente esegue un commit e invia una modifica a un ramo, tali informazioni vengono salvate. Le informazioni sul commit di Git includono l'autore del commit, la persona che ha eseguito la modifica, la data e l'ora e le modifiche apportate. Informazioni simili vengono incluse automaticamente quando crei o modifichi un file nella CodeCatalyst console Amazon, ma il nome dell'autore è il tuo nome CodeCatalyst utente. Puoi anche aggiungere tag Git ai commit per aiutarti a identificare commit specifici.

In Amazon CodeCatalyst, puoi:

- Visualizza un elenco di commit per una filiale.
- Visualizza i singoli commit, incluse le modifiche apportate in un commit rispetto ai commit principali.

Puoi anche visualizzare file e cartelle. Per ulteriori informazioni, consulta [Gestione dei file di codice sorgente in Amazon CodeCatalyst](#).

Argomenti

- [Visualizzazione dei commit verso un ramo](#)
- [Modifica della modalità di visualizzazione dei commit \(console\) CodeCatalyst](#)

Visualizzazione dei commit verso un ramo

Puoi visualizzare la cronologia delle modifiche apportate a una filiale esaminando i commit della filiale nella CodeCatalyst console. Questo ti aiuta a capire chi ha apportato modifiche alla filiale e quando. Puoi anche rivedere le modifiche apportate in un commit specifico.

Tip

È inoltre possibile visualizzare la cronologia dei commit che hanno apportato modifiche a un file specifico. Per ulteriori informazioni, consulta [Visualizzazione di un file](#).

Puoi anche visualizzare i commit usando il tuo client Git. Per ulteriori informazioni, consulta la documentazione di Git.

Per visualizzare i commit (console)

1. Passa al progetto che contiene l'archivio dei sorgenti in cui desideri visualizzare i commit.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri visualizzare i commit verso una filiale.

3. Viene visualizzato il ramo predefinito del repository, incluse le informazioni sul commit più recente verso il ramo. Scegli Commit. In alternativa, scegli Altro, quindi scegli Visualizza i commit.
4. Per visualizzare i commit per un ramo diverso, scegli il selettore del ramo, quindi scegli il nome del ramo.
5. Per visualizzare i dettagli su un particolare commit, sceglilo il titolo da Commit title. Vengono visualizzati i dettagli del commit, incluse le informazioni sul commit principale e le modifiche apportate al codice confrontando il commit principale con il commit specificato.

 Tip

Se un commit ha più di un commit principale, puoi scegliere per quale commit genitore visualizzare le informazioni e visualizzare le modifiche scegliendo l'icona a discesa accanto all'ID del commit principale.

Modifica della modalità di visualizzazione dei commit (console) CodeCatalyst

Puoi modificare le informazioni visualizzate nella vista Commit. Puoi scegliere di nascondere o visualizzare colonne come autore e ID di commit.

Per modificare la modalità di visualizzazione dei commit (console)

1. Passa al progetto che contiene il repository di origine in cui desideri visualizzare i commit.
2. Scegli il nome del repository dall'elenco dei repository di origine per il progetto. In alternativa, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Scegli il repository in cui desideri modificare la modalità di visualizzazione dei commit.

3. Viene visualizzato il ramo predefinito del repository, incluse le informazioni sul commit più recente nel ramo. Scegli Commit.
4. Scegliere l'icona a forma di ingranaggio.
5. In Preferenze, scegli il numero di commit da visualizzare e scegli se visualizzare le informazioni sull'autore del commit, sulla data di commit e sull'ID del commit.

 Note

Non puoi nascondere il titolo del commit nella visualizzazione delle informazioni.

6. Dopo aver apportato le modifiche, scegli Salva per salvarle o Annulla per eliminarle.

Quote per i repository di sorgenti in CodeCatalyst

La tabella seguente descrive le quote e i limiti per gli archivi di origine in Amazon. CodeCatalyst Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Risorsa	Informazioni
Nomi dei rami	Qualsiasi combinazione di caratteri consentiti compresa tra 1 e 256 caratteri di lunghezza deve essere univoca all'interno di un repository. I nomi dei rami non possono: - iniziare né terminare con una barra (/) o un punto (.) - contenere il singolo carattere @ - contenere due o più punti consecutivi (..), barre (//) o la seguente combinazione di caratteri: @{ - contenere spazi o uno qualsiasi dei seguenti caratteri: ? ^ * [\ ~ : I nomi dei rami sono riferimenti. Molte delle restrizioni relative ai nomi dei rami sono basate sullo standard di riferimento Git. Per ulteriori informazioni, consulta Git Internals e git-check-ref-format.
Commenti su una pull request	Massimo 1.000 su una pull request.
Messaggio di commit	Massimo 1024 caratteri.
Percorsi di file	Qualsiasi combinazione di caratteri consentiti, per una lunghezza compresa tra 1 e 4.096 caratteri. I percorsi dei file devono essere costituiti da un nome non ambiguo che specifica il file e dalla posizione esatta del file. Oltre a non poter contenere più di 20 directory, Inoltre, i percorsi di file non possono: - contenere stringhe vuote - rappresentare un percorso file relativo

Risorsa	Informazioni
	• includere una qualsiasi delle seguenti combinazioni di caratteri: <code>/./</code> <code>/../</code> <code>//</code> • terminare con una barra o una barra rovesciata I nomi e i percorsi di file devono essere completi. Il nome e il percorso di un file sul tuo computer locale devono essere conformi agli standard del sistema operativo in uso. Quando specifichi il percorso di un file in un repository, usa gli standard per Amazon Linux.
Dimensioni file	Massimo 6 MB per ogni singolo file quando si utilizza la CodeCatalyst console.
Dimensione del file visualizzabile nella console CodeCatalyst	Massimo 6 MB per ogni singolo file quando si utilizza la CodeCatalyst console.

Risorsa	Informazioni
Dimensioni blob Git	Massimo 2 GB.  Note Non esiste alcuna restrizione relativa al numero di file o alla dimensione totale di tutti i file in un singolo commit, purché i metadati non superino i 6 MB e un singolo blob non superi i 2 GB. Tuttavia, come best practice, prendi in considerazione la possibilità di effettuare e più commit più piccoli anziché un unico commit di grandi dimensioni.
Metadati per un commit	Massimo 6 MB per i metadati combinati per un commit (ad esempio, la combinazione di informazioni sull'autore, data, elenco di commit principale e messaggi di commit).  Note Non esiste alcuna restrizione relativa al numero di file o alla dimensione totale di tutti i file in un singolo commit, purché i dati non superino i 20 MB, un file singolo non superi i 6 MB e un singolo blob non superi i 2 GB.
Numero di CodeCatalyst problemi che possono essere collegati a una pull request	50
Numero di problemi di Jira che possono essere collegati a una pull request	50

Risorsa	Informazioni
Numero di richieste pull aperte in uno spazio	Massimo 1.000 per uno CodeCatalyst spazio Amazon.
Numero totale di pull request in uno spazio	Massimo 10.000 per uno CodeCatalyst spazio Amazon.
Numero di riferimenti in un singolo push	Massimo 4.000, include operazioni di creazione , eliminazione e aggiornamento. Non esiste tuttavia alcuna restrizione sul numero complessivo di riferimenti nel repository.
Numero di repository in uno spazio	Massimo 5.000 per uno CodeCatalyst spazio Amazon.
Descrizioni dei repository	Qualsiasi combinazione di caratteri, per una lunghezza compresa tra 0 e 1.000 caratteri. Le descrizioni dei repository sono facoltative.
Nomi dei repository	I nomi dei repository devono essere univoci all'interno di un progetto. Possono contenere qualsiasi combinazione di lettere, numeri, punti, caratteri di sottolineatura e trattini di lunghezza compresa tra 1 e 100 caratteri. I nomi non distinguono tra maiuscole e minuscole. I nomi dei repository non possono terminare con .git, non possono contenere spazi e non possono contenere nessuno dei seguenti caratteri: ! ? @ # \$ % ^ & * () + = { } [] \ / > < ~ ` ' " ; :
Dimensioni del repository	Le dimensioni del repository sono influenzate dai limiti complessivi di spazio di archiviazione dell'utente. Per ulteriori informazioni, consulta Prezzi di e Risoluzione dei problemi relativi ai repository di origine .

Risorsa	Informazioni
Revisori per una pull request	Un massimo di 100 revisori in totale (facoltativo o obbligatorio) per una pull request.
Riepiloghi scritti per le pull request	Il numero massimo di riepiloghi scritti per le pull request dipende dal livello di fatturazione del tuo spazio. Per ulteriori informazioni, consultare Prezzi .

Scrivi e modifica il codice con Dev Environments in CodeCatalyst

Gli ambienti di sviluppo sono ambienti di sviluppo basati sul cloud. In Amazon CodeCatalyst, usi Dev Environments per lavorare sul codice memorizzato nei repository di origine del tuo progetto. Quando crei un ambiente di sviluppo, hai diverse opzioni:

- Crea un ambiente di sviluppo specifico per il progetto CodeCatalyst per lavorare sul codice con un ambiente di sviluppo integrato (IDE) supportato.
- Crea un ambiente di sviluppo vuoto, clona il codice da un repository sorgente e lavora su quel codice con un IDE supportato.
- Crea un ambiente di sviluppo in un IDE e clona un repository di sorgenti nell'ambiente di sviluppo

Un devfile è un file YAML standard aperto che standardizza i tuoi ambienti di sviluppo. In altre parole, questo file codifica gli strumenti di sviluppo necessari per il tuo ambiente di sviluppo. Di conseguenza, puoi configurare rapidamente un ambiente di sviluppo, passare da un progetto all'altro e replicare la configurazione dell'ambiente di sviluppo tra i membri del team. Gli ambienti di sviluppo riducono al minimo il tempo dedicato alla creazione e alla manutenzione di un ambiente di sviluppo locale, perché utilizzano un devfile che configura tutti gli strumenti necessari per codificare, testare ed eseguire il debug per un determinato progetto.

Important

Dev Environments può eseguire script tramite devfiles che hanno accesso alle tue credenziali. CodeCatalyst ispeziona il repository prima di aprire risorse non attendibili.

Gli strumenti di progetto e le librerie di applicazioni inclusi nel tuo ambiente di sviluppo sono definiti dal devfile nel repository di origine del tuo progetto. Se non avete un devfile nel vostro repository di origine, applica CodeCatalyst automaticamente un devfile predefinito. Questo devfile predefinito include strumenti per i linguaggi e i framework di programmazione più utilizzati. Se il progetto è stato creato utilizzando un blueprint, viene creato automaticamente un devfile da. CodeCatalyst [Per ulteriori informazioni sul devfile, vedere https://devfile.io](https://devfile.io).

Dopo aver creato un ambiente di sviluppo, solo tu puoi accedervi. Nel tuo ambiente di sviluppo, puoi visualizzare e lavorare sul codice del tuo repository sorgente in un IDE supportato.

Per impostazione predefinita, un ambiente di sviluppo è configurato per avere un processore a 2 core, 4 GB di RAM e 16 GB di storage persistente. Se disponi delle autorizzazioni di amministratore di Space, puoi modificare il livello di fatturazione del tuo spazio per utilizzare diverse opzioni di configurazione di Dev Environment e gestire i limiti di calcolo e archiviazione.

Argomenti

- [Creazione di un ambiente di sviluppo](#)
- [Interruzione di un ambiente di sviluppo](#)
- [Ripresa del lavoro in un ambiente di sviluppo](#)
- [Modifica di un ambiente di sviluppo](#)
- [Eliminazione di un ambiente di sviluppo](#)
- [Connessione a un ambiente di sviluppo tramite SSH](#)
- [Configurazione di un devfile per un ambiente di sviluppo](#)
- [Associazione di una connessione VPC a un ambiente di sviluppo](#)
- [Quote per ambienti di sviluppo in CodeCatalyst](#)

Creazione di un ambiente di sviluppo

Ci sono diversi modi per creare un ambiente di sviluppo:

- Crea un ambiente di sviluppo CodeCatalyst con un repository di CodeCatalyst origine o un repository di [sorgenti collegato dalle pagine Overview, Dev Environments o Source repository](#)
- Crea un ambiente di sviluppo vuoto CodeCatalyst che non sia collegato a un repository di origine dalla pagina Dev Environments
- Crea un ambiente di sviluppo nel tuo IDE preferito e clona qualsiasi repository di origine nell'ambiente di sviluppo

Important

Gli ambienti di sviluppo non sono disponibili per gli utenti negli spazi in cui Active Directory viene utilizzato come provider di identità. Per ulteriori informazioni, consulta [Non riesco a creare un ambiente di sviluppo quando ho effettuato l'accesso CodeCatalyst utilizzando un account Single Sign-on](#).

È possibile creare un ambiente di sviluppo per ramo di un repository. Un progetto può avere più repository. Gli ambienti di sviluppo che crei possono essere gestiti solo con il tuo CodeCatalyst account, ma puoi aprire l'ambiente di sviluppo e utilizzarlo con uno qualsiasi degli ambienti supportati IDEs. Devi averlo Kit di strumenti AWS installato per utilizzare Dev Environments nel tuo IDE. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#). Per impostazione predefinita, gli ambienti di sviluppo vengono creati con un processore a 2 core, 4 GB di RAM e 16 GB di storage persistente.

Note

Se hai creato un ambiente di sviluppo associato a un repository di origine, la colonna Risorse mostra sempre il ramo specificato durante la creazione di questo ambiente di sviluppo. Questo vale anche se crei un altro ramo, passi a un altro ramo all'interno di Dev Environment o cloni un repository aggiuntivo. Se hai creato un ambiente di sviluppo vuoto, la colonna Resource sarà vuota.

Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo

È possibile utilizzare Dev Environments con i seguenti ambienti di sviluppo integrati supportati (IDEs):

- [AWS Cloud9](#)
- [JetBrains IDEs](#)
 - [IntelliJ IDEA Ultimate](#)
 - [GoLand](#)
 - [PyCharm Professionale](#)
- [Visual Studio Code](#)

Creazione di un ambiente di sviluppo in CodeCatalyst

[Per iniziare a lavorare con Dev Environment in CodeCatalyst, autenticali e accedi con il tuo AWS Builder ID o SSO.](#)

Per creare un ambiente di sviluppo da una filiale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri creare un ambiente di sviluppo.

3. Nel riquadro di navigazione, esegui una delle seguenti operazioni:
 - Scegli Panoramica, quindi vai alla sezione I miei ambienti di sviluppo.
 - Scegli Code, quindi scegli Dev Environments.
 - Scegli Code, scegli Repository di origine e scegli il repository per il quale desideri creare un ambiente di sviluppo.
4. Scegli Crea ambiente di sviluppo.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Scegli Clona un repository.
7. Esegui una delle seguenti operazioni:
 - a. Scegli il repository da clonare, scegli Lavora in un ramo esistente, quindi scegli un ramo dal menu a discesa Existing branch.

 Note

Se scegli un repository di terze parti, devi lavorare in un ramo esistente.

- b. Scegli il repository da clonare, scegli Lavora in un nuovo ramo, inserisci il nome di un ramo nel campo Nome del ramo e scegli un ramo dal quale creare il nuovo ramo dal menu a discesa Crea ramo da.

 Note

Se crei un ambiente di sviluppo dalla pagina dei repository di origine o da un repository di origine specifico, non è necessario scegliere un repository. L'ambiente di sviluppo verrà creato dal repository di origine che hai scelto nella pagina dei repository di origine.

8. (Facoltativo) In Alias: facoltativo, inserisci un alias per l'ambiente di sviluppo.
9. (Facoltativo) Scegliete il pulsante di modifica della configurazione dell'ambiente di sviluppo per modificare la configurazione di calcolo, archiviazione o timeout dell'ambiente di sviluppo.
10. (Facoltativo) In Amazon Virtual Private Cloud (Amazon VPC): facoltativo, seleziona una connessione VPC da associare al tuo ambiente di sviluppo dal menu a discesa.

Se per il tuo spazio è impostato un VPC predefinito, i tuoi ambienti di sviluppo verranno eseguiti collegati a quel VPC. Puoi sovrascrivere questo problema associando una connessione VPC diversa. Inoltre, tieni presente che gli ambienti di sviluppo connessi a VPC non sono supportati. Kit di strumenti AWS

Se la connessione VPC che desideri utilizzare non è elencata, potrebbe essere perché include una Account AWS connessione non consentita nel tuo progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

 Note

Quando crei un ambiente di sviluppo con una connessione VPC, viene creata una nuova interfaccia di rete all'interno del VPC. CodeCatalyst interagisce con questa interfaccia utilizzando il ruolo VPC associato. Inoltre, assicurati che il blocco IPv4 CIDR non sia configurato nell'intervallo di 172.16.0.0/12 indirizzi IP.

11. Scegli Create (Crea). Durante la creazione del tuo ambiente di sviluppo, la colonna di stato dell'ambiente di sviluppo mostrerà Avvio e la colonna di stato mostrerà In esecuzione una volta creato l'ambiente di sviluppo.

Per creare un ambiente di sviluppo vuoto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri creare un ambiente di sviluppo.
3. Nel riquadro di navigazione, esegui una delle seguenti operazioni:
 - Scegli Panoramica, quindi vai alla sezione I miei ambienti di sviluppo.
 - Scegli Code, quindi scegli Dev Environments.
4. Scegli Crea ambiente di sviluppo.
5. Scegli un IDE supportato dal menu a discesa. Per ulteriori informazioni, consulta [Ambienti di sviluppo integrati supportati per gli ambienti di sviluppo](#).
6. Scegli Crea un ambiente di sviluppo vuoto.
7. (Facoltativo) In Alias: facoltativo, inserisci un alias per l'ambiente di sviluppo.

8. (Facoltativo) Scegliete il pulsante di modifica della configurazione dell'ambiente di sviluppo per modificare la configurazione di calcolo, archiviazione o timeout dell'ambiente di sviluppo.
9. (Facoltativo) In Amazon Virtual Private Cloud (Amazon VPC): facoltativo, seleziona una connessione VPC da associare al tuo ambiente di sviluppo dal menu a discesa.

Se per il tuo spazio è impostato un VPC predefinito, i tuoi ambienti di sviluppo verranno eseguiti collegati a quel VPC. Puoi sovrascrivere questo problema associando una connessione VPC diversa. Inoltre, tieni presente che gli ambienti di sviluppo connessi a VPC non sono supportati.

Kit di strumenti AWS

Se la connessione VPC che desideri utilizzare non è elencata, potrebbe essere perché include una Account AWS connessione non consentita nel tuo progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

Note

Quando crei un ambiente di sviluppo con una connessione VPC, viene creata una nuova interfaccia di rete all'interno del VPC. CodeCatalyst interagisce con questa interfaccia utilizzando il ruolo VPC associato. Inoltre, assicurati che il blocco IPv4 CIDR non sia configurato nell'intervallo di 172.16.0.0/12 indirizzi IP.

10. Scegli Create (Crea). Durante la creazione del tuo ambiente di sviluppo, la colonna di stato dell'ambiente di sviluppo mostrerà Avvio e la colonna di stato mostrerà In esecuzione una volta creato l'ambiente di sviluppo.

Note

La creazione e l'apertura di un ambiente di sviluppo per la prima volta potrebbero richiedere da uno a due minuti.

Note

Dopo l'apertura dell'ambiente di sviluppo nell'IDE, potrebbe essere necessario modificare la directory nel repository di origine prima di eseguire il commit e inviare le modifiche al codice.

Creazione di un ambiente di sviluppo in un IDE

Puoi usare Dev Environments per lavorare rapidamente sul codice memorizzato nei repository di origine del tuo progetto. Gli ambienti di sviluppo aumentano la velocità di sviluppo perché è possibile iniziare a scrivere codice immediatamente in un ambiente di sviluppo cloud completamente funzionante e specifico del progetto con un ambiente di sviluppo integrato (IDE) supportato.

Per informazioni sull'utilizzo CodeCatalyst da un IDE, consulta la seguente documentazione.

- [Amazon CodeCatalyst per JetBrains IDEs](#)
- [Amazon CodeCatalyst per VS Code](#)
- [Amazon CodeCatalyst per AWS Cloud9](#)

Interruzione di un ambiente di sviluppo

La `/projects` directory di un ambiente di sviluppo memorizza i file estratti dal repository di origine e il devfile utilizzato per configurare l'ambiente di sviluppo. La `/home` directory, che è vuota al momento della creazione di Dev Environment, memorizza i file creati durante l'utilizzo del Dev Environment. Tutto ciò che si trova `/home` nelle directory `/projects` e di un ambiente di sviluppo viene archiviato in modo persistente, quindi puoi smettere di lavorare in un ambiente di sviluppo se devi passare a un altro ambiente di sviluppo, repository o progetto.

Warning

Un ambiente di sviluppo non si verificherà il timeout se alcune istanze, inclusi browser Web, shell remote e, rimangono connesse. IDEs Quindi, assicurati di chiudere tutte le istanze connesse per evitare di incorrere in costi aggiuntivi.

Un ambiente di sviluppo si interromperà automaticamente se è inattivo per il periodo di tempo selezionato nei campi Timeout durante la creazione dell'ambiente di sviluppo. Puoi fermare l'ambiente di sviluppo prima che diventi inattivo. Se hai scelto Nessun timeout quando hai creato il tuo ambiente di sviluppo, l'ambiente di sviluppo non si interromperà automaticamente. Invece, verrà eseguito continuamente.

 Warning

Se interrompi un ambiente di sviluppo associato a una connessione VPC eliminata, non può essere ripristinato.

Per interrompere un ambiente di sviluppo dalla pagina Dev Environments

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Passa al progetto in cui desideri interrompere un ambiente di sviluppo.
3. Nel riquadro di navigazione, scegli Codice.
4. Scegli Dev Environments.
5. Scegli il pulsante di opzione per l'ambiente di sviluppo che desideri interrompere.
6. Dal menu Azioni, scegli Stop.

 Note

L'utilizzo dell'elaborazione viene fatturato solo mentre l'ambiente di sviluppo è in esecuzione, ma l'utilizzo dello storage viene fatturato per tutto il periodo in cui esiste l'ambiente di sviluppo. Interrompi il tuo ambiente di sviluppo quando non è in uso per interrompere la fatturazione di calcolo.

Ripresa del lavoro in un ambiente di sviluppo

La `/projects` directory di un ambiente di sviluppo memorizza i file estratti dal repository di origine e il devfile utilizzato per configurare l'ambiente di sviluppo. La `/home` directory, che è vuota al momento della creazione di Dev Environment, memorizza i file creati durante l'utilizzo del Dev Environment. Tutto ciò che si trova `/home` nelle directory `/projects` e di un ambiente di sviluppo viene archiviato in modo persistente, quindi puoi smettere di lavorare in un ambiente di sviluppo se devi passare a un altro ambiente di sviluppo, repository o progetto e riprendere a lavorare nel tuo ambiente di sviluppo in un secondo momento.

Un ambiente di sviluppo si interromperà automaticamente se è inattivo per il periodo di tempo selezionato nei campi Timeout durante la creazione dell'ambiente di sviluppo. È necessario chiudere la scheda del AWS Cloud9 browser per rendere inattivo l'ambiente di sviluppo.

 Note

L'ambiente di sviluppo è ancora disponibile e funzionante anche se si elimina il ramo con cui è stato creato l'ambiente di sviluppo. Se vuoi riprendere a lavorare in un ambiente di sviluppo per il quale hai eliminato il ramo, crea un nuovo ramo e inserisci le modifiche ad esso.

Per riprendere un ambiente di sviluppo dalla pagina di panoramica

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al progetto in cui desideri riprendere un ambiente di sviluppo e vai alla sezione I miei ambienti di sviluppo.
3. Scegli Resume in (IDE).
 - Per JetBrains IDEs, scegli JetBrains Gateway-EAP quando ti viene richiesto di scegliere un'applicazione per aprire il collegamento -gateway. JetBrains Scegliete Apri collegamento per confermare quando richiesto.
 - Per l'IDE VS Code, scegli VS Code quando ti viene richiesto di scegliere un'applicazione per aprire il link VS Code. Scegli Apri link per confermare.

Per riprendere un ambiente di sviluppo dal repository di origine

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri riprendere un ambiente di sviluppo.
3. Nel pannello di navigazione, scegli Codice.
4. Scegli Source Repositories.
5. Scegli il repository di origine che contiene l'ambiente di sviluppo che desideri riprendere.
6. Scegli il nome della filiale per visualizzare un menu a discesa delle tue filiali, quindi scegli la filiale.
7. Scegli Resume Dev Environment.
 - Per JetBrains IDEs, scegli Open Link per confermare quando viene richiesto di consentire a questo sito di aprire il collegamento JetBrains -gateway con Gateway? JetBrains .
 - Per l'IDE VS Code, scegli Apri collegamento per confermare quando viene richiesto di consentire a questo sito di aprire il collegamento VS Code con Visual Studio Code? .

Per riprendere un ambiente di sviluppo dalla pagina Dev Environments

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al progetto in cui desideri riprendere un ambiente di sviluppo.
3. Nel pannello di navigazione, scegli Codice.
4. Scegli Dev Environments.
5. Dalla colonna IDE, scegli Resume in (IDE) per Dev Environment.
 - Per JetBrains IDEs, scegli Open Link per confermare quando ti viene richiesto di Consentire a questo sito di aprire il collegamento JetBrains -gateway con Gateway? JetBrains .
 - Per l'IDE VS Code, scegli Apri collegamento per confermare quando viene richiesto di consentire a questo sito di aprire il collegamento VS Code con Visual Studio Code? .

Note

La ripresa del lavoro in un ambiente di sviluppo può richiedere alcuni minuti.

Modifica di un ambiente di sviluppo

Mentre il tuo IDE è in esecuzione, puoi modificare l'ambiente di sviluppo. Se modifichi il timeout di calcolo o di inattività, l'ambiente di sviluppo verrà riavviato dopo aver salvato le modifiche.

Per modificare un ambiente di sviluppo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri modificare un ambiente di sviluppo.
3. Nel pannello di navigazione, scegli Codice.
4. Scegli Dev Environments.
5. Scegli l'ambiente di sviluppo che desideri modificare.
6. Scegli Modifica.
7. Apporta le modifiche desiderate al timeout di calcolo o di inattività.
8. Scegli Save (Salva).

Eliminazione di un ambiente di sviluppo

Quando hai finito di lavorare sul contenuto archiviato nel tuo ambiente di sviluppo, puoi eliminare l'ambiente di sviluppo. Crea un nuovo ambiente di sviluppo per lavorare su nuovi contenuti. Se elimini il tuo ambiente di sviluppo, il contenuto persistente verrà eliminato definitivamente. Prima di eliminare il tuo ambiente di sviluppo, assicurati di eseguire il commit e di inviare le modifiche al codice nell'archivio dei sorgenti originale di Dev Environment. Dopo aver eliminato il Dev Environment, la fatturazione di elaborazione e archiviazione per Dev Environment verrà interrotta.

Dopo aver eliminato il tuo ambiente di sviluppo, potrebbero essere necessari alcuni minuti per aggiornare la quota di spazio di archiviazione. Se hai raggiunto la quota di archiviazione, non potrai creare un nuovo ambiente di sviluppo durante questo periodo.

Important

L'eliminazione di un ambiente di sviluppo non può essere annullata. Dopo aver eliminato un ambiente di sviluppo, non è più possibile ripristinarlo.

Per eliminare un ambiente di sviluppo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri eliminare un ambiente di sviluppo.
3. Nel riquadro di navigazione, scegli Codice.
4. Scegli Dev Environments.
5. Scegli l'ambiente di sviluppo che desideri eliminare.
6. Scegli Elimina.
7. Invio **delete** per confermare l'eliminazione dell'ambiente di sviluppo.
8. Scegli Elimina.

Note

Prima di eliminare una connessione VPC nel tuo spazio, assicurati di rimuovere l'ambiente di sviluppo associato a quel VPC.

Anche se elimini un ambiente di sviluppo, potresti non eliminare l'interfaccia di rete nel VPC. Assicurati di ripulire le tue risorse secondo necessità. Se si verifica un errore quando si

elimina un ambiente di sviluppo connesso a VPC, è necessario [scollegare](#) la connessione obsoleta ed [eliminarla](#) dopo aver verificato che non viene utilizzata.

Connessione a un ambiente di sviluppo tramite SSH

Puoi connetterti al tuo ambiente di sviluppo utilizzando SSH per eseguire azioni senza limitazioni, come il port forwarding, il caricamento e il download di file e l'utilizzo di altro. IDEs

Note

Se desideri continuare a utilizzare SSH per un periodo prolungato dopo aver chiuso la scheda o la finestra IDE, assicurati di impostare un timeout elevato per il tuo ambiente di sviluppo in modo che non si interrompa a causa dell'inattività nell'IDE.

Prerequisiti

- È necessario uno dei seguenti sistemi operativi:
 - Windows 10 o versioni successive e OpenSSH abilitato
 - macOS e Bash versione 3 o superiore
 - Linux con yum dpkg o gestori di rpm pacchetti e Bash versione 3 o successiva
- È inoltre necessaria la AWS CLI versione 2.9.4 o successiva.

Per connettersi a un ambiente di sviluppo tramite SSH

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa al progetto in cui desideri connetterti a un ambiente di sviluppo tramite SSH.
3. Nel riquadro di navigazione, scegli Codice.
4. Scegli Dev Environments.
5. Scegli un ambiente di sviluppo in esecuzione a cui desideri connetterti tramite SSH.
6. Scegli Connect via SSH, scegli il sistema operativo desiderato ed esegui le seguenti operazioni:
 - Se non l'hai già fatto, incolla ed esegui il primo comando nel terminale specificato. Il comando scarica uno script ed esegue le seguenti modifiche nell'ambiente locale in modo che possiate connettervi al vostro ambiente di sviluppo tramite SSH:

- Installa il plugin [Session Manager per AWS CLI](#)
- Modifica il locale AWS Config e aggiunge un CodeCatalyst profilo in modo da poter eseguire l'accesso SSO. Per ulteriori informazioni, consulta [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#).
- Modifica la configurazione SSH locale e aggiunge la configurazione richiesta per la connessione al tuo ambiente di sviluppo tramite SSH.
- Aggiunge uno script nella `~/.aws/codecatalyst-dev-env` directory utilizzata dal client SSH per connettersi al tuo ambiente di sviluppo. Questo script chiama l'[CodeCatalyst StartDevEnvironmentSession API](#) e utilizza il AWS Systems Manager Session Manager plugin per stabilire una AWS Systems Manager sessione con il tuo Dev Environment, che viene utilizzato dal client SSH locale per connettersi in modo sicuro all'ambiente di sviluppo remoto.
- Accedi ad Amazon CodeCatalyst utilizzando AWS SSO utilizzando il secondo comando. [Questo comando richiede e recupera le credenziali in modo che lo script nella `~/.aws/codecatalyst-dev-env` directory possa chiamare l'API. CodeCatalyst StartDevEnvironmentSession](#) Questo comando deve essere eseguito ogni volta che le credenziali scadono. Quando esegui l'ultimo comando in modalità modale (`ssh<destination>`), riceverai un errore se le tue credenziali sono scadute o se non hai eseguito l'accesso SSO come indicato in questo passaggio.
- Connect al Dev Environment specificato usando SSH usando il terzo comando. Questo comando ha la seguente struttura:

```
ssh codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

Puoi anche usare questo comando per eseguire altre azioni consentite dal client SSH, come il port forwarding o il caricamento e lo scaricamento di file:

- Port forwarding:

```
ssh -L <local-port>:127.0.0.1:<remote-port> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

- Caricamento di un file nella home directory del tuo ambiente di sviluppo:

```
scp -0 </path-to-local-file> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>:</path-to-remote-file-or-directory>
```

Configurazione di un devfile per un ambiente di sviluppo

Un devfile è uno standard aperto che ti aiuta a personalizzare gli ambienti di sviluppo di tutto il team. Un devfile è un file YAML che codifica gli strumenti di sviluppo necessari. Configurando un devfile, puoi predeterminare gli strumenti di progetto e le librerie di applicazioni di cui hai bisogno e Amazon li CodeCatalyst installa nel tuo ambiente di sviluppo per te. Il devfile è specifico del repository per il quale è stato creato e puoi creare un devfile separato per ogni repository. Il tuo ambiente di sviluppo supporta comandi ed eventi e fornisce un'immagine devfile universale predefinita.

Se si crea un progetto utilizzando il blueprint vuoto, è possibile creare un devfile manualmente. Se si crea un progetto utilizzando un blueprint diverso, CodeCatalyst crea automaticamente un devfile. La `/projects` directory di un ambiente di sviluppo memorizza i file che vengono estratti dal repository di origine e dal devfile. La `/home` directory, che è vuota quando crei per la prima volta un ambiente di sviluppo, memorizza i file che crei durante l'utilizzo del tuo ambiente di sviluppo. Tutto ciò che si trova nelle `/home` directory `/projects` and di un ambiente di sviluppo viene archiviato in modo persistente.

Note

La `/home` cartella cambia solo se si modifica il nome del componente devfile o devfile. Se si modifica il nome del componente devfile o devfile, il contenuto della directory viene sostituito e i dati della `/home` directory precedente non possono essere recuperati.

Se create un ambiente di sviluppo con un repository di origine che non contiene un devfile nella radice, o se create un ambiente di sviluppo senza un archivio di origine, al repository di origine viene applicato automaticamente un devfile universale predefinito. La stessa immagine devfile universale predefinita viene utilizzata per tutti. IDEs CodeCatalyst attualmente supporta la versione 2.0.0 di devfile. Per ulteriori informazioni sul devfile, vedere [Devfile](#) schema - Versione 2.0.0.

Note

Puoi includere solo immagini di container pubblici nel tuo devfile.

Nota che gli ambienti di sviluppo connessi a VPC supportano solo le seguenti immagini devfile:

- Immagine universale

- Immagini Amazon ECR private, se il repository si trova nella stessa regione del VPC

Argomenti

- [Modifica di un devfile del repository per un ambiente di sviluppo](#)
- [Funzionalità Devfile supportate da CodeCatalyst](#)
- [Esempio di devfile per un ambiente di sviluppo](#)
- [Risoluzione dei problemi relativi a un devfile del repository utilizzando la modalità di ripristino](#)
- [Specificare immagini devfile universali per un ambiente di sviluppo](#)
- [Comandi Devfile](#)
- [Eventi Devfile](#)
- [Componenti Devfile](#)

Modifica di un devfile del repository per un ambiente di sviluppo

Utilizzate la seguente procedura per modificare un devfile del repository per un ambiente di sviluppo.

Modifica di un devfile del repository per un ambiente di sviluppo in CodeCatalyst

Per modificare il devfile del repository

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Passa al progetto che contiene l'archivio dei sorgenti per il quale desideri modificare il devfile.
3. Nel riquadro di navigazione, scegli Codice.
4. Scegli Source Repositories.
5. Scegliete il repository di origine che contiene il devfile che desiderate modificare.
6. Dall'elenco dei file, scegli il `devfile.yaml` file.
7. Scegli Modifica.
8. Modifica il devfile.
9. Scegli Commit o crea una pull request in modo che un membro del team possa rivedere e approvare le modifiche.

Note

Se modifichi il tuo devfile, devi riavviarlo per rendere effettive le modifiche. Questo può essere fatto eseguendo `/aws/mde/mde start --location devfile.yaml`. Se c'è un problema durante l'avvio del file dev, entrerà in modalità di ripristino. Tuttavia, se modifichi un devfile associato a un ambiente di sviluppo connesso a VPC, devi invece riavviare l'ambiente di sviluppo affinché le modifiche abbiano effetto.

È possibile verificare quale devfile viene utilizzato eseguendo `/aws/mde/mde status`. Il campo `location` contiene il percorso del devfile relativo alla cartella dell'ambiente.

```
{
  "status": "STABLE",
  "location": "devfile.yaml"
}
```

Puoi anche spostare il devfile predefinito nel tuo `/projects/devfile.yaml` repository del codice sorgente. Per aggiornare la posizione del devfile, usa il seguente comando: `/aws/mde/mde start --location repository-name/devfile.yaml`

Modifica di un devfile di archivio per un ambiente di sviluppo in un IDE

Per modificare la configurazione di un ambiente di sviluppo, è necessario modificare il devfile. Ti consigliamo di modificare il devfile in un IDE supportato e quindi di aggiornare il tuo ambiente di sviluppo, ma puoi anche modificare il devfile dalla radice del repository di origine in CodeCatalyst. Se modifichi il devfile in un IDE supportato, devi eseguire il commit e inviare le modifiche al repository di origine o creare una richiesta pull in modo che un membro del team possa rivedere e approvare le modifiche al devfile.

- [Modifica del file devfile del repository per un ambiente di sviluppo in AWS Cloud9](#)
- [Modifica del devfile del repository per un ambiente di sviluppo in VS Code](#)
- [Modifica del file devfile del repository per un ambiente di sviluppo in JetBrains](#)

Funzionalità Devfile supportate da CodeCatalyst

CodeCatalyst supporta le seguenti funzionalità di devfile nella versione 2.0.0. Per ulteriori informazioni sul devfile, vedere [Devfile](#) schema - Versione 2.0.0.

Funzionalità	Tipo
exec	Comando
postStart	Event
container	Componente
args	Proprietà dei componenti
env	Proprietà dei componenti
mountSources	Proprietà dei componenti
volumeMounts	Proprietà dei componenti

Esempio di devfile per un ambiente di sviluppo

Quello che segue è un esempio di un semplice devfile.

```
schemaVersion: 2.0.0
metadata:
  name: al2
components:
  - name: test
    container:
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      mountSources: true
      command: ['sleep', 'infinity']
  - name: dockerstore
commands:
  - id: setupscript
    exec:
      component: test
      commandLine: "chmod +x script.sh"
      workingDir: /projects/devfiles
  - id: executescript
    exec:
      component: test
      commandLine: "/projects/devfiles/script.sh"
  - id: yumupdate
```

```
exec:
  component: test
  commandLine: "yum -y update --security"
events:
  postStart:
    - setupscript
    - executescript
    - yumupdate
```

I registri di avvio, comando ed eventi di Devfile vengono acquisiti e archiviati in. `/aws/mde/logs` Per eseguire il debug del comportamento di devfile, avvia il tuo ambiente di sviluppo utilizzando un devfile funzionante e accedi ai log.

Risoluzione dei problemi relativi a un devfile del repository utilizzando la modalità di ripristino

Se c'è un problema durante l'avvio del file dev, questo entrerà in modalità di ripristino in modo che possiate comunque connettervi al vostro ambiente e correggere il vostro devfile. In modalità di ripristino, l'esecuzione `/aws/mde/mde status` non conterrà la posizione del tuo devfile.

```
{
    "status": "STABLE"
}
```

Puoi controllare l'errore nei log sotto `/aws/mde/logs`, correggere il devfile e riprovare a eseguire. `/aws/mde/mde start`

Specificare immagini devfile universali per un ambiente di sviluppo

L'immagine universale predefinita include i linguaggi di programmazione più utilizzati e gli strumenti correlati che possono essere utilizzati per l'IDE. Se non viene specificata alcuna immagine, CodeCatalyst fornisce questa immagine e contiene strumenti gestiti da CodeCatalyst. Per rimanere informati sulle nuove versioni di immagini, consulta [iscrizione alle notifiche universali con immagini con SNS](#).

Amazon supporta CodeCatalyst attivamente le seguenti immagini devfile:

Versione dell'immagine	Identificatore dell'immagine
Universal image 4.0	<code>public.ecr.aws/aws-mde/universal-image:4.0</code>
Universal image 5.0	<code>public.ecr.aws/aws-mde/universal-image:5.0</code>

Note

Puoi anche usarla `public.ecr.aws/aws-mde/universal-image:latest` per ottenere l'immagine più recente, che è attualmente disponibile `public.ecr.aws/aws-mde/universal-image:5.0`.

CodeCatalyst ha reso obsolete le seguenti immagini. È ancora possibile utilizzare queste immagini, ma non verranno memorizzate nella cache dell'host di build e comporteranno un aumento dei tempi di avvio dell'ambiente di sviluppo.

Versione dell'immagine	Identificatore dell'immagine	Data di ritiro
Universal image 1.0	<code>public.ecr.aws/aws-mde/universal-image:1.0</code>	16 agosto 2024
Universal image 2.0	<code>public.ecr.aws/aws-mde/universal-image:2.0</code>	16 agosto 2024
Universal image 3.0	<code>public.ecr.aws/aws-mde/universal-image:3.0</code>	30 luglio 2025

Note

Se lo stai utilizzando AWS Cloud9, il completamento automatico non funzionerà per PHP, Ruby e CSS dopo l'aggiornamento a `universal-image:3.0`

Argomenti

- [Iscrizione alle notifiche universali con immagini con SNS](#)
- [Versioni runtime di Universal Image 4.0](#)
- [Versioni di runtime di Universal Image 5.0](#)

Iscrizione alle notifiche universali con immagini con SNS

CodeCatalyst fornisce un servizio universale di notifica delle immagini. Puoi usarlo per abbonarti a un argomento di Amazon Simple Notification Service (SNS) che ti avvisa quando vengono rilasciati aggiornamenti CodeCatalyst universali delle immagini. Per ulteriori informazioni sugli argomenti relativi a SNS, consulta [What is Amazon Simple Notification Service?](#) .

Ogni volta che vengono rilasciate nuove immagini universali, inviamo notifiche agli abbonati; questa sezione descrive come abbonarsi agli aggiornamenti CodeCatalyst universali delle immagini.

Messaggio di esempio

```
{
  "Type": "Notification",
  "MessageId": "123456789",
  "TopicArn": "arn:aws:sns:us-east-1:1234657890:universal-image-updates",
  "Subject": "New Universal Image Release",
  "Message": {
    "v1": {
      "Message": "A new version of the Universal Image has been released. You are now able to launch new DevEnvironments using this image.",
      "image": {
        "release_type": "MAJOR VERSION",
        "image_name": "universal-image",
        "image_version": "2.0",
        "image_uri": "public.ecr.aws/amazonlinux/universal-image:2.0"
      }
    }
  }
}
```

```
},  
  "Timestamp": "2021-09-03T19:05:57.882Z",  
  "UnsubscribeURL": "example url"  
}
```

Per abbonarsi agli aggiornamenti CodeCatalyst universali delle immagini utilizzando la console Amazon SNS

1. [Apri la console Amazon SNS nel pannello di controllo.](#)
2. Nella barra di navigazione, scegli il tuo Regione AWS.
3. Nel riquadro di navigazione scegliere Subscriptions (Sottoscrizioni), quindi selezionare Create subscription (Crea sottoscrizione).
4. In Argomento ARN, immettere. `arn:aws:sns:us-east-1:089793673375:universal-image-updates`
5. Per Protocol (Protocollo), scegli Email.
6. In Endpoint, fornisci un indirizzo e-mail. Questo indirizzo email verrà utilizzato per ricevere notifiche.
7. Scegliere Create Subscription (Crea iscrizione).
8. Riceverai un'e-mail di conferma con oggetto "AWS Notifica - Conferma dell'abbonamento». Apri l'e-mail e scegli Conferma iscrizione.

Per annullare l'iscrizione agli aggiornamenti CodeCatalyst universali delle immagini utilizzando la console Amazon SNS

1. [Apri la console Amazon SNS nel pannello di controllo.](#)
2. Nella barra di navigazione, scegli il tuo Regione AWS.
3. Nel riquadro di navigazione, scegli Abbonamenti, quindi seleziona l'abbonamento da cui desideri annullare l'iscrizione.
4. Scegli Azioni, quindi scegli Elimina abbonamenti.
5. Scegliere Delete (Elimina).

Versioni runtime di Universal Image 4.0

La tabella seguente elenca i runtime disponibili per `universal-image:4.0`.

universal-image:4.0versioni di runtime

Nome runtime	Versione	Versione principale e secondaria specifica
aws cli	2.11	aws-cli: 2.x
docker compone	2.17	docker-compose: 2.x
dotnet	8.0	dotnet: 8.x
golang	1,22	golang: 1.x
java	corretto 21	java: corretto21.x
nodejs	20.6	nodejs: 20.x
php	8.2	php: 8.x
python	3.9	python: 3.x
	3,12	
ruby	3.3	ruby: 3.x
terraformare	1.5	terraform: 1.x

Versioni di runtime di Universal Image 5.0

La tabella seguente elenca i runtime disponibili per `universal-image:5.0`.

universal-image:5.0versioni di runtime

Nome runtime	Versione	Versione principale e secondaria specifica
aws cli	2.25	aws-cli: 2.x
docker compone	2.34	docker-compose: 2.x
dotnet	8.0	dotnet: 8.x
golang	1,24	golang: 1.x

Nome runtime	Versione	Versione principale e secondaria specifica
java	corretto 21	java: corretto21.x
nodejs	22.0	nodejs: 22.x
php	8,3,16	php: 8.x
python	3,12	python: 3.x
ruby	34,2	ruby: 3.x
terraform	1.10.5	terraform: 1.x

Comandi Devfile

Attualmente, supporta CodeCatalyst solo i `exec` comandi nel tuo devfile. Per ulteriori informazioni, consulta [Aggiungere comandi nella documentazione](#) di Devfile.io.

L'esempio seguente mostra come specificare i `exec` comandi nel tuo devfile.

```
commands:
- id: setupscript
  exec:
    component: test
    commandLine: "chmod +x script.sh"
    workingDir: /projects/devfiles
- id: executescript
  exec:
    component: test
    commandLine: "./projects/devfiles/script.sh"
- id: updateyum
  exec:
    component: test
    commandLine: "yum -y update --security"
```

Dopo esserti connesso al tuo Dev Environment, puoi eseguire comandi definiti tramite il terminale.

```
/aws/mde/mde command <command-id>
/aws/mde/mde command executescript
```

Per i comandi a esecuzione prolungata, puoi utilizzare il flag di streaming `-s` per visualizzare l'esecuzione del comando in tempo reale.

```
/aws/mde/mde -s command <command-id>
```

Note

`command-id` deve essere in minuscolo.

Parametri Exec supportati da CodeCatalyst

CodeCatalyst supporta i seguenti exec parametri nella versione devfile 2.0.0.

- `commandLine`
- `component`
- `id`
- `workingDir`

Eventi Devfile

Attualmente, supporta CodeCatalyst solo `postStart` gli eventi nel tuo devfile. Per ulteriori informazioni, consultate la [postStartObject](#) documentazione di Devfile.io.

L'esempio seguente mostra come aggiungere associazioni di `postStart` eventi nel tuo devfile.

```
commands:  
  - id: executescript  
    exec:  
      component: test  
      commandLine: "./projects/devfiles/script.sh"  
  - id: updateyum  
    exec:  
      component: test  
      commandLine: "yum -y update --security"
```

```
events:
  postStart:
    - updateyum
    - executescript
```

Dopo l'avvio, il tuo Dev Environment eseguirà `postStart` i comandi specificati nell'ordine in cui sono definiti. Se un comando fallisce, Dev Environment continuerà a funzionare e l'output dell'esecuzione verrà memorizzato nei log sotto `/aws/mde/logs`.

Componenti Devfile

Attualmente, supporta CodeCatalyst solo i container componenti del tuo devfile. Per ulteriori informazioni, vedete [Aggiungere componenti nella documentazione](#) di Devfile.io.

L'esempio seguente mostra come aggiungere un comando di avvio al contenitore nel devfile.

```
components:
  - name: test
    container:
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      command: ['sleep', 'infinity']
```

Note

Quando il contenitore ha un comando di immissione di breve durata, è necessario includerlo `command: ['sleep', 'infinity']` per mantenere il contenitore in esecuzione.

CodeCatalyst supporta anche le seguenti proprietà nel componente del contenitore: `argsenv`, `mountSources`, `evolumeMounts`.

Associazione di una connessione VPC a un ambiente di sviluppo

Una connessione VPC è una CodeCatalyst risorsa che contiene tutte le configurazioni necessarie al flusso di lavoro per accedere a un VPC. Gli amministratori dello spazio possono aggiungere le proprie connessioni VPC nella console CodeCatalyst Amazon per conto dei membri dello spazio. Aggiungendo una connessione VPC, i membri dello spazio possono eseguire azioni di flusso di lavoro e creare ambienti di sviluppo che rispettano le regole di rete e possono accedere alle risorse nel VPC associato.

 Important

Gli ambienti di sviluppo con una connessione VPC non supportano [repository di sorgenti di terze parti](#) collegati a. CodeCatalyst

È possibile associare un ambiente di sviluppo a una connessione VPC solo al momento della creazione dell'ambiente di sviluppo. Non puoi modificare la connessione VPC associata al tuo ambiente di sviluppo dopo averla creata. Se desideri utilizzare una connessione VPC diversa, devi eliminare l'ambiente di sviluppo corrente e crearne uno nuovo.

 Note

Gli ambienti di sviluppo possono essere associati solo a una connessione VPC con un AWS account che ha accesso al progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

Tieni presente che gli ambienti di sviluppo utilizzano diverse AWS risorse e servizi al momento della creazione. Ciò significa che gli ambienti di sviluppo si connettono ai seguenti AWS servizi:

- Amazon CodeCatalyst
- AWS SSM
- AWS KMS
- Amazon ECR
- Amazon CloudWatch
- Amazon ECS

 Note

Kit di strumenti AWS non supporta la creazione di ambienti di sviluppo con una connessione VPC associata. Tieni inoltre presente che se utilizzi un IDE diverso da AWS Cloud9, potresti riscontrare tempi di caricamento di circa cinque minuti.

È necessario disporre del ruolo di amministratore dello spazio o del ruolo Power user per gestire le connessioni VPC a livello di spazio. Per ulteriori informazioni VPCs, consulta [Managing Amazon VPCs CodeCatalyst nella CodeCatalyst Administrator Guide](#).

Quote per ambienti di sviluppo in CodeCatalyst

La tabella seguente descrive le quote e i limiti per gli ambienti di sviluppo in Amazon CodeCatalyst. Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Numero di ore mensili per ambiente di sviluppo	Le ore di lavoro di Dev Environment sono influenzate dai limiti complessivi di archiviazione per lo spazio in uso. Per ulteriori informazioni, consulta Prezzi di e Risoluzione dei problemi con gli ambienti di sviluppo .
Quantità di storage in Dev Environment per spazio	Lo spazio di archiviazione in Dev Environment è influenzato dai limiti complessivi di archiviazione per lo spazio dell'utente. Per ulteriori informazioni, consulta Prezzi di e Risoluzione dei problemi con gli ambienti di sviluppo .
Quantità di calcolo in Dev Environment	L'elaborazione di Dev Environment è influenzata dai limiti complessivi di archiviazione per lo spazio in uso. Per ulteriori informazioni, consulta Prezzi di e Risoluzione dei problemi con gli ambienti di sviluppo .

Pubblica e condividi pacchetti software in CodeCatalyst

Amazon CodeCatalyst contiene un servizio di repository di pacchetti completamente gestito che consente al team di sviluppo di archiviare e condividere in modo semplice e sicuro i pacchetti software utilizzati per lo sviluppo di applicazioni. Questi pacchetti sono archiviati in archivi di pacchetti, che vengono creati e organizzati all'interno di progetti in CodeCatalyst.

Un singolo archivio di pacchetti può memorizzare pacchetti di ogni tipo di pacchetto supportato. CodeCatalyst supporta i seguenti formati di pacchetti:

- npm
- Maven
- NuGet
- Python

I pacchetti in un archivio di pacchetti possono essere scoperti e condivisi tra i membri del progetto che contiene il repository.

Per pubblicare e utilizzare i pacchetti da un repository, configura un gestore di pacchetti per utilizzare l'endpoint (URL) del repository. È quindi possibile utilizzare il gestore di pacchetti per pubblicare i pacchetti nel repository. Puoi usare gestori di pacchetti come Maven, Gradle, npm, yarn, nuget, dotnet, pip e twine.

Puoi anche configurare i flussi di lavoro per utilizzare gli archivi di pacchetti. CodeCatalyst
CodeCatalyst Per ulteriori informazioni sull'utilizzo dei pacchetti nei flussi di lavoro, consulta.

[Connessione degli archivi di pacchetti ai flussi di lavoro](#)

È possibile rendere i pacchetti in un repository di pacchetti disponibili per un altro repository dello stesso progetto aggiungendolo come repository upstream. Tutte le versioni dei pacchetti disponibili nell'archivio upstream sono disponibili anche nel repository downstream. Per ulteriori informazioni, consulta [Configurazione e utilizzo degli archivi upstream](#).

È possibile rendere disponibili pacchetti open source nel CodeCatalyst repository creando un tipo speciale di repository chiamato gateway. L'upstreaming verso un repository gateway ti consente di utilizzare pacchetti da archivi pubblici popolari come npmjs.com e pypi.org e di memorizzarli automaticamente nella cache del tuo repository. CodeCatalyst Per ulteriori informazioni, consulta [Connessione a archivi esterni pubblici](#).

Argomenti

- [Concetti di pacchetti](#)
- [Configurazione e utilizzo degli archivi di pacchetti](#)
- [Configurazione e utilizzo degli archivi upstream](#)
- [Connessione a archivi esterni pubblici](#)
- [Pubblicazione e modifica di pacchetti](#)
- [Utilizzo di npm](#)
- [Usare Maven](#)
- [Usando NuGet](#)
- [Uso di Python](#)
- [Quote per i pacchi](#)

Concetti di pacchetti

Ecco alcuni concetti e termini da conoscere per la gestione, la pubblicazione o il consumo di pacchetti in CodeCatalyst.

Pacchetti

Un pacchetto è un pacchetto che include sia il software che i metadati necessari per installare il software e risolvere eventuali dipendenze. CodeCatalyst supporta il formato del pacchetto npm.

Un pacchetto è composto da:

- Un nome (ad esempio, webpack è il nome di un popolare pacchetto npm)
- Uno spazio [dei nomi](#) opzionale (ad esempio, in) @types @types/node
- Un insieme di [versioni](#) (ad esempio, 1.0.0, 1.0.1) 1.0.2
- Metadati a livello di pacchetto (ad esempio, tag npm dist)

Namespace dei pacchetti

Alcuni formati di pacchetti supportano nomi di pacchetto gerarchici per organizzare i pacchetti in gruppi logici e per evitare collisioni di nomi. I pacchetti con lo stesso nome possono essere

memorizzati in namespace diversi. Ad esempio, npm supporta gli ambiti e il pacchetto `npm @types/node` ha un ambito e un nome di `@types node`. Ci sono molti altri nomi di pacchetti nell'ambito. `@types` In CodeCatalyst, l'ambito («tipi») viene definito spazio dei nomi del pacchetto e il nome («nodo») viene definito nome del pacchetto. Per i pacchetti Maven, lo spazio dei nomi del pacchetto corrisponde a Maven GroupID. Il pacchetto `Maven org.apache.logging.log4j:log4j` ha un groupID (spazio dei nomi del pacchetto) e un artifactID (nome del pacchetto). `org.apache.logging.log4j log4j` Alcuni formati di pacchetti come Python non supportano nomi gerarchici con un concetto simile a npm scope o Maven GroupID. Se non si dispone di un modo per raggruppare i nomi dei pacchetti, può essere più difficile evitare le collisioni di nomi.

Versioni del pacchetto

Una versione del pacchetto identifica la versione specifica di un pacchetto, ad esempio `@types/node@12.6.9`. Il formato e la semantica del numero di versione variano a seconda dei diversi formati di pacchetto. [Ad esempio, le versioni del pacchetto npm devono essere conformi alla specifica Semantic Versioning.](#) In CodeCatalyst, una versione del pacchetto è composta dall'identificatore di versione, dai package-version-level metadati e da un set di risorse.

Asset

Una risorsa è un singolo file archiviato in CodeCatalyst che è associato a una versione del pacchetto, ad esempio un file npm o un `.tgz` file Maven POM o JAR.

Archivi di pacchetti

[Un archivio di CodeCatalyst pacchetti contiene un insieme di pacchetti, che contengono versioni di pacchetti, ognuna delle quali è mappata a un insieme di risorse.](#) I repository di pacchetti sono poliglotti, il che significa che un singolo repository può contenere pacchetti di qualsiasi tipo supportato. Ogni archivio di pacchetti espone gli endpoint per il recupero e la pubblicazione di pacchetti utilizzando strumenti come NuGet CLIs (`nuget`), `dotnet`, CLI, Maven npm CLI (`npm`) e Python `mvn` (and). CLIs `pip` `twine` Per informazioni sulle quote dei pacchetti CodeCatalyst, incluso il numero di repository di pacchetti che è possibile creare in ogni spazio, vedere. [Quote per i pacchi](#)

È possibile collegare un archivio di pacchetti a un altro impostandolo come archivio upstream. Quando un repository è impostato come upstream, è possibile utilizzare qualsiasi pacchetto dell'upstream e qualsiasi altro repository upstream della catena. Per ulteriori informazioni, consulta [Archivi upstream](#).

I repository gateway sono un tipo speciale di repository di pacchetti che estrae e archivia pacchetti provenienti da autorità esterne ufficiali per la gestione dei pacchetti. Per ulteriori informazioni, consulta [Archivi Gateway](#).

Archivi upstream

È possibile utilizzarlo CodeCatalyst per creare una relazione upstream tra due repository di pacchetti. Un repository di pacchetti è a monte di un altro quando è possibile accedere alle versioni dei pacchetti che contiene dall'endpoint del repository di pacchetti del repository downstream. Con una relazione a monte, i contenuti dei due repository di pacchetti vengono effettivamente uniti dal punto di vista di un client.

Ad esempio, se un gestore di pacchetti richiede una versione del pacchetto che non esiste in un repository, CodeCatalyst cercherà la versione del pacchetto negli archivi upstream configurati. La ricerca nei repository upstream viene effettuata nell'ordine in cui sono configurati e, una volta trovato un pacchetto, interromperà la ricerca. CodeCatalyst

Archivi Gateway

Un repository gateway è un tipo speciale di repository di pacchetti collegato a un'autorità di gestione dei pacchetti esterna e ufficiale supportata. Quando si aggiunge un repository gateway come repository [upstream, è possibile utilizzare i pacchetti dell'](#)autorità di gestione dei pacchetti ufficiale corrispondente. Il repository downstream non comunica con l'archivio pubblico, ma tutto viene intermediato dal repository gateway. I pacchetti utilizzati in questo modo vengono archiviati sia nel repository del gateway che nell'archivio a valle che ha ricevuto la richiesta originale.

I repository gateway sono predefiniti, ma devono essere creati in ogni progetto da utilizzare. L'elenco seguente contiene tutti i repository gateway in cui è possibile creare CodeCatalyst e l'autorità del pacchetto a cui sono collegati.

- npm-public-registry-gatewayfornisce pacchetti npm da npmjs.com.
- maven-central-gatewayfornisce pacchetti Maven dal repository Maven Central.
- google-android-gatewayfornisce pacchetti Maven da Google Android.
- commonsware-gateway fornisce pacchetti Maven da CommonsWare
- gradle-plugins-gatewayfornisce pacchetti Maven da Gradle Plugins.
- nuget-gallery-gatewayfornisce NuGet pacchetti dalla Galleria. NuGet
- pypi-gateway fornisce pacchetti Python dal Python Package Index.

Configurazione e utilizzo degli archivi di pacchetti

In CodeCatalyst, i pacchetti vengono archiviati e gestiti all'interno di archivi di pacchetti. Per pubblicare CodeCatalyst o utilizzare pacchetti da un CodeCatalyst (o da qualsiasi archivio pubblico di pacchetti supportato), è necessario creare un archivio di pacchetti e collegarvi il gestore di pacchetti.

Argomenti

- [Creazione di un archivio di pacchetti](#)
- [Connessione a un archivio di pacchetti](#)
- [Eliminazione di un archivio di pacchetti](#)

Creazione di un archivio di pacchetti

Effettuare le seguenti operazioni per creare un archivio di pacchetti in CodeCatalyst

Per creare un archivio di pacchetti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa al progetto in cui desideri creare un archivio di pacchetti.
3. Dal riquadro di navigazione, scegli Pacchetti.
4. Nella pagina Package repository, scegli Create package repository.
5. Nella sezione Dettagli del repository Package, aggiungi quanto segue:
 - a. Nome del repository. Prendi in considerazione l'utilizzo di un nome descrittivo con dettagli come il nome del progetto o del team o come verrà utilizzato il repository.
 - b. (Facoltativo) Descrizione. Una descrizione del repository è particolarmente utile quando in un progetto sono presenti più repository in più team.
6. Nella sezione Archivi upstream, scegli Seleziona repository upstream per aggiungere tutti gli archivi di pacchetti a cui desideri accedere tramite il tuo repository di pacchetti. CodeCatalyst Puoi aggiungere repository Gateway per connetterti a repository di pacchetti esterni o ad altri repository. CodeCatalyst
 - Quando un pacchetto viene richiesto da un repository di pacchetti, la ricerca nei repository originari verrà effettuata nell'ordine in cui appaiono in questo elenco. Una volta trovato un pacchetto, CodeCatalyst interromperà la ricerca. Per modificare l'ordine dei repository originali, puoi trascinarli nell'elenco.

7. Scegli Crea per creare il tuo repository di pacchetti.

Connessione a un archivio di pacchetti

Per pubblicare o utilizzare pacchetti da CodeCatalyst, è necessario configurare il gestore di pacchetti con le informazioni e le credenziali dell'endpoint del repository dei pacchetti. CodeCatalyst Se non hai creato un repository, puoi farlo seguendo le istruzioni riportate in. [Creazione di un archivio di pacchetti](#)

Per istruzioni su come connettere un gestore di pacchetti a un repository di CodeCatalyst pacchetti, consulta la seguente documentazione.

- [Configurazione e utilizzo di Gradle Groovy](#)
- [Configurazione e utilizzo di mvn](#)
- [Configurazione e utilizzo della CLI nuget o dotnet](#)
- [Configurazione e utilizzo di npm](#)
- [Configurazione di pip e installazione di pacchetti Python](#)
- [Configurazione di Twine e pubblicazione di pacchetti Python](#)

Eliminazione di un archivio di pacchetti

Effettuare le seguenti operazioni per eliminare un archivio di pacchetti in. CodeCatalyst

Per eliminare un archivio di pacchetti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Passa al progetto che contiene l'archivio dei pacchetti che desideri eliminare.
3. Dal riquadro di navigazione, scegli Pacchetti.
4. Nella pagina Package repository, scegli il repository che desideri eliminare.
5. Scegli Elimina.
6. Esamina le informazioni fornite sugli effetti dell'eliminazione di un archivio di pacchetti.
7. Entra **delete** nel campo di immissione e scegli Elimina.

Configurazione e utilizzo degli archivi upstream

Puoi connettere sia i repository gateway che gli altri repository di CodeCatalyst pacchetti, come stream upstream ai tuoi repository di pacchetti. Ciò consente a un client di gestione dei pacchetti di accedere ai pacchetti contenuti in più di un archivio di pacchetti utilizzando un singolo endpoint di repository di pacchetti. Di seguito sono riportati i principali vantaggi dell'utilizzo degli archivi upstream:

- Devi solo configurare il tuo gestore di pacchetti con un singolo endpoint di repository da estrarre da più fonti.
- I pacchetti utilizzati da un repository upstream vengono archiviati nel repository downstream, il che garantisce la disponibilità dei pacchetti anche in caso di interruzioni impreviste del repository upstream o se i pacchetti nell'archivio upstream vengono eliminati.

È possibile aggiungere repository upstream quando si crea un archivio di pacchetti. È inoltre possibile aggiungere o rimuovere repository upstream dagli archivi di pacchetti esistenti nella console.

CodeCatalyst

Quando si aggiunge un repository gateway come repository upstream, l'archivio dei pacchetti viene collegato al corrispondente repository pubblico di pacchetti del repository gateway. Per un elenco degli archivi pubblici di pacchetti supportati, vedere. [Archivi di pacchetti esterni supportati e relativi repository gateway](#)

È possibile collegare più repository tra loro come repository upstream. Ad esempio, supponiamo che il tuo team crei un repository denominato `project-repo` e stia già utilizzando un altro repository denominato `a cui è stato npm-public-registry-gateway` aggiunto come repository upstream, `team-repo` che è collegato al repository pubblico npm, `npmjs.com`. Puoi aggiungere come repository upstream a `team-repo` `project-repo`. In questo caso, devi solo configurare il tuo gestore di pacchetti da utilizzare per `project-repo` estrarre i pacchetti da `project-repo`, `team-repo` e `npm-public-registry-gateway`, e `npmjs.com`.

Argomenti

- [Aggiungere un repository upstream](#)
- [Modifica dell'ordine di ricerca dei repository upstream](#)
- [Richiesta di una versione del pacchetto con repository upstream](#)
- [Rimozione di un repository upstream](#)

Aggiungere un repository upstream

L'aggiunta di un archivio pubblico di pacchetti o di un altro CodeCatalyst archivio di pacchetti come repository upstream al repository downstream rende tutti i pacchetti nell'archivio upstream disponibili ai gestori di pacchetti collegati al repository downstream.

Per aggiungere un repository upstream

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Nella pagina Package repository, scegli il repository di pacchetti a cui desideri aggiungere un repository upstream.
3. Sotto il nome del repository dei pacchetti, scegli Upstreams e scegli Seleziona repository upstream.
4. In Seleziona il tipo di upstream, scegli una delle seguenti opzioni:
 - Archivi Gateway

È possibile scegliere da un elenco di repository gateway disponibili.

Note

Per connettersi alle autorità pubbliche di gestione dei pacchetti esterne come Maven Central, npmjs.com o Nuget Gallery, CodeCatalyst utilizza gli archivi gateway come repository intermedi che cercano e archiviano i pacchetti estratti da repository esterni. Ciò consente di risparmiare tempo e trasferimento di dati poiché tutti gli archivi di pacchetti di un progetto utilizzeranno i pacchetti del repository intermediario del gateway. Per ulteriori informazioni, consulta [Connessione a archivi esterni pubblici](#).

- CodeCatalyst archivi

Puoi scegliere da un elenco di repository di CodeCatalyst pacchetti disponibili nel tuo progetto.

5. Dopo aver selezionato tutti i repository che desideri aggiungere come repository upstream, scegli Seleziona, quindi scegli Salva.

Per ulteriori informazioni sulla modifica dell'ordine di ricerca dei repository upstream, consulta [Modifica dell'ordine di ricerca dei repository upstream](#)

Dopo aver aggiunto un repository upstream, puoi utilizzare un gestore di pacchetti collegato al tuo repository locale per recuperare i pacchetti dal repository upstream. Non è necessario aggiornare la configurazione del gestore di pacchetti. Per ulteriori informazioni sulla richiesta delle versioni dei pacchetti da un repository upstream, consulta [Richiesta di una versione del pacchetto con repository upstream](#)

Modifica dell'ordine di ricerca dei repository upstream

CodeCatalyst cerca negli archivi upstream nell'ordine di ricerca configurato. Quando viene trovato un pacchetto, CodeCatalyst interrompe la ricerca. È possibile modificare l'ordine in cui vengono cercati i pacchetti nei repository originali.

Per modificare l'ordine di ricerca dei repository upstream

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Nella pagina Package repository, scegli l'archivio dei pacchetti di cui desideri modificare l'ordine di ricerca del repository upstream.
3. Sotto il nome del repository dei pacchetti, scegli Upstreams.
4. Nella sezione Archivi upstream, puoi visualizzare i repository upstream e il loro ordine di ricerca. Per modificare l'ordine di ricerca, trascina i repository nell'elenco.
5. Quando hai finito di modificare l'ordine di ricerca dei repository originali, scegli Salva.

Richiesta di una versione del pacchetto con repository upstream

L'esempio seguente mostra i possibili scenari in cui un gestore di pacchetti richiede un pacchetto da un CodeCatalyst repository di pacchetti che dispone di repository upstream.

Per questo esempio, un gestore di pacchetti, ad esempio npm, richiede una versione del pacchetto da un archivio di pacchetti denominato downstream che ha più repository upstream. Quando viene richiesto il pacchetto, può verificarsi quanto segue:

- Se downstream contiene la versione del pacchetto richiesta, viene restituita al client.
- Se downstream non contiene la versione del pacchetto richiesta, la CodeCatalyst cerca nei downstream repository upstream, nell'ordine di ricerca configurato. Se viene trovata la versione del pacchetto, viene copiato un riferimento ad essa e la versione del pacchetto viene restituita al downstream client.

- Se nessuno dei downstream suoi repository upstream contiene la versione del pacchetto, al client viene restituita una Not Found risposta HTTP 404.

Il numero massimo di repository diretti upstream consentito per un repository è 10. Il numero massimo di CodeCatalyst ricerche nei repository quando viene richiesta una versione del pacchetto è 25.

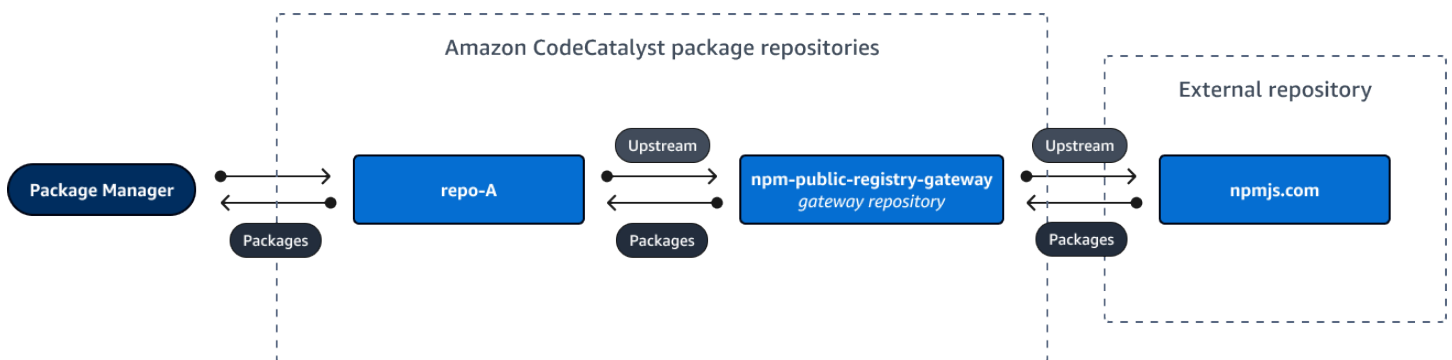
Package retention dai repository upstream

Se una versione del pacchetto richiesta viene trovata in un repository upstream, viene mantenuto un riferimento ad essa ed è sempre disponibile nell'archivio che l'ha richiesta. Ciò garantisce l'accesso ai pacchetti in caso di interruzione imprevista dell'archivio upstream. La versione del pacchetto mantenuta non è influenzata da nessuno dei seguenti fattori:

- Eliminazione del repository upstream.
- Disconnessione del repository upstream dal repository downstream.
- Eliminazione della versione del pacchetto dal repository upstream.
- Modifica della versione del pacchetto nell'archivio upstream (ad esempio, aggiungendovi una nuova risorsa).

Recupero dei pacchetti tramite una relazione a monte

CodeCatalyst può recuperare pacchetti tramite più repository collegati chiamati repository upstream. Se un repository di CodeCatalyst pacchetti ha una connessione upstream a un altro repository di CodeCatalyst pacchetti che ha una connessione upstream a un repository gateway, le richieste di pacchetti non presenti nell'archivio upstream vengono copiate dal repository esterno. Ad esempio, considerate la seguente configurazione: un repository denominato repo-A ha una connessione upstream al repository del gateway, npm-public-registry-gateway. npm-public-registry-gateway ha una connessione upstream al repository pubblico dei pacchetti, <https://npmjs.com>



Se npm è configurato per utilizzare il repo-A repository, l'esecuzione `npm install` avvia la copia dei pacchetti da dentro. <https://npmjs.com> npm-public-registry-gateway Vengono inoltre inserite le versioni installate. repo-A L'esempio seguente installa lodash.

```
$ npm config get registry
https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/
$ npm install lodash
+ lodash@4.17.20
added 1 package from 2 contributors in 6.933s
```

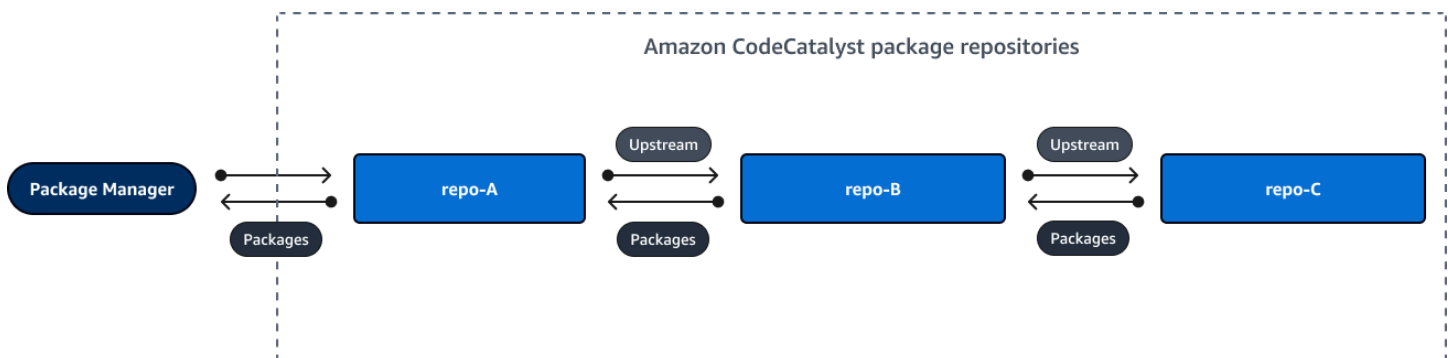
Dopo l'esecuzione `npm install`, repo-A contiene solo la versione più recente (lodash 4.17.20) perché è la versione che è stata recuperata da npm. repo-A

Perché npm-public-registry-gateway dispone di una connessione upstream esterna a <https://npmjs.com>, tutte le versioni del pacchetto da cui vengono importate <https://npmjs.com> vengono archiviate in. npm-public-registry-gateway Queste versioni del pacchetto avrebbero potuto essere recuperate da qualsiasi repository downstream con una connessione upstream che porta a. npm-public-registry-gateway

Il contenuto di npm-public-registry-gateway fornisce un modo per visualizzare tutti i pacchetti e le versioni dei pacchetti importati nel tempo. <https://npmjs.com>

Conservazione dei pacchetti in repository intermedi

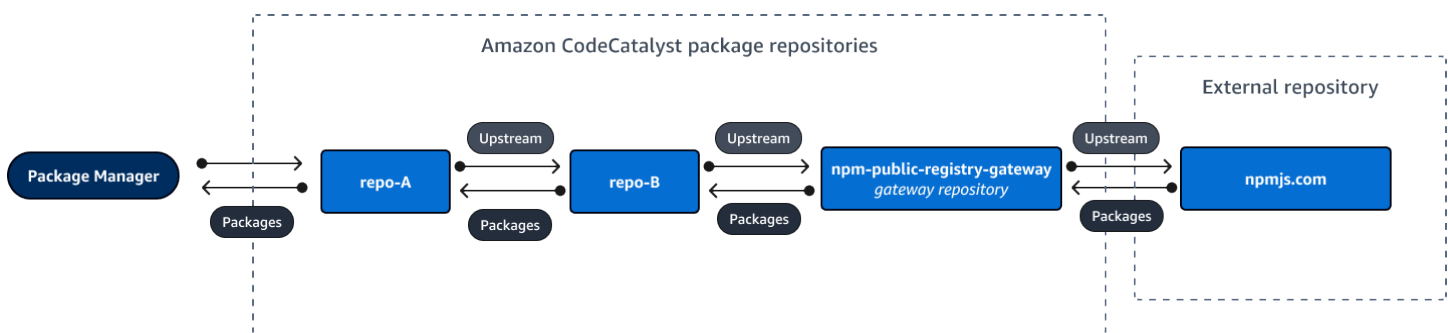
CodeCatalyst consente di concatenare repository upstream. Ad esempio, repo-A può avere repo-B come repository upstream e repo-B può avere repo-C come repository upstream. Questa configurazione rende le versioni del pacchetto in repo-B e disponibili da. repo-C repo-A



Quando un gestore di pacchetti si connette al repository repo-A e recupera una versione del pacchetto dal repository repo-C, la versione del pacchetto non viene conservata nell'archivio. repo-B La versione del pacchetto viene conservata solo nell'archivio a valle più lontano, che in

questo esempio è. **repo-A** Non viene conservata in nessun archivio intermedio. Questo vale anche per le catene più lunghe; ad esempio, se ci fossero quattro repository: **repo-A**, **repo-B**, **repo-C** e **repo-D**, e un gestore di pacchetti collegato al quale è stata recuperata una versione del pacchetto **repo-D**, la versione del pacchetto verrebbe conservata in **repo-A** ma non in **repo-B** o **repo-C**.

Il comportamento di conservazione dei pacchetti è simile quando si estrae una versione del pacchetto da un archivio pubblico di pacchetti, tranne per il fatto che la versione del pacchetto viene sempre conservata nell'archivio del gateway che ha la connessione diretta a monte all'archivio pubblico. Ad esempio, ha come repository upstream **repo-A**. **repo-B** ha **npm-public-registry-gateway** come repository upstream, che ha una connessione upstream al repository pubblico, **npmjs.com**; vedi lo schema seguente.



Se un gestore di pacchetti connesso a **repo-A** richiede una versione specifica del pacchetto, ad esempio **lodash 4.17.20**, e la versione del pacchetto non è presente in nessuno dei tre repository, verrà recuperata da **npmjs.com**. Quando **lodash 4.17.20** viene recuperato, viene mantenuto in **repo-A** quanto si tratta del repository a valle più lontano e poiché dispone della connessione a monte al repository esterno pubblico, **npmjs.com**. **npm-public-registry-gateway** **lodash 4.17.20** **repo-B** non viene mantenuto perché si tratta di un repository intermedio.

Rimozione di un repository upstream

Se non desideri più accedere ai pacchetti all'interno di un repository upstream, puoi rimuovere il repository upstream da un repository di pacchetti.

Warning

Quando rimuovi un repository upstream, potresti interrompere le catene di relazioni originarie, il che potrebbe interrompere i tuoi progetti o le tue build.

Per rimuovere un repository upstream

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Nella pagina Package repository, scegli il repository di pacchetti da cui desideri rimuovere un repository upstream.
3. Sotto il nome del repository dei pacchetti, scegli Upstreams.
4. Nella sezione Modifica repository upstream, trova il repository upstream che desideri rimuovere e scegli.



5. Quando hai finito di rimuovere i repository upstream, scegli Salva.

Connessione a archivi esterni pubblici

È possibile connettere gli archivi di CodeCatalyst pacchetti ai repository esterni pubblici supportati aggiungendo il repository gateway corrispondente come repository upstream. I repository gateway fungono da repository intermedi che cercano e archiviano i pacchetti estratti da repository esterni. Ciò consente di risparmiare tempo e trasferimento di dati perché tutti gli archivi di pacchetti di un progetto possono utilizzare i pacchetti archiviati dal repository del gateway.

Per connettersi a un archivio pubblico utilizzando i repository gateway

1. Nel riquadro di navigazione scegliere Pacchetti.
2. In Pacchetti, scegli la pagina dei repository Gateway. È possibile visualizzare un elenco di repository gateway supportati e le relative descrizioni.
3. Per utilizzare un repository gateway, è necessario innanzitutto crearlo. Se il repository gateway è stato creato, vengono visualizzate la data e l'ora di creazione. In caso contrario, scegli Crea per crearlo.
4. Per utilizzare i pacchetti del repository del gateway, è necessario impostare una connessione upstream ad esso da un CodeCatalyst repository. Scegli Package repository e scegli il repository di pacchetti a cui vuoi connetterti.
5. Per connetterti al repository pubblico, scegli Upstreams e seleziona Seleziona repository upstream.
6. Scegliete Gateway repository, selezionate il repository gateway che corrisponde al repository pubblico a cui desiderate connettervi come repository upstream.

7. Dopo aver selezionato tutti i repository gateway che desideri aggiungere come repository upstream, scegli **Seleziona**.
8. Quando hai finito di ordinare i repository upstream, scegli **Salva**.

Per ulteriori informazioni sui repository upstream, consulta [Configurazione e utilizzo degli archivi upstream](#)

Dopo aver aggiunto un repository gateway come repository upstream, puoi utilizzare un gestore di pacchetti collegato al tuo repository locale per recuperare i pacchetti dall'archivio di pacchetti pubblico ed esterno che gli corrisponde. Non è necessario aggiornare la configurazione del gestore dei pacchetti. I pacchetti utilizzati in questo modo vengono archiviati sia nel repository del gateway che nell'archivio dei pacchetti locale. Per ulteriori informazioni sulla richiesta delle versioni dei pacchetti da un repository upstream, consulta [Richiesta di una versione del pacchetto con repository upstream](#)

Archivi di pacchetti esterni supportati e relativi repository gateway

CodeCatalyst supporta l'aggiunta di una connessione upstream alle seguenti autorità ufficiali di gestione dei pacchetti con repository gateway.

Tipo di pacchetto del repository	Description	Nome del repository del gateway
npm	registro pubblico npm	npm-public-registr y-gateway
Python	Indice dei pacchetti Python	pypi-gateway
Maven	Maven Central	maven-central-gate way
Maven	Archivio Google Android	google-android-gat eway
Maven	CommonsWare	commonsware-gatewa y
Maven	Archivio di plugin Gradle	gradle-plugins-gat eway

Tipo di pacchetto del repository	Description	Nome del repository del gateway
NuGet	NuGet Galleria	nuget-gallery-gateway

Pubblicazione e modifica di pacchetti

Un pacchetto in CodeCatalyst è un insieme di software e metadati necessari per risolvere le dipendenze e installare il software. Per un elenco dei formati di pacchetto supportati in CodeCatalyst, vedere [Pubblica e condividi pacchetti software in CodeCatalyst](#). Questa sezione fornisce informazioni sulla pubblicazione, la visualizzazione, l'eliminazione dei pacchetti e l'aggiornamento dello stato di una versione del pacchetto.

Argomenti

- [Pubblicazione di pacchetti in un archivio di CodeCatalyst pacchetti](#)
- [Visualizzazione dei dettagli della versione del pacchetto](#)
- [Eliminazione di una versione del pacchetto](#)
- [Aggiornamento dello stato di una versione del pacchetto](#)
- [Modifica dei controlli di origine dei pacchetti](#)

Pubblicazione di pacchetti in un archivio di CodeCatalyst pacchetti

È possibile pubblicare versioni di qualsiasi tipo di pacchetto supportato in un archivio di CodeCatalyst pacchetti utilizzando gli strumenti di gestione dei pacchetti. I passaggi per pubblicare una versione del pacchetto sono i seguenti:

Per pubblicare una versione del pacchetto in un archivio di CodeCatalyst pacchetti

1. In caso contrario, [crea un archivio di pacchetti](#).
2. Connetti il gestore di pacchetti al repository dei pacchetti. Per istruzioni su come connettere il gestore di pacchetti npm a un repository di CodeCatalyst pacchetti, vedi [Configurazione e utilizzo di npm](#).
3. Usa il tuo gestore di pacchetti connesso per pubblicare le versioni dei pacchetti.

Indice

- [Pubblicazione e repository upstream](#)
- [Pacchetti privati e archivi pubblici](#)
- [Sovrascrivere le risorse del](#)

Pubblicazione e repository upstream

In CodeCatalyst, non è possibile pubblicare versioni di pacchetti presenti in repository upstream raggiungibili o in repository pubblici. Ad esempio, supponiamo di voler pubblicare un pacchetto npm in un repository di pacchetti e di essere connesso a npmjs.com tramite un repository myrepo gateway configurato come repository upstream. `lodash@1.0` myrepo Se `lodash@1.0` è presente nel repository upstream o in npmjs.com, rifiuta qualsiasi tentativo di pubblicazione su di esso emettendo un errore di conflitto 409. CodeCatalyst myrepo Ciò consente di evitare la pubblicazione accidentale di un pacchetto con lo stesso nome e la stessa versione di un pacchetto in un repository upstream, il che può causare un comportamento imprevisto.

È comunque possibile pubblicare versioni diverse del nome di pacchetto che esistono in un repository upstream. Ad esempio, se `lodash@1.0` è presente in un repository upstream ma non lo è, `lodash@1.1` è possibile pubblicarlo `lodash@1.1` nel repository downstream.

Pacchetti privati e archivi pubblici

CodeCatalyst non pubblica i pacchetti archiviati negli CodeCatalyst archivi in archivi pubblici, come npmjs.com o Maven Central. CodeCatalyst importa pacchetti da repository pubblici in un CodeCatalyst repository, ma non sposta i pacchetti nella direzione opposta. I pacchetti pubblicati CodeCatalyst nei repository rimangono privati e sono disponibili solo per il CodeCatalyst progetto a cui appartiene il repository.

Sovrascrivere le risorse del

Non è possibile ripubblicare una risorsa del pacchetto già esistente con contenuti diversi. Ad esempio, supponete di aver già pubblicato un pacchetto Maven con una risorsa JAR. `mypackage-1.0.jar` Puoi pubblicare nuovamente quella risorsa solo se il checksum delle risorse vecchie e nuove è identico. Per ripubblicare la stessa risorsa con nuovi contenuti, eliminate prima la versione del pacchetto. Il tentativo di ripubblicare lo stesso nome di risorsa con contenuti diversi genererà un errore di conflitto HTTP 409.

Per i formati di pacchetto che supportano più risorse (Python e Maven), puoi aggiungere nuove risorse con nomi diversi a una versione del pacchetto esistente in qualsiasi momento, supponendo che tu disponga delle autorizzazioni richieste. Poiché npm supporta NuGet solo una singola risorsa per versione del pacchetto, per modificare una versione del pacchetto pubblicata devi prima eliminarla.

Se provate a ripubblicare una risorsa già esistente (ad esempio `mypackage-1.0.jar`) e il contenuto della risorsa pubblicata e della nuova risorsa sono identici, l'operazione avrà successo perché l'operazione è idempotente.

Visualizzazione dei dettagli della versione del pacchetto

È possibile utilizzare la CodeCatalyst console per visualizzare i dettagli su una versione specifica del pacchetto.

Per visualizzare i dettagli della versione del pacchetto

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Nella pagina Package repository, scegli il repository che contiene la versione del pacchetto di cui desideri visualizzare i dettagli.
3. Cerca la versione del pacchetto nella tabella Pacchetti. È possibile utilizzare la barra di ricerca per filtrare i pacchetti in base al nome e al formato del pacchetto. Scegli il pacchetto dall'elenco.
4. Nella pagina dei dettagli del pacchetto, scegli Versioni, quindi scegli la versione che desideri visualizzare.

Eliminazione di una versione del pacchetto

È possibile eliminare una versione del pacchetto dalla pagina dei dettagli della versione del pacchetto nella CodeCatalyst console.

Per eliminare una versione del pacchetto

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Nella pagina Package repository, scegli il repository che contiene la versione del pacchetto che desideri eliminare.
3. Cerca e scegli il pacchetto dalla tabella.
4. Nella pagina dei dettagli del pacchetto, scegli Versioni e scegli la versione che desideri eliminare.

5. Nella pagina dei dettagli della versione del Package, scegli Azioni versione, quindi scegli Elimina.
6. Inserisci delete nel campo di testo e scegli Elimina.

Aggiornamento dello stato di una versione del pacchetto

Ogni versione del pacchetto in CodeCatalyst ha uno stato che descrive lo stato attuale e la disponibilità della versione del pacchetto. È possibile modificare lo stato della versione del pacchetto nella CodeCatalyst console. Per ulteriori informazioni sui possibili valori di stato delle versioni dei pacchetti e sui relativi significati, vedere [Stato della versione del pacchetto](#).

Per aggiornare lo stato di una versione del pacchetto

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Nella pagina Package repository, scegli il repository che contiene la versione del pacchetto di cui vuoi aggiornare lo stato.
3. Cerca e scegli il pacchetto dalla tabella.
4. Nella pagina dei dettagli del pacchetto, scegli Versioni, quindi scegli la versione che desideri visualizzare.
5. Nella pagina dei dettagli della versione del Package, scegli Azioni, quindi scegli Rimuovi dalla lista, Archivia o Elimina. Per informazioni sullo stato di ogni versione del pacchetto, consulta [Stato della versione del pacchetto](#).
6. Inserisci il testo di conferma nel campo di testo, quindi scegli Rimuovi dalla lista, Archivia o Elimina, a seconda dello stato a cui stai effettuando l'aggiornamento.

Stato della versione del pacchetto

Di seguito sono riportati i valori possibili per lo stato della versione del pacchetto. È possibile modificare lo stato della versione del pacchetto nella console. Per ulteriori informazioni, consulta [Aggiornamento dello stato di una versione del pacchetto](#).

- **Pubblicato:** la versione del pacchetto è stata pubblicata con successo e può essere richiesta da un gestore di pacchetti. La versione del pacchetto verrà inclusa negli elenchi delle versioni dei pacchetti restituiti ai gestori di pacchetti; ad esempio, nell'output `dinpm view <package-name> versions`. Tutte le risorse della versione del pacchetto sono disponibili nel repository.
- **Incompiuto:** l'ultimo tentativo di pubblicazione non è stato completato. Attualmente solo le versioni dei pacchetti Maven possono avere lo stato Incompiuto. Ciò può verificarsi quando il client carica

una o più risorse per una versione del pacchetto ma non pubblica un `maven-metadata.xml` file per il pacchetto che include quella versione.

- **Non in elenco:** gli asset della versione del pacchetto possono essere scaricati dal repository, ma la versione del pacchetto non è inclusa nell'elenco delle versioni restituite ai gestori di pacchetti. Ad esempio, per un pacchetto npm, l'output di `npm view <package-name> versions` non include la versione del pacchetto. Ciò significa che la logica di risoluzione delle dipendenze di npm non seleziona la versione del pacchetto perché la versione non compare nell'elenco delle versioni disponibili. Tuttavia, se la versione del pacchetto Unlisted è già referenziata in un `npm package-lock.json` file, può comunque essere scaricata e installata; ad esempio, durante l'esecuzione. `npm ci`
- **Archivate:** le risorse della versione del pacchetto non possono essere scaricate. La versione del pacchetto non verrà inclusa nell'elenco delle versioni restituite ai gestori di pacchetti. Poiché gli asset non sono disponibili, il consumo della versione del pacchetto da parte dei client è bloccato. Se la compilazione dell'applicazione dipende da una versione aggiornata a Archived, la compilazione avrà esito negativo, a meno che la versione del pacchetto non sia stata memorizzata nella cache locale. Non puoi utilizzare un gestore di pacchetti o uno strumento di compilazione per ripubblicare una versione del pacchetto archiviato perché è ancora presente nel repository. Tuttavia, puoi modificare lo stato della versione del pacchetto riportandolo a Non in elenco o Pubblicato nella console.
- **Eliminata:** la versione del pacchetto non viene visualizzata negli elenchi e le risorse non possono essere scaricate dal repository. La differenza fondamentale tra Disposed e Archived è che con lo stato Disposed, le risorse della versione del pacchetto vengono eliminate definitivamente da CodeCatalyst. Per questo motivo, non è possibile spostare una versione del pacchetto da Disposed a Archiviata, Non elencata o Pubblicata. La versione del pacchetto non può essere utilizzata perché le risorse sono state eliminate. Quando una versione del pacchetto è contrassegnata come Disposta, non vi viene addebitata alcuna fattura per l'archiviazione delle risorse del pacchetto.

Oltre agli stati nell'elenco precedente, è possibile eliminare anche una versione del pacchetto. Una volta eliminata, la versione del pacchetto non è più presente nell'archivio ed è possibile ripubblicare liberamente quella versione del pacchetto utilizzando un gestore di pacchetti o uno strumento di compilazione.

Nome del pacchetto, versione del pacchetto e normalizzazione del nome dell'asset

CodeCatalyst normalizza i nomi dei pacchetti, le versioni dei pacchetti e i nomi delle risorse prima di archivarli, il che significa che i nomi o le versioni CodeCatalyst possono essere diversi dal nome

o dalla versione forniti al momento della pubblicazione del pacchetto. Per ulteriori informazioni su come vengono normalizzati i nomi e le versioni CodeCatalyst per ogni tipo di pacchetto, consultate la seguente documentazione.

- [Normalizzazione dei nomi dei pacchetti in Python](#)
- [NuGet normalizzazione del nome del pacchetto, della versione e del nome dell'asset](#)

CodeCatalyst non esegue la normalizzazione su altri formati di pacchetti.

Modifica dei controlli di origine dei pacchetti

In Amazon CodeCatalyst, le versioni dei pacchetti possono essere aggiunte a un archivio di pacchetti pubblicandole direttamente, estraendole da un repository upstream o importandole da un archivio pubblico esterno tramite un gateway. Se consenti l'aggiunta di versioni di un pacchetto sia tramite pubblicazione diretta che tramite acquisizione da archivi pubblici, sei vulnerabile a un attacco di sostituzione delle dipendenze. Per ulteriori informazioni, consulta [Attacchi di sostituzione delle dipendenze](#). Per proteggerti da un attacco di sostituzione delle dipendenze, configura i controlli di origine dei pacchetti su un pacchetto in un repository per limitare il modo in cui le versioni di quel pacchetto possono essere aggiunte al repository.

Dovresti prendere in considerazione la possibilità di configurare i controlli di origine dei pacchetti per fare in modo che le nuove versioni di pacchetti diversi provengano sia da fonti interne, come la pubblicazione diretta, sia da fonti esterne, come gli archivi pubblici. Per impostazione predefinita, i controlli di origine dei pacchetti sono configurati in base al modo in cui la prima versione di un pacchetto viene aggiunta all'archivio.

Impostazioni di controllo dell'origine del pacchetto

Con i controlli sull'origine dei pacchetti, è possibile configurare il modo in cui le versioni dei pacchetti possono essere aggiunte a un repository. Gli elenchi seguenti includono le impostazioni e i valori disponibili per il controllo dell'origine dei pacchetti.

Pubblicare

Questa impostazione configura se le versioni dei pacchetti possono essere pubblicate direttamente nel repository utilizzando gestori di pacchetti o strumenti simili.

- **ALLOW:** le versioni dei Package possono essere pubblicate direttamente.
- **BLOCK:** Le versioni dei pacchetti non possono essere pubblicate direttamente.

A monte

Questa impostazione configura se le versioni dei pacchetti possono essere importate da archivi pubblici esterni o conservate dagli archivi upstream quando richiesto da un gestore di pacchetti.

- **CONSENTI**: qualsiasi versione del pacchetto può essere conservata da altri CodeCatalyst repository configurati come archivi upstream o importata da una fonte pubblica con una connessione esterna.
- **BLOCK**: le versioni dei pacchetti non possono essere conservate da altri CodeCatalyst repository configurati come repository upstream o importate da una fonte pubblica con una connessione esterna.

Impostazioni predefinite per il controllo dell'origine dei pacchetti

I controlli di origine del pacchetto predefiniti per un pacchetto si baseranno su come la prima versione di quel pacchetto viene aggiunta all'archivio dei pacchetti.

- Se la prima versione del pacchetto viene pubblicata direttamente da un gestore di pacchetti, le impostazioni saranno Publish: ALLOW e Upstream: BLOCK.
- Se la prima versione del pacchetto viene importata da una fonte pubblica, le impostazioni saranno Publish: BLOCK e Upstream: ALLOW.

Scenari comuni di controllo dell'accesso ai pacchetti

Questa sezione descrive alcuni scenari comuni di aggiunta di una versione del pacchetto a un repository di CodeCatalyst pacchetti. Le impostazioni di controllo dell'origine dei pacchetti vengono impostate per i nuovi pacchetti a seconda di come viene aggiunta la prima versione del pacchetto.

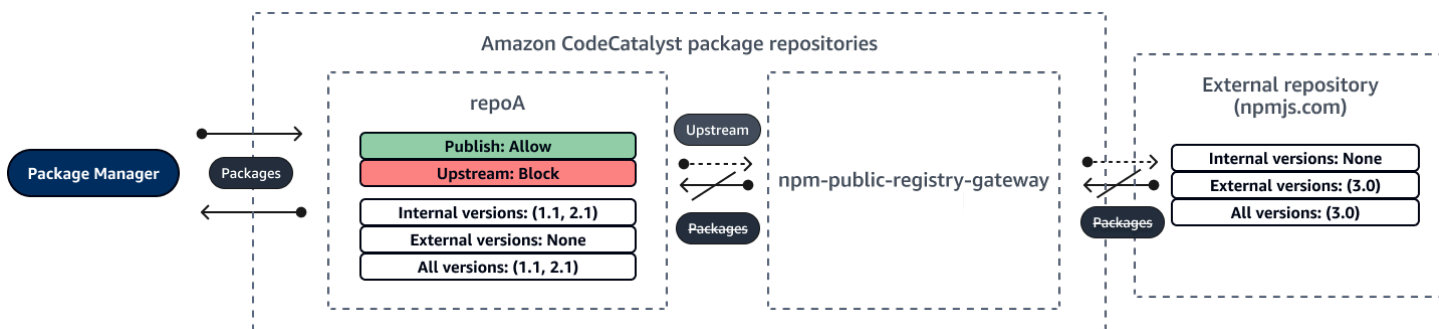
Nei seguenti scenari, un pacchetto interno viene pubblicato direttamente da un gestore di pacchetti nel repository, ad esempio un pacchetto gestito dall'utente. Un pacchetto esterno è un pacchetto esistente in un archivio pubblico che può essere inserito nel repository dell'utente tramite un repository gateway upstream.

Viene pubblicata una versione esterna del pacchetto per un pacchetto interno esistente

In questo scenario, si consideri un pacchetto interno, PackageA. Il tuo team pubblica la prima versione del pacchetto per PackageA in un archivio di pacchetti. CodeCatalyst Poiché questa è la prima versione del pacchetto, le impostazioni di controllo dell'origine del pacchetto vengono

impostate automaticamente su **Pubblica: Consenti** e **Upstream: Block**. Dopo la pubblicazione del pacchetto nel repository, un pacchetto con lo stesso nome viene pubblicato in un archivio pubblico collegato all'archivio dei pacchetti. CodeCatalyst Potrebbe trattarsi di un tentativo di attacco di sostituzione delle dipendenze contro il pacchetto interno o potrebbe essere una coincidenza. In ogni caso, i controlli di origine dei pacchetti sono configurati per bloccare l'ingestione della nuova versione esterna per proteggersi da un potenziale attacco.

Nell'immagine seguente, RePoA è il tuo repository di CodeCatalyst pacchetti con una connessione upstream al repository. npm-public-registry-gateway Il tuo repository contiene le versioni 1.1 e 2.1 di PackageA, ma la versione 3.0 è pubblicata nell'archivio pubblico. Normalmente, RePoA inserisce la versione 3.0 dopo che il pacchetto è stato richiesto da un gestore di pacchetti. Poiché l'ingestione dei pacchetti è impostata su **Block**, la versione 3.0 non viene inserita nell'archivio dei pacchetti e non è disponibile per i gestori di CodeCatalyst pacchetti ad esso collegati.

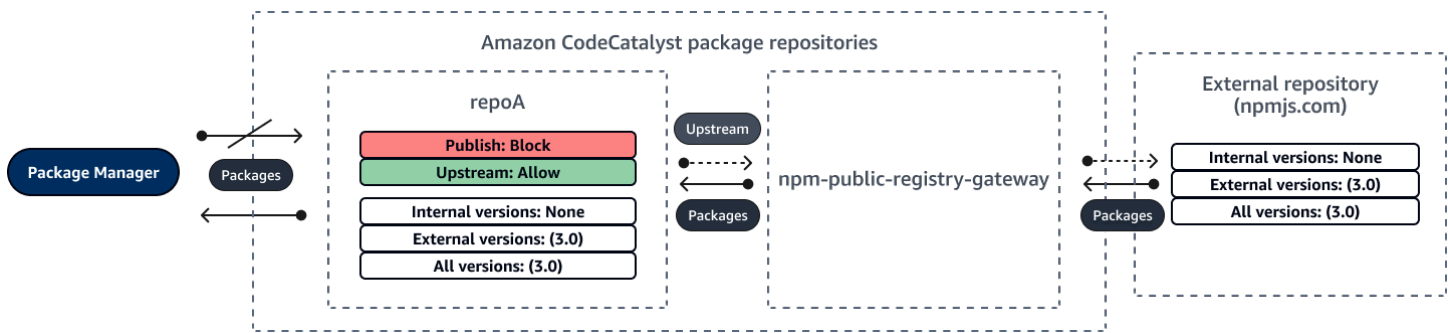


Viene pubblicata una versione interna del pacchetto per un pacchetto esterno esistente

In questo scenario, un pacchetto, PackageB, esiste esternamente in un repository pubblico collegato al repository. Quando un gestore di pacchetti connesso al tuo repository richiede PackageB, la versione del pacchetto viene inserita nel tuo repository dal repository pubblico. Poiché questa è la prima versione del pacchetto di PackageB aggiunta al repository, le impostazioni di origine del pacchetto sono configurate su **Publish: BLOCK** e **Upstream: ALLOW**. Successivamente, si tenta di pubblicare una versione con lo stesso nome di pacchetto nel repository. È possibile che non siate a conoscenza del pacchetto pubblico e stiate cercando di pubblicare un pacchetto non correlato con lo stesso nome, oppure state tentando di pubblicare una versione con patch o forse state tentando di pubblicare direttamente la versione esatta del pacchetto che già esiste esternamente. CodeCatalyst rifiuta la versione che state tentando di pubblicare, ma potete ignorare esplicitamente il rifiuto e pubblicare la versione, se necessario.

Nell'immagine seguente, RePoA è il tuo repository di CodeCatalyst pacchetti con una connessione upstream al repository. npm-public-registry-gateway L'archivio dei pacchetti contiene la versione 3.0 che ha acquisito dal repository pubblico. Vuoi pubblicare la versione 1.2 nel tuo

repository di pacchetti. In genere, è possibile pubblicare la versione 1.2 su RePoA, ma poiché la pubblicazione è impostata su Block, la versione 1.2 non può essere pubblicata.



Pubblicazione di una versione patchata di un pacchetto esterno esistente

In questo scenario, un pacchetto, PackageB, esiste esternamente in un archivio pubblico collegato all'archivio dei pacchetti. Quando un gestore di pacchetti collegato al tuo repository richiede PackageB, la versione del pacchetto viene inserita nel tuo repository dal repository pubblico. Poiché questa è la prima versione del pacchetto di PackageB aggiunta al repository, le impostazioni di origine del pacchetto sono configurate su Publish: BLOCK e Upstream: ALLOW. Il tuo team decide di pubblicare le versioni del pacchetto con patch di questo pacchetto nel repository. Per poter pubblicare direttamente le versioni dei pacchetti, il team modifica le impostazioni di controllo dell'origine del pacchetto in Publish: ALLOW e Upstream: BLOCK. Le versioni di questo pacchetto possono ora essere pubblicate direttamente nel repository e importate da archivi pubblici. Dopo che il team ha pubblicato le versioni del pacchetto con patch, ripristina le impostazioni di origine del pacchetto su Publish: BLOCK e Upstream: ALLOW.

Modifica dei controlli di origine dei pacchetti

I controlli di origine dei pacchetti vengono configurati automaticamente in base al modo in cui la prima versione di un pacchetto viene aggiunta all'archivio dei pacchetti. Per ulteriori informazioni, consulta [Impostazioni predefinite per il controllo dell'origine dei pacchetti](#). Per aggiungere o modificare i controlli di origine dei pacchetti per un pacchetto in un archivio dei CodeCatalyst pacchetti, effettuate i passaggi indicati nella procedura seguente.

Per aggiungere o modificare i controlli di origine del pacchetto

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Scegliete l'archivio del pacchetto che contiene il pacchetto che desiderate modificare.
3. Nella tabella Pacchetti, cercate e scegliete il pacchetto che desiderate modificare.
4. Dalla pagina di riepilogo del pacchetto, scegli Origin controls.

5. In Origin controls, scegli i controlli di origine del pacchetto che desideri impostare per questo pacchetto. Entrambe le impostazioni di controllo dell'origine del pacchetto, Publish e Upstream, devono essere impostate contemporaneamente.
 - Per consentire la pubblicazione diretta delle versioni dei pacchetti, in Pubblica, scegli Consenti. Per bloccare la pubblicazione delle versioni dei pacchetti, scegli Blocca.
 - Per consentire l'inserimento di pacchetti da repository esterni e l'estrazione di pacchetti da repository upstream, in Origini upstream, scegli Consenti. Per bloccare tutte le importazioni e l'estrazione di versioni di pacchetti da repository esterni e upstream, scegliete Blocca.
6. Scegli Save (Salva).

Archiviazione e archivi upstream

In CodeCatalyst, non è possibile pubblicare versioni di pacchetti presenti in repository upstream raggiungibili o in repository pubblici. Ad esempio, supponiamo di voler pubblicare un pacchetto npm in un repository e myrepo di disporre di un repository upstream con una myrepo connessione esterna `lodash@1.0` a `npmjs.com`. Considerate i seguenti scenari.

1. Le impostazioni di controllo dell'origine del pacchetto `lodash` sono Publish: ALLOW e Upstream: ALLOW. Se `lodash@1.0` è presente nel repository upstream o in `npmjs.com`, CodeCatalyst rifiuta qualsiasi tentativo di pubblicazione su di esso generando un errore di conflitto 409. myrepo È comunque possibile pubblicare una versione diversa, ad esempio. `lodash@1.1`
2. Le impostazioni di controllo dell'origine del pacchetto `lodash` sono Publish: ALLOW e Upstream: BLOCK. Puoi pubblicare nel tuo repository qualsiasi versione `lodash` di che non esista già perché le versioni del pacchetto non sono raggiungibili.
3. Le impostazioni di controllo dell'origine del pacchetto `lodash` sono Publish: BLOCK e Upstream: ALLOW. Non puoi pubblicare alcuna versione del pacchetto direttamente nel tuo repository.

Attacchi di sostituzione delle dipendenze

I Package Manager semplificano il processo di imballaggio e condivisione del codice riutilizzabile. Questi pacchetti possono essere pacchetti privati sviluppati da un'organizzazione per essere utilizzati nelle proprie applicazioni, oppure possono essere pacchetti pubblici, in genere pacchetti open source sviluppati all'esterno di un'organizzazione e distribuiti da archivi di pacchetti pubblici. Quando richiedono pacchetti, gli sviluppatori si affidano al proprio gestore di pacchetti per recuperare nuove versioni delle proprie dipendenze. Gli attacchi di sostituzione delle dipendenze, noti anche come

attacchi basati sulla confusione delle dipendenze, sfruttano il fatto che un gestore di pacchetti in genere non ha modo di distinguere le versioni legittime di un pacchetto da quelle dannose.

Gli attacchi di sostituzione delle dipendenze appartengono a un sottoinsieme di attacchi noti come attacchi alla catena di fornitura del software. Un attacco alla catena di fornitura del software è un attacco che sfrutta le vulnerabilità in qualsiasi punto della catena di fornitura del software.

Un attacco basato sulla sostituzione delle dipendenze può colpire chiunque utilizzi sia pacchetti sviluppati internamente sia pacchetti recuperati da archivi pubblici. Gli aggressori identificano i nomi interni dei pacchetti e quindi posizionano strategicamente il codice dannoso con lo stesso nome negli archivi pubblici dei pacchetti. In genere, il codice dannoso viene pubblicato in un pacchetto con un numero di versione elevato. I gestori di pacchetti recuperano il codice dannoso da questi feed pubblici perché ritengono che i pacchetti dannosi siano le versioni più recenti del pacchetto. Ciò causa una «confusione» o una «sostituzione» tra il pacchetto desiderato e il pacchetto dannoso, con conseguente compromissione del codice.

Per prevenire attacchi di sostituzione delle dipendenze, Amazon CodeCatalyst fornisce controlli sull'origine dei pacchetti. I controlli di origine dei pacchetti sono impostazioni che controllano il modo in cui i pacchetti possono essere aggiunti ai repository. I controlli vengono configurati automaticamente quando la prima versione del pacchetto di un nuovo pacchetto viene aggiunta a un CodeCatalyst repository. I controlli possono garantire che le versioni dei pacchetti non possano essere sia pubblicate direttamente nel repository sia importate da fonti pubbliche, proteggendo così l'utente dagli attacchi di sostituzione delle dipendenze. Per ulteriori informazioni sui controlli di origine dei pacchetti e su come modificarli, consulta [Modifica dei controlli di origine dei pacchetti](#)

Utilizzo di npm

Questi argomenti descrivono come è possibile utilizzare npm il gestore di pacchetti Node.js con CodeCatalyst.

Note

CodeCatalyst supporta node v4.9.1 e versioni successive npm v5.0.0 e successive.

Argomenti

- [Configurazione e utilizzo di npm](#)
- [gestione dei tag npm](#)

Configurazione e utilizzo di npm

Per npm utilizzarlo CodeCatalyst, è necessario connettersi npm al repository dei pacchetti e fornire un token di accesso personale (PAT) per l'autenticazione. È possibile visualizzare le istruzioni per la connessione npm al repository dei pacchetti nella console. CodeCatalyst

Indice

- [Configurazione di npm con CodeCatalyst](#)
- [Installazione dei pacchetti npm da un archivio di pacchetti CodeCatalyst](#)
- [Installazione dei pacchetti npm da npmjs tramite CodeCatalyst](#)
- [Pubblicazione di pacchetti npm nel tuo repository di pacchetti CodeCatalyst](#)
- [supporto ai comandi npm](#)
 - [Comandi supportati che interagiscono con un archivio di pacchetti](#)
 - [Comandi lato client supportati](#)
 - [Comandi non supportati](#)

Configurazione di npm con CodeCatalyst

Le seguenti istruzioni spiegano come autenticarsi e connettersi npm al repository dei pacchetti. CodeCatalyst Per ulteriori informazioni su npm, consulta la documentazione [ufficiale](#) di npm.

Per connetterti **npm** al tuo repository di pacchetti CodeCatalyst

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto.
3. Nel riquadro di navigazione scegliere Pacchetti.
4. Scegli il tuo repository di pacchetti dall'elenco.
5. Scegli Connect to repository.
6. Nei dettagli di configurazione, in Package manager client, scegli npm client.
7. Scegli il tuo sistema operativo per visualizzare i passaggi di configurazione corrispondenti.
8. È necessario un token di accesso personale (PAT) per autenticare npm con. CodeCatalyst Se hai già un token, puoi usarlo. In caso contrario, puoi crearne uno utilizzando i seguenti passaggi.
 - a. (Facoltativo): aggiorna il nome e la data di scadenza del PAT.

- b. Scegli Crea token.
- c. Copia e archivia il tuo PAT in un luogo sicuro.

 Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo. Le credenziali devono essere di breve durata per ridurre al minimo il periodo di tempo in cui un utente malintenzionato può utilizzarle dopo averle sottratte indebitamente.

9. Esegui i seguenti comandi dalla directory principale del tuo progetto per configurare npm con il tuo repository di pacchetti. I comandi faranno quanto segue:
 - Crea un `.npmrc` file a livello di progetto se il tuo progetto non ne ha uno.
 - Aggiungi le informazioni sull'endpoint del repository del pacchetto al file a livello di progetto.
`.npmrc`
 - Aggiungi le tue credenziali (PAT) al tuo file a livello utente. `.npmrc`

Sostituisci i seguenti valori.

 Note

Se state copiando le istruzioni dalla console, i valori dei seguenti comandi vengono aggiornati automaticamente e non è necessario modificarli.

- *username* Sostituiscilo con il tuo nome CodeCatalyst utente.
- *PAT* Sostituiscilo con il tuo CodeCatalyst PAT.
- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
npm set registry=https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/ --location project
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:_authToken=username:PAT
```

Per npm 6 o versioni precedenti: per fare in modo che npm passi sempre il token di autenticazione a CodeCatalyst, anche per GET le richieste, imposta la variabile di configurazione `always-auth` come segue. `npm config set`

```
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:always-auth=true --location project
```

Installazione dei pacchetti npm da un archivio di pacchetti CodeCatalyst

Dopo aver collegato npm al tuo repository seguendo i passaggi indicati [Configurazione di npm con CodeCatalyst](#), puoi eseguire npm comandi sul tuo repository.

Puoi installare un pacchetto npm che si trova nel tuo repository di CodeCatalyst pacchetti o in uno dei suoi repository upstream con il comando. `npm install`

```
npm install lodash
```

Installazione dei pacchetti npm da npmjs tramite CodeCatalyst

È possibile installare i pacchetti npm da npmjs.com tramite un CodeCatalyst repository configurando il repository con una connessione upstream al repository gateway collegato a npmjs.com,. `npm-public-registry-gateway` I pacchetti installati da npmjs vengono inseriti e archiviati nel repository del gateway e nel repository di pacchetti più lontano a valle.

Per installare pacchetti da npmjs

1. Se non l'hai già fatto, esegui la configurazione npm con il tuo repository di CodeCatalyst pacchetti seguendo i passaggi riportati di seguito. [Configurazione di npm con CodeCatalyst](#)
2. Verifica che il tuo repository abbia aggiunto il repository gateway come connessione `npm-public-registry-gatewayupstream`. Puoi verificare quali sorgenti upstream vengono aggiunte o aggiungerle `npm-public-registry-gateway` come sorgente upstream seguendo le istruzioni contenute [Aggiungere un repository upstream](#) e scegliendo il repository. `npm-public-registry-gateway`
3. Installa i pacchetti con il comando. `npm install`

```
npm install package_name
```

Per ulteriori informazioni sulla richiesta di pacchetti dai repository upstream, vedere [Richiesta di una versione del pacchetto con repository upstream](#)

Pubblicazione di pacchetti npm nel tuo repository di pacchetti CodeCatalyst

Dopo aver completato [Configurazione di npm con CodeCatalyst](#), puoi eseguire npm i comandi.

È possibile pubblicare un pacchetto npm in un repository di CodeCatalyst pacchetti con il npm publish comando.

```
npm publish
```

Per informazioni su come creare pacchetti npm, vedere [Creazione di moduli Node.js](#) su npm Docs.

supporto ai comandi npm

Le sezioni seguenti riassumono i npm comandi supportati dai repository di CodeCatalyst pacchetti, oltre a elencare comandi specifici che non sono supportati.

Argomenti

- [Comandi supportati che interagiscono con un archivio di pacchetti](#)
- [Comandi lato client supportati](#)
- [Comandi non supportati](#)

Comandi supportati che interagiscono con un archivio di pacchetti

Questa sezione elenca npm i comandi in cui il npm client effettua una o più richieste al registro in cui è configurato (ad esempio, `npm config set registry`). È stato verificato che questi comandi funzionino correttamente quando vengono richiamati CodeCatalyst su un archivio di pacchetti.

Comando	Description
bug	Indovina la posizione dell'URL del bug tracker di un pacchetto, quindi tenta di aprirlo.
ci	Installa un progetto con una tabula rasa.
deprecare	Depreca una versione di un pacchetto.

Comando	Description
dist-tag	Modifica i tag di distribuzione dei pacchetti.
documenti	Indovina la posizione dell'URL della documentazione di un pacchetto, quindi tenta di aprirlo utilizzando il <code>--browser</code> parametro config.
dottore	Esegue una serie di controlli per verificare che l'installazione di npm sia in grado di gestire i JavaScript pacchetti.
installa	Installa un pacchetto.
install-ci-test	Installa un progetto con una lavagna pulita ed esegue dei test. Pseudonimo: <code>npm ci</code> . Questo comando esegue <code>unnpm ci</code> , seguito immediatamente da <code>unnpm test</code> .
install-test	Installa il pacchetto ed esegue i test. Esegue <code>unnpm install</code> , seguito immediatamente da <code>unnpm test</code> .
obsoleto	Controlla il registro configurato per determinare se i pacchetti installati sono obsoleti.
ping	Esegue il ping del registro npm configurato o specificato e verifica l'autenticazione.
pubblica	Pubblica una versione del pacchetto nel registro.
update	Indovina la posizione dell'URL del repository di un pacchetto, quindi tenta di aprirlo utilizzando il parametro config. <code>--browser</code>
visualizzare	Visualizza i metadati del pacchetto. Può essere utilizzato anche per stampare le proprietà dei metadati.

Comandi lato client supportati

Questi comandi non richiedono alcuna interazione diretta con un repository di pacchetti, quindi CodeCatalyst non richiede nulla per supportarli.

Comando	Description
bin (legacy)	Visualizza la bin directory npm.
costruire	Costruisce un pacchetto.
cache	Manipola la cache dei pacchetti.
completamento	Abilita il completamento delle schede in tutti i comandi npm.
config	Aggiorna il contenuto dell'utente e dei npmrc file globali.
deduplicare	Cerca nell'albero dei pacchetti locale e tenta di semplificare la struttura spostando le dipendenze più in alto nell'albero, dove possono essere condivise più efficacemente da più pacchetti dipendenti.
modifica	Modifica un pacchetto installato. Seleziona una dipendenza nella directory di lavoro corrente e apre la directory dei pacchetti nell'editor predefinito.
esplora	Sfoglia un pacchetto installato. Genera una subshell nella directory del pacchetto installato specificato. Se viene specificato un comando, questo viene eseguito nella sottoshell, che si spegne immediatamente.
help	Ottiene aiuto su npm.
aiuto-ricerca	Cerca nella documentazione di aiuto di npm.

Comando	Description
init	Crea un <code>package.json</code> file.
collegamento	Collega simbolicamente una cartella di pacchetti.
ls	Elenca i pacchetti installati.
pacchetto	Crea un archivio tar da un pacchetto.
prefisso	Visualizza un prefisso. Questa è la directory principale più vicina a contenere un <code>package.json</code> file, a meno che non <code>-g</code> sia specificata anche questa.
prugna	Rimuove i pacchetti che non sono elencati nell'elenco delle dipendenze del pacchetto principale.
ricostruire	Esegue il <code>npm build</code> comando sulle cartelle corrispondenti.
riavviare	Esegue gli script di arresto, riavvio e avvio di un pacchetto e i pre-script e post-script associati.
root	Stampa la directory effettiva per renderla <code>standardnode_modules</code> .
esegui script	Esegue script di pacchetti arbitrari.
pellicola shrinkwrap	Blocca le versioni dipendenti per la pubblicazione.
disinstallare	Disinstalla un pacchetto.

Comandi non supportati

Questi npm comandi non sono supportati dagli archivi dei CodeCatalyst pacchetti.

Comando	Description	Note
accesso	Imposta il livello di accesso sui pacchetti pubblicati.	CodeCatalyst utilizza un modello di autorizzazione diverso dal repository pubblico npmjs.
aggiungi utente	Aggiunge un account utente del registro	CodeCatalyst utilizza un modello utente diverso dal repository pubblico npmjs.
audit	Esegue un controllo di sicurezza.	CodeCatalyst attualmente non vende dati sulle vulnerabilità di sicurezza.
gancio	Gestisce gli hook di npm, inclusi l'aggiunta, la rimozione, l'elenco e l'aggiornamento.	CodeCatalyst attualmente non supporta alcun meccanismo di notifica delle modifiche.
Login	Autentica un utente. Questo è un alias per <code>npm adduser</code> .	CodeCatalyst utilizza un modello di autenticazione diverso dal repository pubblico npmjs. Per informazioni, consulta Configurazione di npm con CodeCatalyst .
Disconnessione	Si disconnette dal registro.	CodeCatalyst utilizza un modello di autenticazione diverso dal repository pubblico npmjs. Non è possibile disconnettersi da un CodeCatalyst repository, ma i token di autenticazione scadono dopo la data di scadenza configurabile. La durata predefinita del token è di 12 ore.

Comando	Description	Note
proprietario	Gestisce i proprietari dei pacchetti.	CodeCatalyst utilizza un modello di autorizzazioni diverso dal repository pubblico npmjs.
profile	Modifica le impostazioni sul profilo del registro.	CodeCatalyst utilizza un modello utente diverso dal repository pubblico npmjs.
cerca	Cerca nel registro i pacchetti che corrispondono ai termini di ricerca.	CodeCatalyst non supporta il search comando.
stella	Contrassegna i tuoi pacchetti preferiti.	CodeCatalyst attualmente non supporta alcun meccanismo di creazione di preferiti.
stelle	Visualizza i pacchetti contrassegnati come preferiti.	CodeCatalyst attualmente non supporta alcun meccanismo di creazione di preferiti.
squadra	Gestisce i team e le appartenenze ai team.	CodeCatalyst utilizza un modello di appartenenza di utenti e gruppi diverso dal repository pubblico npmjs.
token	Gestisce i token di autenticazione.	CodeCatalyst utilizza un modello diverso per ottenere i token di autenticazione. Per informazioni, consulta Configurazione di npm con CodeCatalyst .

Comando	Description	Note
annullare la pubblicazione	Rimuove un pacchetto dal registro.	CodeCatalyst non supporta la rimozione di una versione del pacchetto da un repository utilizzando il client npm. È possibile eliminare un pacchetto nella console.
whoami	Visualizza il nome utente npm.	CodeCatalyst utilizza un modello utente diverso dal repository pubblico npmjs.

gestione dei tag npm

I registri npm supportano i tag, che sono alias di stringa per le versioni dei pacchetti. È possibile utilizzare i tag per fornire un alias invece di utilizzare i numeri di versione. Ad esempio, hai un progetto con più flussi di sviluppo e utilizzi un tag diverso per ogni flusso (ad esempio, `stable`, `betadev`, `canary`). Per ulteriori informazioni, consulta [dist-tag](#) su npm Docs.

Per impostazione predefinita, npm utilizza il `latest` tag per identificare la versione corrente di un pacchetto. `npm install pkg` (senza `@version` o `@tag` specificatore) installa il tag più recente. In genere, i progetti utilizzano solo il tag più recente per le versioni di rilascio stabili. Altri tag vengono utilizzati per le versioni instabili o non definitive.

Modifica dei tag con il client npm

I tre npm `dist-tag` comandi (`add`, `ls`) funzionano allo stesso modo nei repository di CodeCatalyst pacchetti come funzionano nel registro [npm predefinito](#).

tag npm e repository upstream

Quando npm richiede, i tag per un pacchetto e le versioni di quel pacchetto sono presenti anche in un repository upstream, CodeCatalyst unisce i tag prima di restituirli al client. Ad esempio, un repository denominato R ha un repository upstream denominato U. La tabella seguente mostra i tag per un pacchetto denominato presente in `web-helper` entrambi i repository.

Repository	Nome pacchetto	Tag del pacchetto
R	web-helper	più recente (alias per la versione 1.0.0)
U	web-helper	alpha (alias per la versione 1.0.1)

In questo caso, quando il client npm recupera i tag per il `web-helper` pacchetto dal repository R, riceve sia i tag più recenti che quelli alpha. Le versioni a cui puntano i tag non cambieranno.

Quando lo stesso tag è presente sullo stesso pacchetto sia nell'archivio originale che in quello locale, CodeCatalyst utilizza il tag che è stato aggiornato l'ultima volta. Ad esempio, supponiamo che i tag su webhelper siano stati modificati in modo da assomigliare ai seguenti.

Repository	Nome pacchetto	Tag del pacchetto	Ultimo aggiornamento
R	web-helper	più recente (alias per la versione 1.0.0)	1 gennaio 2023
U	web-helper	più recente (alias per la versione 1.0.1)	1 giugno 2023

In questo caso, quando il client npm recupera i tag per il pacchetto `web-helper` dal repository R, l'ultimo tag assumerà l'alias della versione 1.0.1 perché è stato aggiornato per ultima. Ciò semplifica l'utilizzo di nuove versioni di pacchetti in un repository upstream che non sono ancora presenti in un repository locale mediante l'esecuzione di `npm update`.

Usare Maven

Il formato di repository Maven è utilizzato da molti linguaggi diversi, tra cui Java, Kotlin, Scala e Clojure. È supportato da molti strumenti di compilazione diversi, tra cui Maven, Gradle, Scala SBT, Apache Ivy e Leiningen.

Abbiamo testato e confermato la compatibilità con le seguenti versioni: CodeCatalyst

- Ultima versione di Maven: 3.6.3.

- Ultima versione di Gradle: 6.4.1. Anche la versione 5.5.1 è stata testata.

Argomenti

- [Configurazione e utilizzo di Gradle Groovy](#)
- [Configurazione e utilizzo di mvn](#)
- [Pacchetti di pubblicazione con curl](#)
- [Utilizzo dei checksum e delle istantanee di Maven](#)

Configurazione e utilizzo di Gradle Groovy

Per utilizzare Gradle Groovy con CodeCatalyst, devi connettere Gradle Groovy al tuo repository di pacchetti e fornire un token di accesso personale (PAT) per l'autenticazione. Puoi visualizzare le istruzioni per connettere Gradle Groovy al tuo repository di pacchetti nella console. CodeCatalyst

Indice

- [Recupero delle dipendenze da CodeCatalyst](#)
- [Recupero di plugin da CodeCatalyst](#)
- [Recupero di pacchetti da archivi di pacchetti esterni tramite CodeCatalyst](#)
- [Pubblicazione di pacchetti su CodeCatalyst](#)
- [Esecuzione di una build Gradle in IntelliJ IDEA](#)
 - [Metodo 1: inserisci il PAT gradle.properties](#)
 - [Metodo 2: inserisci il PAT in un file separato](#)

Recupero delle dipendenze da CodeCatalyst

Le seguenti istruzioni spiegano come configurare Gradle Groovy per recuperare le dipendenze dal repository dei pacchetti. CodeCatalyst

Per usare Gradle Groovy per recuperare le dipendenze dal tuo repository di pacchetti CodeCatalyst

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Vai al tuo progetto.
3. Nel riquadro di navigazione scegliere Pacchetti.
4. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.

5. Scegli Connect to repository.
6. Nella finestra di dialogo Connect to repository, scegli Gradle Groovy dall'elenco dei client del gestore di pacchetti.
7. Avrai bisogno di un token di accesso personale (PAT) con cui autenticare Gradle Groovy. CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.

 Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

8. Aggiorna il file delle proprietà di gradle con le tue credenziali di accesso. *username* Sostituiscilo con il tuo CodeCatalyst nome utente e sostituiscilo *PAT* con il tuo token di accesso CodeCatalyst personale. È possibile utilizzare qualsiasi valore per *spaceUsername* e *spacePassword* purché si utilizzino gli stessi valori nei passaggi seguenti.

```
spaceUsername=username  
spacePassword=PAT
```

9. Per recuperare le dipendenze da CodeCatalyst una build di Gradle, copia lo snippet di maven codice e aggiungilo alla *repositories* sezione del file del progetto. *build.gradle* Sostituisci i seguenti valori. È possibile utilizzare qualsiasi valore *spaceName* purché si utilizzino gli stessi valori nei passaggi seguenti.

 Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
maven {
```

```
name = 'spaceName'  
url = uri('https://packages.region.codecatalyst.aws/  
maven/space_name/proj_name/repo_name/')  
credentials(PasswordCredentials)  
}
```

10. (Facoltativo) Per utilizzare l'archivio dei CodeCatalyst pacchetti come unica fonte per le dipendenze del progetto, rimuovete tutte le altre sezioni dei repository dal file. `build.gradle`. Se avete più di un repository, Gradle cerca in ogni repository le dipendenze nell'ordine in cui sono elencate.

Recupero di plugin da CodeCatalyst

[Per impostazione predefinita, Gradle risolverà i plugin dal Gradle Plugin Portal pubblico.](#) I

seguenti passaggi configurano il progetto Gradle per risolvere i plugin dal repository dei pacchetti CodeCatalyst

Per usare Gradle per recuperare i plugin dal tuo repository di pacchetti CodeCatalyst

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/) .
2. Vai al tuo progetto.
3. Nel riquadro di navigazione scegliere Pacchetti.
4. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
5. Scegli Connect to repository.
6. Nella finestra di dialogo Connect to repository, scegli Gradle dall'elenco dei client del gestore di pacchetti.
7. Avrai bisogno di un token di accesso personale (PAT) con cui autenticare Gradle. CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.



Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

8. Aggiorna il file delle proprietà di gradle con le tue credenziali di accesso. `username` Sostituiscilo con il tuo CodeCatalyst nome utente e sostituiscilo `PAT` con il tuo token di accesso CodeCatalyst

personale. È possibile utilizzare qualsiasi valore per *spaceUsername* e *spacePassword* purché si utilizzino gli stessi valori nei passaggi seguenti.

```
spaceUsername=username  
spacePassword=PAT
```

9. Aggiungi un pluginManagement blocco al tuo settings.gradle file. Il pluginManagement blocco deve apparire prima di qualsiasi altra istruzione in settings.gradle. Sostituisci i seguenti valori.

 Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- Sostituire *spaceName* con il valore del nome utilizzato nel passaggio precedente.
- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
pluginManagement {  
    repositories {  
        maven {  
            name = 'spaceName'  
            url = uri('https://packages.region.codecatalyst.aws/  
maven/space_name/proj_name/repo_name/')  
            credentials(PasswordCredentials)  
        }  
    }  
}
```

Ciò assicurerà che Gradle risolva i plugin dal repository specificato. Il repository deve avere una connessione upstream configurata al Gradle Plugin Portal (gradle-plugins-store) in modo che i plugin Gradle comunemente richiesti siano disponibili per la build. [Per ulteriori informazioni, consultate la documentazione di Gradle.](#)

Recupero di pacchetti da archivi di pacchetti esterni tramite CodeCatalyst

È possibile installare i pacchetti Maven dai repository pubblici tramite un CodeCatalyst repository configurandolo con una connessione upstream al gateway che rappresenta il repository del gateway. I pacchetti installati dal repository del gateway vengono inseriti e archiviati nel repository dell'utente. CodeCatalyst

CodeCatalyst supporta i seguenti archivi pubblici di pacchetti Maven.

- maven-central-gateway
- google-android-gateway
- gradle-plugins-gateway
- commonsware-gateway

Per installare pacchetti dagli archivi pubblici di pacchetti Maven

1. Se non l'hai già fatto, configura Gradle con il tuo repository di CodeCatalyst pacchetti seguendo i passaggi in o. [Recupero delle dipendenze da CodeCatalyst](#) [Recupero di plugin da CodeCatalyst](#)
2. Assicurati che il tuo repository abbia aggiunto il repository gateway da cui desideri installare come connessione upstream. Puoi farlo seguendo le istruzioni riportate [Aggiungere un repository upstream](#) e scegliendo il repository pubblico di pacchetti che desideri aggiungere come upstream.

Per ulteriori informazioni sulla richiesta di pacchetti dagli archivi upstream, consulta. [Richiesta di una versione del pacchetto con repository upstream](#)

Pubblicazione di pacchetti su CodeCatalyst

Questa sezione descrive come pubblicare una libreria Java creata con Gradle Groovy in un repository. CodeCatalyst

Per usare Gradle Groovy per pubblicare pacchetti in un repository di pacchetti CodeCatalyst

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/) CodeCatalyst .
2. Nella pagina di panoramica del progetto, scegli Pacchetti.
3. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
4. Scegli Connect to repository.

5. Nella finestra di dialogo Connect to repository, scegli Gradle Groovy dall'elenco dei client del gestore di pacchetti.
6. Avrai bisogno di un token di accesso personale (PAT) con cui autenticare Gradle. CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.

 Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

7. Aggiorna il file delle proprietà di gradle con le tue credenziali di accesso. *username* Sostituiscilo con il tuo CodeCatalyst nome utente e sostituiscilo *PAT* con il tuo token di accesso CodeCatalyst personale. È possibile utilizzare qualsiasi valore per *spaceUsername* e *spacePassword* purché si utilizzino gli stessi valori nei passaggi seguenti.

```
spaceUsername=username  
spacePassword=PAT
```

8. Aggiungi il maven-publish plugin alla plugins sezione del build.gradle file del progetto.

```
plugins {  
    id 'java-library'  
    id 'maven-publish'  
}
```

9. Quindi, aggiungi una publishing sezione al build.gradle file di progetto. Sostituire i seguenti valori.

 Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.

- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
publishing {
    publications {
        mavenJava(MavenPublication) {
            groupId = 'group-id'
            artifactId = 'artifact-id'
            version = 'version'
            from components.java
        }
    }
    repositories {
        maven {
            name = 'spaceName'
            url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
            credentials(PasswordCredentials)
        }
    }
}
```

Il maven-publish plugin genera un file POM basato su groupIdartifactId, e version specificato nella publishing sezione.

10. Una volta build.gradle completate queste modifiche, esegui il comando seguente per creare il progetto e caricarlo nel repository.

```
./gradlew publish
```

11. Accedi all'archivio dei pacchetti nella CodeCatalyst console per verificare che il pacchetto sia stato pubblicato correttamente. Dovresti vedere il pacchetto nell'elenco dei pacchetti del tuo archivio dei pacchetti.

Per ulteriori informazioni, consulta questi argomenti sul sito Web di Gradle:

- [Creazione di librerie Java](#)
- [Pubblicazione di un progetto come modulo](#)

Esecuzione di una build Gradle in IntelliJ IDEA

È possibile eseguire una build Gradle in IntelliJ IDEA da cui estrae le dipendenze. CodeCatalyst Per autenticare Gradle CodeCatalyst, è necessario utilizzare un token di accesso personale (PAT). Puoi archiviare il tuo CodeCatalyst PAT in un file separato `gradle.properties` o in un file separato a tua scelta.

Metodo 1: inserisci il PAT **gradle.properties**

Usa questo metodo se non stai usando il `gradle.properties` file e puoi sovrascriverne il contenuto con il tuo PAT. Se si utilizza `gradle.properties`, è possibile modificare questo metodo per aggiungere il PAT anziché sovrascrivere il contenuto del file.

Note

L'esempio mostra il `gradle.properties` file che si trova in. `GRADLE_USER_HOME`

Innanzitutto, create un PAT se non ne avete uno.

Per creare un token di accesso personale (PAT)

1. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri relativa a un progetto o a uno spazio e scegliendo il tuo nome dall'elenco dei membri.

2. Nel nome PAT, inserisci un nome descrittivo per il tuo PAT.
3. In Data di scadenza, lascia la data predefinita o scegli l'icona del calendario per selezionare una data personalizzata. La data di scadenza predefinita è un anno dalla data corrente.
4. Scegli Create (Crea).

Puoi anche creare questo token quando scegli Clone repository come repository di origine.

5. Salva il segreto PAT in un luogo sicuro.

 Important

Il segreto PAT viene visualizzato solo una volta. Non è possibile recuperarlo dopo aver chiuso la finestra.

Successivamente, aggiorna il `build.gradle` file con il seguente frammento:

```
repositories {  
    maven {  
        name = 'spaceName'  
        url = uri('https://packages.region.codecatalyst.aws/  
maven/space_name/proj_name/repo_name/')  
        credentials(PasswordCredentials)  
    }  
}
```

Metodo 2: inserisci il PAT in un file separato

Usa questo metodo se non vuoi modificare il tuo `gradle.properties` file.

Innanzitutto, crea un PAT se non ne hai uno.

Per creare un token di accesso personale (PAT)

1. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni.

 Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri relativa a un progetto o a uno spazio e scegliendo il tuo nome dall'elenco dei membri.

2. Nel nome PAT, inserisci un nome descrittivo per il tuo PAT.
3. In Data di scadenza, lascia la data predefinita o scegli l'icona del calendario per selezionare una data personalizzata. La data di scadenza predefinita è un anno dalla data corrente.
4. Scegli Create (Crea).

Puoi anche creare questo token quando scegli Clone repository come repository di origine.

5. Salva il segreto PAT in un luogo sicuro.

 Important

Il segreto PAT viene visualizzato solo una volta. Non è possibile recuperarlo dopo aver chiuso la finestra.

Per inserire il file PAT in un file separato

1. Aggiorna il `build.gradle` file con il seguente frammento. Sostituisci `repo_name` con il tuo nome CodeCatalyst utente `space_name`, `proj_name` il nome dello spazio, il nome del progetto e il nome del repository del pacchetto.

```
def props = new Properties()
file("fileName").withInputStream { props.load(it) }

repositories {
    maven {
        name = 'spaceName'
        url = uri('https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/')
        credentials(PasswordCredentials)
    }
}
```

2. Scrivi il tuo PAT nel file che è stato specificato nel tuo `build.gradle` file:

```
echo "codecatalystArtifactsToken=PAT" > fileName
```

Configurazione e utilizzo di mvn

Si utilizza il `mvn` comando per eseguire le build di Maven. È necessario configurare `mvn` per utilizzare l'archivio dei pacchetti e fornire un token di accesso personale (PAT) per l'autenticazione.

Indice

- [Recupero delle dipendenze da CodeCatalyst](#)
- [Recupero di pacchetti da archivi di pacchetti esterni tramite CodeCatalyst](#)
- [Pubblicazione di pacchetti su CodeCatalyst](#)

- [Pubblicazione di pacchetti di terze parti](#)

Recupero delle dipendenze da CodeCatalyst

mvnPer configurare il recupero delle dipendenze da un CodeCatalyst repository, devi modificare il file di configurazione di Maven `settings.xml` e, facoltativamente, il file Project Model Object (POM) del tuo progetto. Il file POM contiene informazioni sul progetto e informazioni di configurazione per consentire a Maven di creare il progetto, come dipendenze, directory di compilazione, directory dei sorgenti, directory dei sorgenti di test, plug-in e obiettivi.

Da usare per recuperare le **mvn** dipendenze dal repository dei pacchetti CodeCatalyst

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella pagina di panoramica del progetto, scegli Pacchetti.
3. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
4. Scegli Connect to repository.
5. Nella finestra di dialogo Connect to repository, scegliete mvn dall'elenco dei client del gestore di pacchetti.
6. Avrai bisogno di un token di accesso personale (PAT) con cui autenticarti. mvn CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.



Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

7. Aggiungi un profilo contenente il tuo repository al tuo `settings.xml` file. Sostituisci i seguenti valori.



Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
<profiles>
  <profile>
    <id>repo_name</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <repositories>
      <repository>
        <id>repo_name</id>
        <url>https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name/</url>
      </repository>
    </repositories>
  </profile>
</profiles>
```

8. Aggiungi il tuo server all'elenco dei server nel `settings.xml` file. Sostituisci i seguenti valori.

 Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.
- Sostituisci *username* con il tuo nome CodeCatalyst utente.
- *PAT* Sostituiscilo con il tuo CodeCatalyst PAT.

```
<servers>
  <server>
    <id>repo_name</id>
    <username>username</username>
    <password>PAT</password>
```



```
</server>  
</servers>
```

9. (Facoltativo) Imposta un mirror nel `settings.xml` file che acquisisca tutte le connessioni e le indirizzi al repository anziché a un repository gateway.

Note

Se copiate le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
<mirrors>  
  <mirror>  
    <id>repo_name</id>  
    <name>repo_name</name>  
    <url>https://packages.region.codecatalyst.aws/  
maven/space_name/proj_name/repo_name/</url>  
    <mirrorOf>*</mirrorOf>  
  </mirror>  
</mirrors>
```

Important

È possibile utilizzare qualsiasi valore nell'<id>elemento, ma deve essere lo stesso in entrambi <server> gli <repository> elementi. Ciò consente di includere le credenziali specificate nelle richieste di CodeCatalyst.

Dopo aver apportato queste modifiche alla configurazione, puoi creare il progetto.

```
mvn compile
```

Recupero di pacchetti da archivi di pacchetti esterni tramite CodeCatalyst

È possibile installare i pacchetti Maven dai repository pubblici tramite un CodeCatalyst repository configurandolo con una connessione upstream al gateway che rappresenta il repository del gateway. I pacchetti installati dal repository del gateway vengono inseriti e archiviati nel repository dell'utente. CodeCatalyst

Attualmente, CodeCatalyst supporta i seguenti archivi pubblici di pacchetti Maven.

- maven-central-gateway
- google-android-gateway
- gradle-plugins-gateway
- commonsware-gateway

Per installare pacchetti dagli archivi pubblici di pacchetti Maven

1. Se non l'hai già fatto, esegui la configurazione mvn con il tuo repository di CodeCatalyst pacchetti seguendo i passaggi riportati di seguito. [Recupero delle dipendenze da CodeCatalyst](#)
2. Assicurati che il tuo repository abbia aggiunto il repository gateway da cui desideri installare come connessione upstream. Per verificare quali sorgenti upstream vengono aggiunte o per aggiungere un repository gateway come sorgente upstream, segui le istruzioni riportate in. [Aggiungere un repository upstream](#)

Per ulteriori informazioni sulla richiesta di pacchetti dai repository upstream, consulta. [Richiesta di una versione del pacchetto con repository upstream](#)

Pubblicazione di pacchetti su CodeCatalyst

Per pubblicare un pacchetto Maven in un CodeCatalyst repository, devi anche modificare `~/.m2/settings.xml` e progettare POM. mvn

Da usare per **mvn** pubblicare pacchetti nel tuo repository di pacchetti CodeCatalyst

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella pagina di panoramica del progetto, scegli Pacchetti.
3. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
4. Scegli Connect to repository.

5. Nella finestra di dialogo Connect to repository, scegliete mvn dall'elenco dei client del gestore di pacchetti.
6. Avrai bisogno di un token di accesso personale (PAT) con cui autenticarti. mvn CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.

 Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

7. Configura una variabile di ambiente sul tuo computer locale con il tuo PAT. Utilizzerai questa variabile di ambiente nel tuo `setting.xml` file.

```
export CODECATALYST_ARTIFACTS_TOKEN=your_PAT
```

8. Aggiungi una `<servers>` sezione a `settings.xml` con un riferimento alla variabile di `CodeCatalyst_ARTIFACTS_TOKEN` ambiente in modo che Maven passi il token nelle richieste HTTP.

```
<settings>
...
  <servers>
    <server>
      <id>repo-name</id>
      <username>username</username>
      <password>${env.CodeCatalyst_ARTIFACTS_TOKEN}</password>
    </server>
  </servers>
...
</settings>
```

9. Aggiungi una `<distributionManagement>` sezione a quella del tuo progetto. `pom.xml`

```
<project>
...
  <distributionManagement>
    <repository>
      <id>repo_name</id>
```

```
<name>repo_name</name>
<url>https://packages.region.codecatalyst.aws/
maven/space_name/proj_name/repo_name</url>
</repository>
</distributionManagement>
...
</project>
```

Dopo aver apportato queste modifiche alla configurazione, puoi creare il progetto e pubblicarlo nel repository specificato.

```
mvn deploy
```

È possibile accedere all'archivio dei pacchetti nella CodeCatalyst console per verificare che il pacchetto sia stato pubblicato correttamente.

Pubblicazione di pacchetti di terze parti

Puoi pubblicare pacchetti Maven di terze parti in un CodeCatalyst repository con. `mvn deploy:deploy-file` Questo può essere utile per gli utenti che desiderano pubblicare pacchetti e dispongono solo di file JAR e non hanno accesso al codice sorgente del pacchetto o ai file POM.

Il `mvn deploy:deploy-file` comando genererà un file POM basato sulle informazioni passate nella riga di comando.

Innanzitutto, crea un PAT se non ne hai uno.

Per creare un token di accesso personale (PAT)

1. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri relativa a un progetto o a uno spazio e scegliendo il tuo nome dall'elenco dei membri.

2. Nel nome PAT, inserisci un nome descrittivo per il tuo PAT.
3. In Data di scadenza, lascia la data predefinita o scegli l'icona del calendario per selezionare una data personalizzata. La data di scadenza predefinita è un anno dalla data corrente.

4. Scegli Create (Crea).

Puoi anche creare questo token quando scegli Clone repository come repository di origine.

5. Salva il segreto PAT in un luogo sicuro.

Important

Il segreto PAT viene visualizzato solo una volta. Non è possibile recuperarlo dopo aver chiuso la finestra.

Per pubblicare pacchetti Maven di terze parti

1. Crea un `~/.m2/settings.xml` file con i seguenti contenuti:

```
<settings>
  <servers>
    <server>
      <id>repo_name</id>
      <username>username</username>
      <password>PAT</password>
    </server>
  </servers>
</settings>
```

2. Eseguire il comando `mvn deploy:deploy-file`:

```
mvn deploy:deploy-file -DgroupId=commons-cli \
-DartifactId=commons-cli \
-Dversion=1.4 \
-Dfile=./commons-cli-1.4.jar \
-Dpackaging=jar \
-DrepositoryId=repo_name \
-Durl=https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-
name/
```

 Note

L'esempio precedente pubblica `commons-cli 1.4`. Modifica gli argomenti `GroupId`, `artifactId`, `version` e `file` per pubblicare un JAR diverso.

Queste istruzioni si basano su esempi contenuti nella [Guida alla distribuzione di terze parti in un repository remoto tratti dalla documentazione di JARs Apache Maven](#).

Per ulteriori informazioni, consulta questi argomenti sul sito Web del progetto Apache Maven:

- [Configurazione di più repository](#)
- [Riferimento alle impostazioni](#)
- [Gestione della distribuzione](#)
- [Profili](#)

Pacchetti di pubblicazione con curl

Questa sezione mostra come usare il client HTTP per `curl` pubblicare pacchetti Maven in un CodeCatalyst repository di pacchetti. La pubblicazione di pacchetti con `curl` può essere utile se non si dispone o non si desidera installare il client Maven nei propri ambienti.

Per pubblicare un pacchetto Maven con **curl**

1. È necessario memorizzare un token di accesso personale (PAT) in una variabile di ambiente con cui eseguire l'autenticazione. `curl` CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno e configurare la variabile di ambiente.
 - a. Crea un PAT seguendo la procedura riportata di seguito. [Concedi agli utenti l'accesso al repository con token di accesso personali](#) Copia il PAT per memorizzarlo in una variabile di ambiente.
 - b. Sulla riga di comando del computer locale, configura una variabile di ambiente con il tuo PAT.

```
export CodeCatalyst_ARTIFACTS_TOKEN=your_PAT
```

2. Usa il seguente `curl` comando per pubblicare il JAR in un CodeCatalyst repository. Sostituisci `username`, `proj_name`, e `repo_name` con il tuo nome CodeCatalyst utente, nome dello spazio, nome del progetto e nome del repository del pacchetto.

```
curl --request PUT https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/com/mycompany/app/my-app/1.0/my-app-1.0.jar \  
  --user "username:CodeCatalyst_ARTIFACTS_TOKEN" --header "Content-Type: application/octet-stream" \  
  --data-binary @target/path/to/my-app-1.0.jar
```

3. Utilizzate il `curl` comando seguente per pubblicare il POM in un CodeCatalyst repository. Sostituisci `username`, `proj_name`, e `repo_name` con il tuo nome CodeCatalyst utente, nome dello spazio, nome del progetto e nome del repository del pacchetto.

```
curl --request PUT https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/com/mycompany/app/my-app/1.0/my-app-1.0.pom \  
  --user "username:CodeCatalyst_ARTIFACTS_TOKEN" --header "Content-Type: application/octet-stream" \  
  --data-binary @target/my-app-1.0.pom
```

4. A questo punto, il pacchetto Maven sarà nel tuo CodeCatalyst repository con uno stato di Unfinished. Per poter consumare il pacchetto, deve trovarsi nello stato Published. Puoi spostare il pacchetto da Unfinished a Published caricando un `maven-metadata.xml` file nel pacchetto o modificando lo stato nella CodeCatalyst console.
 - a. Opzione 1: utilizza il seguente `curl` comando per aggiungere un `maven-metadata.xml` file al pacchetto. Sostituisci `username`, `proj_name`, e `repo_name` con il tuo nome CodeCatalyst utente, nome dello spazio, nome del progetto e nome del repository del pacchetto.

```
curl --request PUT https://packages.region.codecatalyst.aws/maven/space-name/proj-name/repo-name/com/mycompany/app/my-app/maven-metadata.xml \  
  --user "username:CodeCatalyst_ARTIFACTS_TOKEN" --header "Content-Type: application/octet-stream" \  
  --data-binary @target/maven-metadata.xml
```

Di seguito è riportato un esempio del contenuto di un `maven-metadata.xml` file:

```
<metadata modelVersion="1.1.0">  
  <groupId>com.mycompany.app</groupId>
```

```
<artifactId>my-app</artifactId>
<versioning>
  <latest>1.0</latest>
  <release>1.0</release>
  <versions>
    <version>1.0</version>
  </versions>
  <lastUpdated>20200731090423</lastUpdated>
</versioning>
</metadata>
```

- b. Opzione 2: aggiorna lo stato del pacchetto Published nella CodeCatalyst console. Per informazioni su come aggiornare lo stato di una versione del pacchetto, consulta [Aggiornamento dello stato di una versione del pacchetto](#).

Se disponi solo del file JAR di un pacchetto, puoi pubblicare una versione consumabile del pacchetto in un CodeCatalyst repository utilizzando `mvn`. Ciò può essere utile se non si ha accesso al codice sorgente o al POM del pacchetto. Per informazioni dettagliate, vedi [Pubblicazione di pacchetti di terze parti](#).

Utilizzo dei checksum e delle istantanee di Maven

Le sezioni seguenti descrivono come utilizzare i checksum Maven e le istantanee Maven in CodeCatalyst.

Utilizzo dei checksum Maven

Quando un pacchetto Maven viene pubblicato in un repository di pacchetti, il checksum associato a ogni risorsa o file del CodeCatalyst pacchetto viene utilizzato per convalidare il caricamento. Esempi di risorse sono i file `jar`, `pom` e `war`. Per ogni risorsa, il pacchetto Maven contiene più file di checksum che utilizzano il nome della risorsa con un'estensione aggiuntiva, ad esempio `o.md5` o `sha1`. Ad esempio, i file di checksum per un file denominato `my-maven-package.jar` potrebbero essere `e.my-maven-package.jar.md5` o `my-maven-package.jar.sha1`.

Ogni pacchetto Maven contiene anche un file `maven-metadata.xml`. Questo file deve essere caricato affinché una pubblicazione abbia successo. Se viene rilevata una mancata corrispondenza del checksum durante il caricamento di qualsiasi file di pacchetto, la pubblicazione si interrompe. Ciò potrebbe impedire il caricamento di `maven-metadata.xml`. Quando ciò accade, lo stato del pacchetto Maven è impostato su `Unfinished`. Non è possibile scaricare risorse che fanno parte di un pacchetto con questo stato.

Tieni presente quanto segue in caso di mancata corrispondenza del checksum quando pubblichi un pacchetto Maven:

- Se il checksum non corrisponde prima del caricamento, lo stato del pacchetto non maven-metadata.xml è impostato su. Unfinished Il pacchetto non è visibile e le sue risorse non possono essere consumate. In questo caso, provate una delle seguenti operazioni, quindi provate a scaricare nuovamente la risorsa.
- Esegui nuovamente il comando che pubblica il pacchetto Maven. Questo potrebbe funzionare se un problema di rete ha danneggiato il file di checksum durante il download. Se il problema di rete viene risolto per il nuovo tentativo, il checksum corrisponde e il download ha esito positivo.
- Se la ripubblicazione del pacchetto Maven non funziona, elimina il pacchetto e ripubblicalo.
- Se il checksum non corrisponde dopo il caricamento, lo stato del pacchetto maven-metadata.xml è impostato su. Published Puoi utilizzare qualsiasi risorsa del pacchetto, incluse quelle con i checksum non corrispondenti. Quando scaricate una risorsa, il checksum generato da CodeCatalyst viene scaricato con essa. Se il file scaricato è associato a un codice di controllo non corrispondente, il relativo file di checksum scaricato potrebbe non corrispondere al checksum caricato al momento della pubblicazione del pacchetto.

Utilizzo delle istantanee di Maven

Un'istananea di Maven è una versione speciale di un pacchetto Maven che si riferisce all'ultimo codice filiale di produzione. È una versione di sviluppo che precede la versione di rilascio finale. È possibile identificare una versione istantanea di un pacchetto Maven dal suffisso SNAPSHOT aggiunto alla versione del pacchetto. Ad esempio, l'istananea della versione è. 1.1 1.1-SNAPSHOT Per ulteriori informazioni, consulta [Cos'è una versione SNAPSHOT?](#) sul sito Web del progetto Apache Maven.

CodeCatalyst supporta la pubblicazione e l'utilizzo di istantanee Maven. È possibile pubblicare un'istananea Maven in un CodeCatalyst repository o, se si è connessi direttamente, in un repository upstream. Tuttavia, una versione snapshot sia in un repository di pacchetti che in uno dei relativi repository upstream non è supportata. Ad esempio, se carichi un pacchetto Maven con versione 1.2-SNAPSHOT nel tuo repository di pacchetti, CodeCatalyst non supporta il caricamento di un pacchetto Maven con la stessa versione di snapshot in uno dei suoi repository upstream. Questo scenario potrebbe restituire risultati imprevedibili.

Quando viene pubblicata un'istananea di Maven, la versione precedente viene conservata in una nuova versione chiamata build. Ogni volta che viene pubblicata un'istananea di Maven, viene

creata una nuova versione di build. Tutte le versioni precedenti di un'istanza vengono mantenute nelle relative versioni di build. Quando viene pubblicata un'istanza di Maven, il suo stato viene impostato su `Published` e lo stato della build che contiene la versione precedente è impostato su `Unlisted`.

Se richiedi un'istanza, viene restituita la versione con lo stato `Published`. Questa è sempre la versione più recente dell'istanza di Maven. Puoi anche richiedere una build particolare di un'istanza.

Per eliminare tutte le versioni di build di un'istanza di Maven, usa la console. CodeCatalyst

Usando NuGet

Questi argomenti descrivono come utilizzare e pubblicare NuGet pacchetti utilizzando CodeCatalyst.

Note

CodeCatalyst supporta [NuGet la versione 4.8](#) e successive.

Argomenti

- [Utilizzo CodeCatalyst con Visual Studio](#)
- [Configurazione e utilizzo della CLI nuget o dotnet](#)
- [NuGet normalizzazione del nome del pacchetto, della versione e del nome dell'asset](#)
- [NuGet compatibilità](#)

Utilizzo CodeCatalyst con Visual Studio

Puoi utilizzare i pacchetti CodeCatalyst direttamente da Visual Studio.

Per configurare e utilizzare NuGet con strumenti CLI come `dotnet` o `nuget`, vedere. [Configurazione e utilizzo della CLI nuget o dotnet](#)

Indice

- [Configurazione di Visual Studio con CodeCatalyst](#)
 - [Windows](#)
 - [macOS](#)

Configurazione di Visual Studio con CodeCatalyst

Windows

Per configurare Visual Studio con CodeCatalyst

1. Per l'autenticazione è necessario un token di accesso personale (PAT) con CodeCatalyst. Se ne hai già uno, puoi usarlo. In caso contrario, segui le istruzioni [Concedi agli utenti l'accesso al repository con token di accesso personali](#) per crearne uno.
2. Usa nuget o dotnet per configurare l'archivio dei pacchetti e le credenziali.

dotnet

Utenti Linux e macOS: poiché la crittografia non è supportata su piattaforme diverse da Windows, è necessario aggiungere il `--store-password-in-clear-text` flag al comando seguente. Tieni presente che in questo modo la password verrà memorizzata come testo semplice nel file di configurazione.

```
dotnet nuget add source https://packages.region.codecatalyst.aws/nuget/space-name/proj-name/repo-name/v3/index.json --name repo_name --password PAT --username user_name
```

nuget

```
nuget sources add -name repo_name -Source https://packages.region.codecatalyst.aws/nuget/space-name/proj-name/repo-name/v3/index.json -password PAT --username user_name
```

Output di esempio:

```
Package source with Name: repo_name added successfully.
```

3. Configura Visual Studio per utilizzare la nuova fonte del pacchetto. In Visual Studio, scegli Strumenti, quindi scegli Opzioni.
4. Nel menu Opzioni, espandi la sezione NuGet Package Manager e scegli Package Sources.
5. Nell'elenco Sorgenti dei pacchetti disponibili, assicurati che la tua *repo_name* fonte sia abilitata. Se hai configurato il tuo repository di pacchetti con una connessione upstream alla NuGet Galleria, disabilita il sorgente nuget.org.

macOS

Per configurare Visual Studio con CodeCatalyst

1. Per l'autenticazione è necessario un token di accesso personale (PAT) con. CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, segui le istruzioni [Concedi agli utenti l'accesso al repository con token di accesso personali](#) per crearne uno.
2. Scegli Preferenze dalla barra dei menu.
3. Nella NuGetsezione, scegli Sorgenti.
4. Scegli Aggiungi e aggiungi le informazioni del tuo repository.
 - a. In Nome, inserisci il nome CodeCatalyst del repository del pacchetto.
 - b. In Ubicazione, inserite l'endpoint CodeCatalyst del repository dei pacchetti. Il seguente frammento mostra un endpoint di esempio. Sostituisci *space-name* e *repo-name* con il nome CodeCatalyst dello spazio, il nome del progetto e il nome del repository. *proj-name*

```
https://packages.region.codecatalyst.aws/nuget/space-name/proj-name/repo-name/
```
 - c. Per Nome utente, inserisci un valore valido.
 - d. Per Password, inserisci il tuo PAT.
5. Scegliere Add source (Aggiungi origine).
6. Se hai configurato il tuo repository di pacchetti con una connessione upstream alla NuGet Galleria, disabilita il sorgente nuget.org.

Dopo la configurazione, Visual Studio può utilizzare i pacchetti dal tuo CodeCatalyst repository, da uno qualsiasi dei suoi repository upstream o da [NuGet.org](#) se lo hai configurato come sorgente upstream. Per altre informazioni sulla navigazione e l'installazione NuGet dei pacchetti in Visual Studio, vedi [Installare e gestire i pacchetti in Visual Studio utilizzando il NuGet Package Manager](#) nella NuGet documentazione.

Configurazione e utilizzo della CLI nuget o dotnet

È possibile utilizzare strumenti CLI come NuGet e dotnet da cui pubblicare e utilizzare pacchetti. CodeCatalyst Questo documento fornisce informazioni sulla configurazione degli strumenti CLI e sul loro utilizzo per pubblicare o utilizzare pacchetti.

Indice

- [Configurazione con NuGet CodeCatalyst](#)
- [Consumo di pacchetti NuGet da un repository CodeCatalyst](#)
- [Consumo di pacchetti da .org a NuGet NuGet CodeCatalyst](#)
- [Pacchetti di pubblicazione su NuGet CodeCatalyst](#)

Configurazione con NuGet CodeCatalyst

Per eseguire la configurazione NuGet CodeCatalyst, aggiungi un endpoint del repository e un token di accesso personale al tuo file di NuGet configurazione per consentire o connetterti al tuo repository di pacchetti. `nuget dotnet CodeCatalyst`

Da configurare con il tuo repository NuGet di pacchetti CodeCatalyst

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella pagina di panoramica del progetto, scegli Pacchetti.
3. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
4. Scegli Connect to repository.
5. Nella finestra di dialogo Connect to repository, scegliete NuGetor dotnet dall'elenco dei client del gestore di pacchetti.
6. Avrai bisogno di un token di accesso personale (PAT) con cui autenticarti. NuGet CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.



Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

7. Configura `nuget` o utilizza `dotnet` l' NuGet endpoint e il PAT del repository. CodeCatalyst Sostituisci i seguenti valori.



Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *username* Sostituiscilo con il tuo nome CodeCatalyst utente.
- *PAT* Sostituiscilo con il tuo CodeCatalyst PAT.
- Sostituisci *space_name* con il nome CodeCatalyst dello spazio.
- Sostituisci *proj_name* con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

- a. Per nuget, utilizzare il comando `nuget sources add`.

```
nuget sources add -name "repo_name" -Source "https://  
packages.region.codecatalyst.aws/nuget/space_name/proj_name/repo_name/v3/  
index.json" -username "username" -password "PAT"
```

- b. Per dotnet, utilizzare il comando `dotnet nuget add source`.

Utenti Linux e macOS: poiché la crittografia non è supportata su piattaforme diverse da Windows, è necessario aggiungere il `--store-password-in-clear-text` flag al comando seguente. Tieni presente che questa operazione memorizzerà la password come testo semplice nel file di configurazione.

```
dotnet nuget add source "https://packages.region.codecatalyst.aws/  
nuget/space_name/proj_name/repo_name/v3/index.json" -n "proj_name/repo_name" -u  
"username" -p "PAT" --store-password-in-clear-text
```

Dopo aver configurato NuGet con CodeCatalyst, potete [utilizzare NuGet i pacchetti](#) archiviati nel vostro CodeCatalyst repository o in uno dei suoi repository upstream e [pubblicare NuGet](#) i pacchetti nel vostro repository. CodeCatalyst

Consumo di pacchetti NuGet da un repository CodeCatalyst

Dopo aver [configurato NuGet con CodeCatalyst](#), puoi utilizzare i NuGet pacchetti archiviati nel tuo CodeCatalyst repository o in uno dei suoi repository upstream.

Per utilizzare una versione del pacchetto da un CodeCatalyst repository o da uno dei suoi repository upstream con nuget o dotnet, esegui il comando seguente. Sostituiscilo *packageName* con il nome del pacchetto che desideri consumare e *packageSourceName* con il nome sorgente del tuo

repository di CodeCatalyst pacchetti nel tuo file di NuGet configurazione, che dovrebbe essere il nome del repository.

Per installare un pacchetto con **dotnet**

```
dotnet add packageName --source packageSourceName
```

Per installare un pacchetto con **nuget**

```
nuget install packageName --source packageSourceName
```

Per ulteriori informazioni, consulta [Gestire i pacchetti utilizzando la CLI di nuget o Installare e gestire i pacchetti utilizzando la CLI dotnet](#) nella documentazione Microsoft.

Consumo di pacchetti da .org a NuGet NuGet CodeCatalyst

È possibile utilizzare NuGet pacchetti da [NuGet.org](#) tramite un CodeCatalyst repository configurando il repository con una connessione upstream a .org. NuGet I pacchetti consumati da NuGet.org vengono importati e archiviati nel tuo repository. CodeCatalyst

Per consumare pacchetti da .org NuGet

1. Se non l'hai già fatto, configura il tuo gestore di NuGet pacchetti con il tuo repository di CodeCatalyst pacchetti seguendo i passaggi riportati di seguito. [Configurazione con NuGet CodeCatalyst](#)
2. Assicurati che il tuo repository abbia aggiunto NuGet.org come connessione upstream. Puoi controllare quali sorgenti upstream vengono aggiunte o aggiungere NuGet.org come fonte upstream seguendo le istruzioni e scegliendo lo store repository. [Aggiungere un repository upstream](#) NuGet

Pacchetti di pubblicazione su NuGet CodeCatalyst

Dopo aver [configurato NuGet con CodeCatalyst](#), puoi utilizzare nuget o pubblicare le versioni dei pacchetti nei dotnet repository. CodeCatalyst

Per inviare una versione del pacchetto a un CodeCatalyst repository, esegui il comando seguente con il percorso completo del .nupkg file e il nome sorgente del CodeCatalyst repository nel file di configurazione NuGet .

Per pubblicare un pacchetto con **dotnet**

```
dotnet nuget push path/to/nupkg/SamplePackage.1.0.0.nupkg --source packageSourceName
```

Per pubblicare un pacchetto con **nuget**

```
nuget push path/to/nupkg/SamplePackage.1.0.0.nupkg --source packageSourceName
```

NuGet normalizzazione del nome del pacchetto, della versione e del nome dell'asset

CodeCatalyst normalizza i nomi e le versioni dei pacchetti e delle risorse prima di archivarli, il che significa che i nomi o le versioni dei pacchetti CodeCatalyst possono essere diversi da quelli forniti al momento della pubblicazione del pacchetto o della risorsa.

Normalizzazione dei nomi dei pacchetti: CodeCatalyst normalizza i nomi dei NuGet pacchetti convertendo tutte le lettere in minuscolo.

Normalizzazione della versione del NuGet pacchetto: CodeCatalyst normalizza le versioni dei pacchetti utilizzando lo stesso schema di NuGet. Le seguenti informazioni sono tratte dai [numeri di versione normalizzati](#) riportati nella documentazione. NuGet

- Gli zeri iniziali vengono rimossi dai numeri di versione:
 - 1.00 viene trattato come 1.0
 - 1.01.1 è trattato come 1.1.1
 - 1.00.0.1 è trattato come 1.0.0.1
- Verrà omesso uno zero nella quarta parte del numero di versione:
 - 1.0.0.0 viene trattato come 1.0.0
 - 1.0.01.0 è trattato come 1.0.1
- SemVer i metadati della build 2.0.0 vengono rimossi:
 - 1.0.7+r3456 viene trattato come 1.0.7

Normalizzazione del nome dell'asset del pacchetto: CodeCatalyst costruisce il nome dell'asset del NuGet pacchetto dal nome del pacchetto e dalla versione del pacchetto normalizzati.

NuGet compatibilità

Questa guida contiene informazioni sulla CodeCatalyst compatibilità con diversi NuGet strumenti e versioni.

Argomenti

- [NuGet Compatibilità generale](#)
- [NuGet supporto da riga di comando](#)

NuGet Compatibilità generale

CodeCatalyst supporta NuGet 4.8 e versioni successive.

CodeCatalyst supporta solo la versione 3 del NuGet protocollo HTTP. Ciò significa che alcuni comandi CLI che si basano sulla versione 2 del protocollo non sono supportati. Per ulteriori informazioni, consulta la [supporto per i comandi nuget](#) sezione seguente.

CodeCatalyst non supporta PowerShellGet 2.x.

NuGet supporto da riga di comando

CodeCatalyst supporta gli NuGet strumenti CLI (nuget) e .NET Core (dotnet).

supporto per i comandi nuget

Poiché supporta CodeCatalyst solo il protocollo HTTP NuGet di V3 of, i seguenti comandi non funzioneranno se utilizzati contro risorse: CodeCatalyst

- `list`: Il `nuget list` comando visualizza un elenco di pacchetti da una determinata fonte. Per ottenere un elenco di pacchetti in un repository di CodeCatalyst pacchetti, accedi al repository nella CodeCatalyst console.

Uso di Python

Questi argomenti descrivono come usare `pip`, il gestore di pacchetti Python e `twine` l'utilità di pubblicazione dei pacchetti Python con. CodeCatalyst

Argomenti

- [Configurazione di pip e installazione di pacchetti Python](#)

- [Configurazione di Twine e pubblicazione di pacchetti Python](#)
- [Normalizzazione dei nomi dei pacchetti in Python](#)
- [Compatibilità con Python](#)

Configurazione di pip e installazione di pacchetti Python

Per pip utilizzarlo CodeCatalyst, è necessario connettersi pip al repository dei pacchetti e fornire un token di accesso personale per l'autenticazione. È possibile visualizzare le istruzioni per la connessione pip al repository dei pacchetti nella CodeCatalyst console. Dopo l'autenticazione e la connessione pip CodeCatalyst, puoi eseguire pip i comandi.

Indice

- [Installazione di pacchetti Python da CodeCatalyst con pip](#)
- [Consumo di pacchetti Python da PyPI tramite CodeCatalyst](#)
- [supporto per i comandi pip](#)
 - [Comandi supportati che interagiscono con un repository](#)
 - [Comandi lato client supportati](#)

Installazione di pacchetti Python da CodeCatalyst con pip

Le seguenti istruzioni spiegano come pip configurare l'installazione dei pacchetti Python dal tuo CodeCatalyst repository di pacchetti o da uno dei suoi repository upstream.

Da configurare e utilizzare **pip** per installare i pacchetti Python dal tuo CodeCatalyst repository di pacchetti

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Nella pagina di panoramica del progetto, scegli Pacchetti.
3. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
4. Scegli Connect to repository.
5. Nella finestra di dialogo Connect to repository, scegli pip dall'elenco dei client del gestore di pacchetti.
6. Avrai bisogno di un token di accesso personale (PAT) con cui autenticare pip. CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.

- a. Scegli Crea token.
- b. Scegli Copia per copiare il tuo PAT.

 Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

7. Utilizzate il `pip config` comando per impostare l'URL e le credenziali del CodeCatalyst registro. Sostituisci i seguenti valori.

 Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- `username` Sostituiscilo con il tuo nome CodeCatalyst utente.
- `PAT` Sostituiscilo con il tuo CodeCatalyst PAT.
- `space_name` Sostituiscilo con il nome CodeCatalyst dello spazio.
- `proj_name` Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- `repo_name` Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
pip config set global.index-url https://username:PAT@https://  
packages.region.codecatalyst.aws/pypi/space_name/proj_name/repo_name/simple/
```

8. Supponendo che un pacchetto sia presente nel tuo repository o in uno dei suoi repository originali, puoi installarlo con. `pip install` Ad esempio, utilizzate il comando seguente per installare il pacchetto. `requests`

```
pip install requests
```

Utilizzate l'-i opzione per tornare temporaneamente all'installazione dei pacchetti da <https://pypi.org> anziché dal vostro repository di CodeCatalyst pacchetti.

```
pip install -i https://pypi.org/simple requests
```

Consumo di pacchetti Python da PyPI tramite CodeCatalyst

È possibile utilizzare pacchetti Python dal Python [Package Index \(PyPI\)](#) tramite un CodeCatalyst repository configurando il repository con una connessione upstream a PyPI. I pacchetti consumati da PyPI vengono importati e archiviati nel tuo repository. CodeCatalyst

Per utilizzare pacchetti da PyPI

1. Se non l'hai già fatto, configura pip con il tuo repository di CodeCatalyst pacchetti seguendo i passaggi indicati. [Installazione di pacchetti Python da CodeCatalyst con pip](#)
2. Assicurati che il tuo repository abbia aggiunto PyPI come sorgente upstream. Puoi controllare quali sorgenti upstream vengono aggiunte o aggiungere PyPI come sorgente upstream seguendo le istruzioni e scegliendo il repository [Aggiungere un repository upstream](#) PyPI store.

Per ulteriori informazioni sulla richiesta di pacchetti dai repository upstream, consulta. [Richiesta di una versione del pacchetto con repository upstream](#)

supporto per i comandi pip

Le sezioni seguenti riepilogano i comandi pip supportati dai CodeCatalyst repository, oltre ai comandi specifici che non sono supportati.

Argomenti

- [Comandi supportati che interagiscono con un repository](#)
- [Comandi lato client supportati](#)

Comandi supportati che interagiscono con un repository

Questa sezione elenca pip i comandi in cui il pip client effettua una o più richieste al registro con cui è stato configurato. È stato verificato che questi comandi funzionino correttamente quando vengono richiamati CodeCatalyst su un archivio di pacchetti.

Comando	Description
installare	Installa pacchetti.
scaricare	Scarica i pacchetti.

CodeCatalyst non implementa `pip search`. Se hai configurato `pip` con un repository di CodeCatalyst pacchetti, l'esecuzione `pip search` cercherà e mostrerà i pacchetti da [PyPI](#).

Comandi lato client supportati

Questi comandi non richiedono alcuna interazione diretta con un repository, quindi CodeCatalyst non è necessario fare nulla per supportarli.

Comando	Description
disinstallare	Disinstalla i pacchetti.
congelare	Visualizza i pacchetti installati nel formato dei requisiti.
elenco	Elenca i pacchetti installati.
show	Mostra informazioni sui pacchetti installati.
controlla	Verifica che i pacchetti installati abbiano dipendenze compatibili.
configurazione	Gestisci la configurazione locale e globale.
ruota	Costruisci ruote in base alle tue esigenze.
cancelletto	Calcola gli hash degli archivi dei pacchetti.
completamento	Aiuta a completare i comandi.
debug	Mostra informazioni utili per il debug.
aiuto	Mostra l'aiuto per i comandi.

Configurazione di Twine e pubblicazione di pacchetti Python

Per `twine` utilizzarlo CodeCatalyst, è necessario connettersi `twine` al repository dei pacchetti e fornire un token di accesso personale per l'autenticazione. È possibile visualizzare le istruzioni per la connessione `twine` al repository dei pacchetti nella CodeCatalyst console. Dopo l'autenticazione e la connessione `twine` CodeCatalyst, puoi eseguire `twine` i comandi.

Pacchetti di pubblicazione su CodeCatalyst with Twine

Le seguenti istruzioni spiegano come autenticarsi e connettersi twine al repository dei CodeCatalyst pacchetti.

Da configurare e utilizzare **twine** per pubblicare pacchetti nel proprio CodeCatalyst repository di pacchetti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella pagina di panoramica del progetto, scegli Pacchetti.
3. Scegli il tuo repository di pacchetti dall'elenco dei repository di pacchetti.
4. Scegli Connect to repository.
5. Nella finestra di dialogo Connect to repository, scegli Twine dall'elenco dei client del gestore di pacchetti.
6. Avrai bisogno di un token di accesso personale (PAT) con cui autenticare Twine. CodeCatalyst Se ne hai già uno, puoi usarlo. In caso contrario, puoi crearne uno qui.
 - a. Scegli Crea token.
 - b. Scegli Copia per copiare il tuo PAT.

Warning

Non potrai più vedere o copiare il tuo PAT dopo aver chiuso la finestra di dialogo.

7. È possibile configurare twine con un `.pypirc` file o con variabili di ambiente.
 - a. Da configurare con un **.pypirc** file.

Apri `~/.pypirc` nell'editor che preferisci.

Aggiungi un server di indicizzazione per CodeCatalyst, incluso il repository, il nome utente e il PAT che hai creato e copiato in un passaggio precedente. Sostituisci i seguenti valori.

Note

Se si copiano le istruzioni dalla console, i seguenti valori devono essere aggiornati automaticamente e non devono essere modificati.

- *username* Sostituiscilo con il tuo nome CodeCatalyst utente.
- *PAT* Sostituiscilo con il tuo CodeCatalyst PAT.
- *space_name* Sostituiscilo con il nome CodeCatalyst dello spazio.
- *proj_name* Sostituiscilo con il nome CodeCatalyst del tuo progetto.
- *repo_name* Sostituiscilo con il nome CodeCatalyst del repository dei pacchetti.

```
[distutils]
index-servers = proj-name/repo-name

[proj-name/repo-name]
repository = https://packages.region.codecatalyst.aws/
pypi/space_name/proj_name/repo_name/
password = PAT
username = username
```

- b. Da configurare con variabili di ambiente.

Imposta le seguenti variabili di ambiente. Nel `TWINE_REPOSITORY_URL` valore *space_name*, *proj_name* aggiorna e *repo_name* inserisci i nomi CodeCatalyst dello spazio, del progetto e del repository dei pacchetti.

```
export TWINE_USERNAME=username
```

```
export TWINE_PASSWORD=PAT
```

```
export TWINE_REPOSITORY_URL="https://packages.region.codecatalyst.aws/
pypi/space_name/proj_name/repo_name/"
```

8. Pubblica una distribuzione Python con il `twine upload` comando.

Normalizzazione dei nomi dei pacchetti in Python

CodeCatalyst normalizza i nomi dei pacchetti prima di memorizzarli, il che significa che i nomi dei pacchetti CodeCatalyst possono essere diversi dal nome fornito quando il pacchetto è stato pubblicato.

Per i pacchetti Python, quando si esegue la normalizzazione il nome del pacchetto è in minuscolo e tutte le istanze dei caratteri vengono sostituite con un . singolo - carattere. _ - Quindi i nomi dei pacchetti e sono normalizzati `pigeon_cli` e `pigeon.cli` memorizzati come `pigeon-cli`. Il nome non normalizzato può essere usato da pip e twine. Per ulteriori informazioni sulla normalizzazione dei nomi dei pacchetti Python, vedere [PEP 503](#) nella documentazione di Python.

Compatibilità con Python

Sebbene CodeCatalyst non supporti l'/simple/API, supporta le operazioni Legacy API. CodeCatalyst non supporta le operazioni PyPI XML-RPC o JSON API.

Per ulteriori informazioni, vedere quanto segue nel repository della Python Packaging Authority. GitHub

- [API legacy](#)
- [API XML-RPC](#)
- [API JSON](#)

Quote per i pacchi

La tabella seguente descrive le quote e i limiti per i pacchi in Amazon CodeCatalyst. Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Risorsa	Quota predefinita
Archivi di pacchetti	Massimo 1000 per spazio.
Archivi diretti a monte	Massimo 10 per archivio di pacchetti.
Ricerca negli archivi di pacchetti upstream	Massimo 25 repository upstream ricercati per versione del pacchetto richiesta.
Dimensione del file di asset del pacchetto	Massimo 5 GB per risorsa del pacchetto.
Risorse del pacchetto	Massimo 150 per versione del pacchetto.

Crea, testa e distribuisce con i flussi di lavoro

Dopo aver scritto il codice dell'applicazione in un [ambiente di CodeCatalyst sviluppo](#) e averlo inviato al [repository di CodeCatalyst origine](#), sei pronto per distribuirlo. Il modo per farlo automaticamente è tramite un flusso di lavoro.

Un flusso di lavoro è una procedura automatizzata che descrive come creare, testare e distribuire il codice come parte di un sistema di integrazione e distribuzione continue (CI/CD). Un flusso di lavoro definisce una serie di passaggi, o azioni, da eseguire durante l'esecuzione di un flusso di lavoro. Un flusso di lavoro definisce anche gli eventi, o trigger, che causano l'avvio del flusso di lavoro. Per configurare un flusso di lavoro, è necessario creare un file di definizione del flusso di lavoro utilizzando l'editor [visivo o YAML](#) della CodeCatalyst console.

Tip

Per una rapida occhiata a come potresti utilizzare i flussi di lavoro in un progetto, [crea un progetto con un blueprint](#). Ogni blueprint implementa un flusso di lavoro funzionante che puoi rivedere, eseguire e sperimentare.

Informazioni sul file di definizione del flusso di lavoro

Un file di definizione del flusso di lavoro è un file YAML che descrive il flusso di lavoro. Per impostazione predefinita, il file viene archiviato in una `~/.codecatalyst/workflows/` cartella nella radice del repository di [origine](#). Il file può avere un'estensione `.yaml` o `.yml` e l'estensione deve essere minuscola.

Di seguito è riportato un esempio di un semplice file di definizione del flusso di lavoro. Spieghiamo ogni riga di questo esempio nella tabella che segue.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
```

```

Build:
  Identifier: aws/build@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    Steps:
      - Run: docker build -t MyApp:latest .

```

Line (Linea)	Description
Name: MyWorkflow	Specificate il nome del flusso di lavoro. Per ulteriori informazioni sulla Name proprietà, vedere Proprietà di primo livello .
SchemaVersion: 1.0	Specifica la versione dello schema del flusso di lavoro. Per ulteriori informazioni sulla SchemaVersion proprietà, vedere Proprietà di primo livello .
RunMode: QUEUED	Indica come CodeCatalyst gestisce più esecuzioni. Per ulteriori informazioni sulla modalità di esecuzione, vedere Configurazione del comportamento di accodamento delle esecuzioni .
Triggers:	Specifica la logica che determinerà l'avvio di un flusso di lavoro. Per ulteriori informazioni sui trigger, consulta L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger .
- Type: PUSH Branches: - main	Indica che il flusso di lavoro deve iniziare ogni volta che si invia codice al main ramo dell'archivio di origine predefinito. Per ulteriori informazioni sull'origine del flusso di lavoro, vedere Connessione degli archivi di origine ai flussi di lavoro .

Line (Linea)	Description
Actions:	Definisce le attività da eseguire durante l'esecuzione di un flusso di lavoro. In questo esempio, la Actions sezione definisce una singola azione chiamata <code>Build</code> . Per ulteriori informazioni sulle azioni, vedere Configurazione delle azioni del flusso di lavoro .
Build:	Definisce le proprietà dell'Buildazione. Per ulteriori informazioni sull'azione di compilazione, vedere Creazione con flussi di lavoro .
Identifier: <code>aws/builddev1</code>	Specifica l'identificatore univoco e codificato per l'azione di compilazione.
Inputs: Sources: - <code>WorkflowSource</code>	Indica che l'azione di compilazione deve cercare nel repository di <code>WorkflowSource</code> origine i file necessari per completare l'elaborazione. Per ulteriori informazioni, consulta Connessione degli archivi di origine ai flussi di lavoro .
Configuration:	Contiene le proprietà di configurazione specifiche dell'azione di compilazione.
Steps: - Run: <code>docker build -t MyApp:latest .</code>	Indica all'azione di compilazione di creare un'immagine Docker chiamata <code>MyApp</code> e di etichettarla. <code>latest</code>

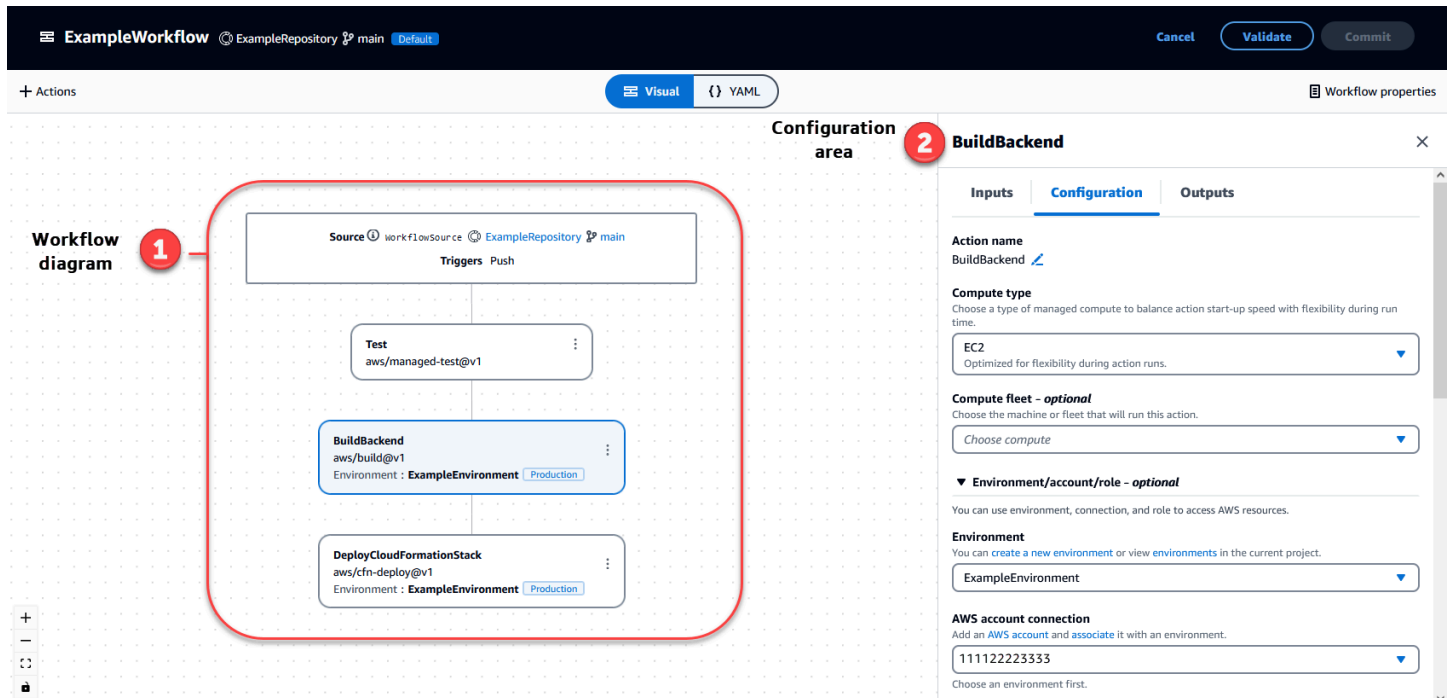
Per un elenco completo di tutte le proprietà disponibili nel file di definizione del flusso di lavoro, consulta [Definizione YAML del flusso di lavoro](#).

Utilizzo degli editor visivi e YAML della CodeCatalyst console

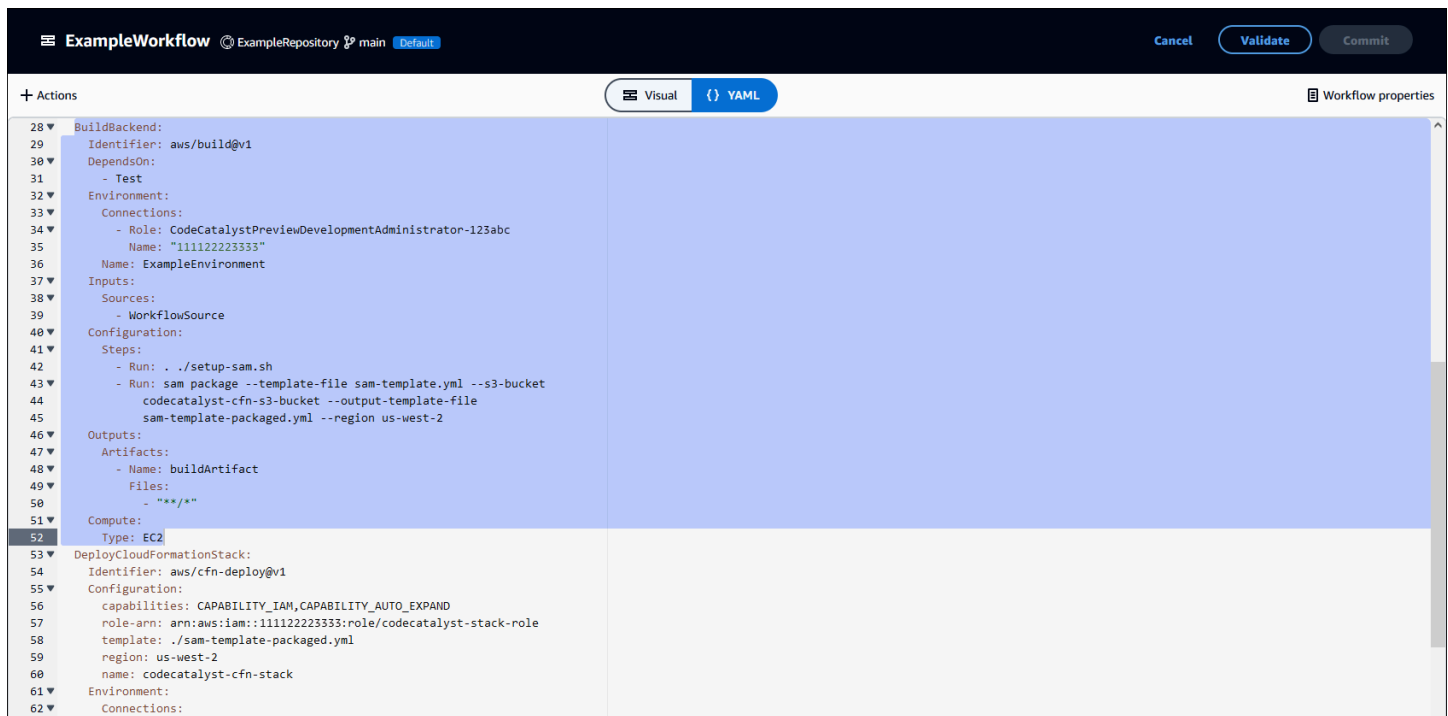
Per creare e modificare il file di definizione del flusso di lavoro, puoi utilizzare il tuo editor preferito, ma ti consigliamo di utilizzare l'editor visivo o l'editor YAML della CodeCatalyst console. Questi

editor offrono un'utile convalida dei file per garantire che i nomi delle proprietà YAML, i valori, la nidificazione, la spaziatura, l'uso delle maiuscole e così via siano corretti.

L'immagine seguente mostra un flusso di lavoro nell'editor visivo. L'editor visivo offre un'interfaccia utente completa attraverso la quale creare e configurare il file di definizione del flusso di lavoro. L'editor visivo include un diagramma del flusso di lavoro (1) che mostra i componenti principali del flusso di lavoro e un'area di configurazione (2).



In alternativa, puoi usare l'editor YAML, mostrato nell'immagine successiva. Utilizzate l'editor YAML per incollare blocchi di codice di grandi dimensioni (tratti da un tutorial, ad esempio) o per aggiungere proprietà avanzate che non sono disponibili tramite l'editor visuale.

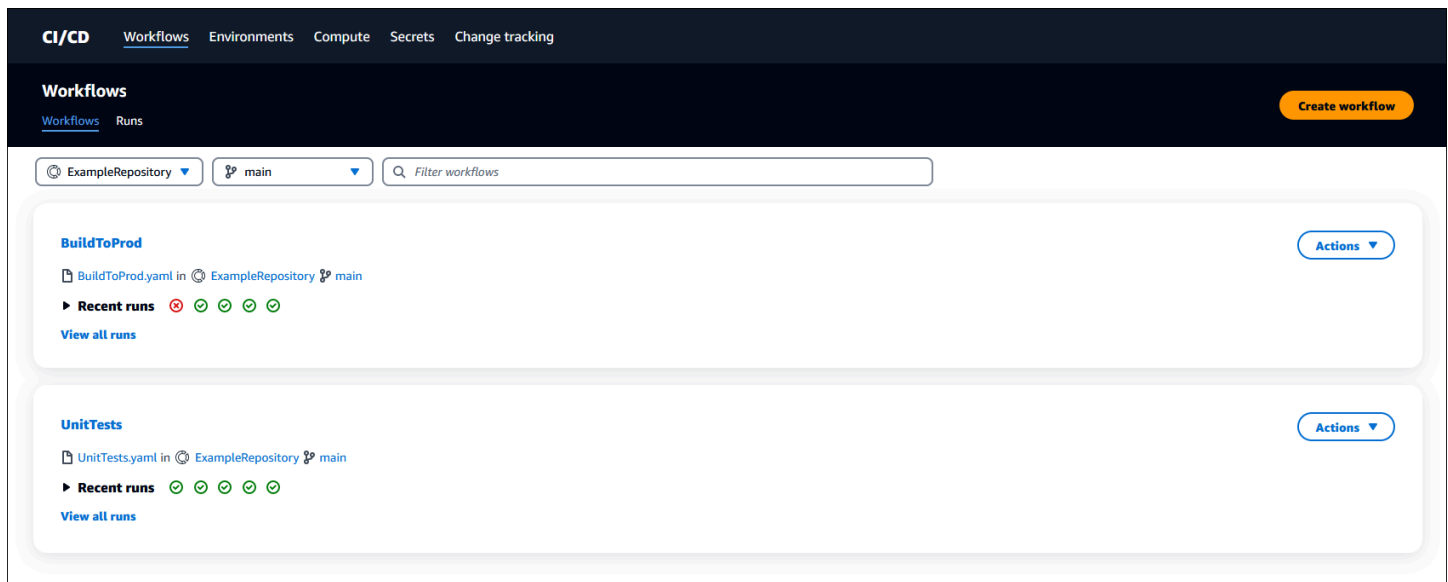


Puoi passare dall'editor visivo all'editor YAML per vedere l'effetto che le tue configurazioni hanno sul codice YAML sottostante.

Alla scoperta dei flussi di lavoro

Puoi visualizzare il tuo flusso di lavoro nella pagina di riepilogo Flussi di lavoro, insieme ad altri flussi di lavoro che hai impostato nello stesso progetto.

L'immagine seguente mostra la pagina di riepilogo dei flussi di lavoro. È popolata da due flussi di lavoro: e. BuildToProdUnitTests Puoi vedere che entrambi sono stati eseguiti alcune volte. Puoi scegliere Esecuzioni recenti per visualizzare rapidamente la cronologia delle esecuzioni oppure scegliere il nome del flusso di lavoro per visualizzare il codice YAML del flusso di lavoro e altre informazioni dettagliate.



Visualizzazione dei dettagli sull'esecuzione del flusso di lavoro

È possibile visualizzare i dettagli di un flusso di lavoro eseguito selezionando l'esecuzione nella pagina di riepilogo dei flussi di lavoro.

L'immagine seguente mostra i dettagli di un workflow eseguito chiamato RUN-CC11D che è stato avviato automaticamente su un commit to source. Il diagramma del flusso di lavoro indica che un'azione non è riuscita (1). È possibile accedere ai log (2) per visualizzare i messaggi di registro dettagliati e risolvere i problemi. Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, vedere.

[Esecuzione di un workflow](#)

ExampleWorkflow ExampleRepos... main **Run-cc11d**

Latest state Definition Runs

Status Failed **Run mode** Queued **Trigger** Started by 0f8bf654 **Start time** 11 minutes ago **Duration** 3 minutes 17 seconds **End time** 8 minutes ago

WorkflowSource ExampleRepository main Commit: 0f8bf654

Test aws/managed-test@v1

Failed action **BuildBackend** aws/build@v1 Environment: ExampleEnvironment Production

DeployCloudFormationStack aws/cfn-deploy@v1 Environment: ExampleEnvironment Production

Detailed log messages

BuildBackend Failed Start time: 9 minutes ago Duration: 1 minute 10 seconds

- Restore cache < 1 second
- ./setup-sam.sh 14 seconds
- sam package --template-file sam-template.yml --s3-bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml --region us-west-2 3 seconds

Showing the last 13 lines.

```

1 [Container] 2023/04/05 15:04:37 Running command sam package --template-file sam-temp
2
3 SAM CLI now collects telemetry to better understand customer needs.
4
5 You can OPT OUT and disable telemetry collection by setting the
6 environment variable SAM_CLI_TELEMETRY=0 in your shell.
7 Thanks for your help!
8
9 Learn More: https://docs.aws.amazon.com/serverless-application-model/latest/develop
10
11 Error: Template file not found at /codecatlyst/output/src3862/src/git-codecommit.us

```

Errors (1)

BuildBackend The action failed during runtime. View the action's logs for more details about the failure.

Fasi successive

Per ulteriori informazioni sui concetti relativi ai flussi di lavoro, consulta [Concetti sui flussi di lavoro](#)

Per creare il tuo primo flusso di lavoro, consulta [Guida introduttiva ai flussi di lavoro](#).

Concetti sui flussi di lavoro

Ecco alcuni concetti e termini da conoscere quando si crea, si testa o si distribuisce codice con flussi di lavoro. CodeCatalyst

Flussi di lavoro

Un flusso di lavoro è una procedura automatizzata che descrive come creare, testare e distribuire il codice come parte di un sistema di integrazione e distribuzione continua (CI/CD). Un flusso di lavoro definisce una serie di passaggi, o azioni, da eseguire durante l'esecuzione di un flusso di lavoro. Un flusso di lavoro definisce anche gli eventi, o trigger, che causano l'avvio del flusso di lavoro. Per configurare un flusso di lavoro, è necessario creare un file di definizione del flusso di lavoro utilizzando l'editor [visivo o YAML](#) della CodeCatalyst console.

 Tip

Per una rapida occhiata a come potresti utilizzare i flussi di lavoro in un progetto, [crea un progetto con un blueprint](#). Ogni blueprint implementa un flusso di lavoro funzionante che puoi rivedere, eseguire e sperimentare.

File di definizione del flusso di lavoro

Un file di definizione del flusso di lavoro è un file YAML che descrive il flusso di lavoro. Per impostazione predefinita, il file viene archiviato in una `~/ .codecatalyst/workflows/` cartella nella radice del repository di [origine](#). Il file può avere un'estensione `.yaml` o `.yml` e l'estensione deve essere minuscola.

Per ulteriori informazioni sul file di definizione del flusso di lavoro, vedere. [Definizione YAML del flusso di lavoro](#)

Azioni

Un'azione è l'elemento costitutivo principale di un flusso di lavoro e definisce un'unità logica di lavoro, o attività, da eseguire durante l'esecuzione di un flusso di lavoro. In genere, un flusso di lavoro include più azioni eseguite in sequenza o in parallelo a seconda di come le hai configurate.

Per ulteriori informazioni sulle azioni, consulta [Configurazione delle azioni del flusso di lavoro](#).

Gruppi di azione

Un gruppo di azioni contiene una o più azioni. Il raggruppamento delle azioni in gruppi di azioni consente di mantenere organizzato il flusso di lavoro e consente inoltre di configurare le dipendenze tra diversi gruppi.

Per ulteriori informazioni sui gruppi di azioni, consulta. [Raggruppamento delle azioni in gruppi di azione](#)

Artifacts

Un artefatto è l'output di un'azione del flusso di lavoro e in genere è costituito da una cartella o da un archivio di file. Gli artefatti sono importanti perché consentono di condividere file e informazioni tra le azioni.

Per ulteriori informazioni sugli artefatti, vedi [Condivisione di artefatti e file tra azioni](#).

Calcolo

Compute si riferisce al motore di elaborazione (CPU, memoria e sistema operativo) gestito e mantenuto da CodeCatalyst per eseguire le azioni del flusso di lavoro.

Per ulteriori informazioni sull'elaborazione, vedere. [Configurazione delle immagini di calcolo e di runtime](#)

Ambienti

Un CodeCatalyst ambiente, da non confondere con un [ambiente di sviluppo](#), definisce l'Amazon VPC di destinazione Account AWS e opzionale a cui si connette un CodeCatalyst [flusso di lavoro](#). Un ambiente definisce anche il [ruolo IAM](#) necessario a un flusso di lavoro per accedere ai AWS servizi e alle risorse all'interno dell'account di destinazione.

È possibile configurare più ambienti e assegnare loro nomi come sviluppo, test, staging e produzione. Quando si esegue la distribuzione in questi ambienti, le informazioni sulle distribuzioni vengono visualizzate nelle schede CodeCatalyst Deployment activity e Deployment targets dell'ambiente.

Per ulteriori informazioni sugli ambienti, vedere. [Implementazione in e Account AWS VPCs](#)

Cancelli

Un gate è un componente del flusso di lavoro che è possibile utilizzare per impedire che l'esecuzione di un flusso di lavoro proceda a meno che non vengano soddisfatte determinate condizioni. Un esempio di gate è l'Approval gate, in cui gli utenti devono inviare un'approvazione nella CodeCatalyst console prima che l'esecuzione del flusso di lavoro possa continuare.

È possibile aggiungere porte tra sequenze di azioni in un flusso di lavoro o prima della prima azione (che viene eseguita immediatamente dopo il download del codice sorgente). Puoi anche aggiungere porte dopo l'ultima azione, se necessario.

Per ulteriori informazioni sui cancelli, vedere [Garantire l'esecuzione di un flusso di lavoro](#).

Report

Un rapporto contiene dettagli sui test che si verificano durante l'esecuzione di un flusso di lavoro. È possibile creare report come un rapporto di test, un rapporto sulla copertura del codice, un rapporto di

analisi della composizione del software e un rapporto di analisi statica. È possibile utilizzare un report per aiutare a risolvere un problema durante un flusso di lavoro. Se disponi di molti report provenienti da più flussi di lavoro, puoi utilizzarli per visualizzare tendenze e tassi di errore per ottimizzare le applicazioni e le configurazioni di distribuzione.

Per ulteriori informazioni sui report, consulta [Tipi di report sulla qualità](#)

Esecuzioni

Un'esecuzione è una singola iterazione di un flusso di lavoro. Durante un'esecuzione, CodeCatalyst esegue le azioni definite nel file di configurazione del flusso di lavoro e restituisce i log, gli artefatti e le variabili associati.

Per ulteriori informazioni sulle esecuzioni, vedere [Esecuzione di un workflow](#)

Origini

Una sorgente, chiamata anche sorgente di input, è un archivio di sorgenti a cui si connette [un'azione del flusso](#) di lavoro per ottenere i file necessari per eseguire le proprie operazioni. Ad esempio, un'azione del flusso di lavoro potrebbe connettersi a un repository di origine per ottenere i file di origine dell'applicazione per creare un'applicazione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Variabili

Una variabile è una coppia chiave-valore che contiene informazioni a cui puoi fare riferimento nel tuo flusso di lavoro Amazon CodeCatalyst. La parte relativa al valore della variabile viene sostituita con un valore effettivo durante l'esecuzione del flusso di lavoro.

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Trigger del flusso di lavoro

Un trigger del flusso di lavoro, o semplicemente un trigger, consente di avviare un flusso di lavoro eseguito automaticamente quando si verificano determinati eventi, ad esempio l'invio di un codice. Potresti voler configurare i trigger per evitare agli sviluppatori di software di dover avviare manualmente il flusso di lavoro tramite la CodeCatalyst console.

Puoi utilizzare tre tipi di trigger:

- **Push:** un trigger di invio di codice provoca l'avvio di un flusso di lavoro ogni volta che viene premuto un commit.
- **Pull request:** un trigger di pull request fa sì che un workflow venga avviato ogni volta che una pull request viene creata, rivista o chiusa.
- **Pianificazione:** un trigger di pianificazione fa sì che l'esecuzione di un workflow venga avviata in base a una pianificazione definita dall'utente. Prendi in considerazione l'utilizzo di un trigger di pianificazione per eseguire le build notturne del tuo software in modo che la build più recente sia pronta per essere utilizzata dagli sviluppatori il mattino successivo.

Puoi utilizzare i trigger push, pull request e schedule da soli o in combinazione nello stesso flusso di lavoro.

I trigger sono facoltativi: se non ne configuri nessuno, puoi avviare un flusso di lavoro solo manualmente.

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Guida introduttiva ai flussi di lavoro

In questo tutorial imparerai come creare e configurare il tuo primo flusso di lavoro.

Tip

Preferisci iniziare con un flusso di lavoro preconfigurato? Vedi [Creare un progetto con un blueprint](#), che include istruzioni per configurare un progetto con un flusso di lavoro funzionante, un'applicazione di esempio e altre risorse.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Crea e configura il tuo flusso di lavoro](#)
- [Fase 2: Salvate il flusso di lavoro con un commit](#)
- [Fase 3: Visualizza i risultati della corsa](#)
- [\(Opzionale\) Fase 4: pulizia](#)

Prerequisiti

Prima di iniziare:

- Hai bisogno di uno CodeCatalyst spazio. Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Nel tuo CodeCatalyst spazio, hai bisogno di un progetto vuoto chiamato:

```
codecatalyst-project
```

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

- Nel tuo progetto, hai bisogno di un CodeCatalyst repository chiamato:

```
codecatalyst-source-repository
```

Per ulteriori informazioni, consulta [Creazione di un repository di sorgenti](#).

Note

Se hai un progetto e un repository di sorgenti esistenti, puoi usarli; tuttavia, crearne di nuovi semplifica la pulizia alla fine di questo tutorial.

Fase 1: Crea e configura il tuo flusso di lavoro

In questo passaggio, crei e configuri un flusso di lavoro che crea e verifica automaticamente il codice sorgente quando vengono apportate modifiche.

Per creare il tuo flusso di lavoro

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli Crea flusso di lavoro.

Il file di definizione del flusso di lavoro viene visualizzato nell' CodeCatalyst editor YAML della console.

Per configurare il flusso di lavoro

Puoi configurare il tuo flusso di lavoro nell'editor visuale o nell'editor YAML. Cominciamo con l'editor YAML e poi passiamo all'editor visuale.

1. Scegli **+ Azioni** per visualizzare un elenco di azioni del flusso di lavoro che puoi aggiungere al tuo flusso di lavoro.
2. Nell'azione **Crea**, scegli **+** per aggiungere lo YAML dell'azione al file di definizione del flusso di lavoro. Il tuo flusso di lavoro ora è simile al seguente.

```
Name: Workflow_fe47
SchemaVersion: "1.0"

# Optional - Set automatic triggers.
Triggers:
  - Type: Push
    Branches:
      - main

# Required - Define action configurations.
Actions:
  Build_f0:
    Identifier: aws/build@v1

    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this workflow as
a source

    Outputs:
      AutoDiscoverReports:
        Enabled: true
        # Use as prefix for the report files
        ReportNamePrefix: rpt

    Configuration:
      Steps:
        - Run: echo "Hello, World!"
        - Run: echo "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>\" >> report.xml
report.xml
        - Run: echo "<testsuite tests=\"1\" name=\"TestAgentJunit\" >\" >>
report.xml
        - Run: echo "<testcase classname=\"TestAgentJunit\" name=\"Dummy
```

```
Test\"/></testsuite>" >> report.xml
```

Il flusso di lavoro copia i file nel repository di **WorkflowSource** origine sulla macchina di elaborazione che esegue l'**Build_f0**azione, stampa **Hello, World!** nei log, scopre i report dei test sulla macchina di elaborazione e li invia alla pagina Report della console. CodeCatalyst

3. Scegli Visual per visualizzare il file di definizione del flusso di lavoro nell'editor visuale. I campi dell'editor visuale consentono di configurare le proprietà YAML mostrate nell'editor YAML.

Fase 2: Salvate il flusso di lavoro con un commit

In questo passaggio, salvi le modifiche. Poiché i flussi di lavoro vengono archiviati come `.yaml` file nel repository, le modifiche vengono salvate con dei commit.

Per confermare le modifiche al flusso di lavoro

1. (Facoltativo) Scegli Convalida per assicurarti che il codice YAML del flusso di lavoro sia valido.
2. Scegli Applica.
3. In Nome file del flusso di lavoro, inserisci un nome per il file di configurazione del flusso di lavoro, ad esempio. **my-first-workflow**
4. In Commit message, inserisci un messaggio per identificare il tuo commit, ad esempio **create my-first-workflow.yaml**.
5. In Repository, scegli il repository in cui vuoi salvare il flusso di lavoro (`codecatalyst-repository`).
6. In Nome del ramo, scegli il ramo in cui vuoi salvare il flusso di lavoro (`main`).
7. Scegli Applica.

Il nuovo flusso di lavoro viene visualizzato nell'elenco dei flussi di lavoro. La visualizzazione potrebbe richiedere alcuni istanti.

Poiché i flussi di lavoro vengono salvati con commit e poiché il flusso di lavoro ha un trigger code push configurato, il salvataggio del flusso di lavoro avvia un flusso di lavoro eseguito automaticamente.

Fase 3: Visualizza i risultati della corsa

In questo passaggio, accedi alla corsa che è stata avviata dal tuo commit e visualizzi i risultati.

Per visualizzare i risultati della corsa

1. Scegli il nome del tuo flusso di lavoro, ad esempio, `Workflow_fe47`.

Un diagramma del flusso di lavoro che mostra l'etichetta del repository di origine (WorkflowSource) e l'azione di creazione (ad esempio, `build_F0`).

2. Nel diagramma di esecuzione del flusso di lavoro, scegliete l'azione di compilazione (ad esempio, `build_F0`).
3. Esamina il contenuto delle schede Registri, Rapporti, Configurazione e Variabili. Queste schede mostrano i risultati dell'azione di compilazione.

Per ulteriori informazioni, consulta [Visualizzazione dei risultati di un'azione di compilazione](#).

(Opzionale) Fase 4: pulizia

In questo passaggio, pulisci le risorse che hai creato in questo tutorial.

Per eliminare le risorse

- Se hai creato un nuovo progetto per questo tutorial, eliminalo. Per istruzioni, consulta [Eliminazione di un progetto](#). L'eliminazione del progetto elimina anche l'archivio di origine e il flusso di lavoro.

Creazione con flussi di lavoro

Utilizzando [CodeCatalyst i flussi di lavoro](#), puoi creare applicazioni e altre risorse.

Argomenti

- [Come posso creare un'applicazione?](#)
- [Vantaggi dell'azione di compilazione](#)
- [Alternative all'azione di compilazione](#)
- [Aggiungere l'azione di compilazione](#)
- [Visualizzazione dei risultati di un'azione di compilazione](#)
- [Tutorial: caricare artefatti su Amazon S3](#)
- [Crea e testa azioni YAML](#)

Come posso creare un'applicazione?

Per creare un'applicazione o una risorsa CodeCatalyst, devi prima creare un flusso di lavoro, quindi specificare un'azione di compilazione al suo interno.

Un'azione di compilazione è un elemento costitutivo del flusso di lavoro che compila il codice sorgente, esegue test unitari e produce artefatti pronti per la distribuzione.

Puoi aggiungere un'azione di compilazione al tuo flusso di lavoro utilizzando l'editor visivo o l'editor YAML della CodeCatalyst console.

I passaggi di alto livello per creare un'applicazione o una risorsa sono i seguenti.

Per creare un'applicazione (attività di alto livello)

1. In CodeCatalyst, aggiungi il codice sorgente per un'applicazione che desideri creare. Per ulteriori informazioni, consulta [Memorizzazione del codice sorgente nei repository per un progetto in CodeCatalyst](#).
2. Nel CodeCatalyst, crei un flusso di lavoro. Il flusso di lavoro è il luogo in cui si definisce come creare, testare e distribuire l'applicazione. Per ulteriori informazioni, consulta [Guida introduttiva ai flussi di lavoro](#).
3. (Facoltativo) Nel flusso di lavoro, aggiungete un trigger che indica gli eventi che causeranno l'avvio automatico del flusso di lavoro. Per ulteriori informazioni, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#)
4. Nel flusso di lavoro, aggiungi un'azione di compilazione che compila e impacchetta il codice sorgente dell'applicazione o della risorsa. Facoltativamente, puoi anche fare in modo che l'azione di compilazione esegua test unitari, generi report e distribuisca l'applicazione se non desideri utilizzare un'azione di test o di distribuzione per questi scopi. Per ulteriori informazioni sulle azioni di test e distribuzione, consulta [Aggiungere l'azione di compilazione](#)
5. (Facoltativo) Nel flusso di lavoro, aggiungi un'azione di test e un'azione di distribuzione per testare e distribuire l'applicazione o la risorsa. Puoi scegliere tra diverse azioni preconfigurate per distribuire la tua applicazione su destinazioni diverse, come Amazon ECS. Per ulteriori informazioni, consultare [Test con flussi di lavoro](#) e [Implementazione con flussi di lavoro](#).
6. Il flusso di lavoro viene avviato manualmente o automaticamente tramite un trigger. Il flusso di lavoro esegue le azioni di compilazione, test e distribuzione in sequenza per creare, testare e distribuire l'applicazione e le risorse sulla destinazione. Per ulteriori informazioni, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Vantaggi dell'azione di compilazione

L'utilizzo dell'azione di compilazione all'interno di un flusso di lavoro offre i seguenti vantaggi:

- Completamente gestito: l'azione di compilazione elimina la necessità di configurare, applicare patch, aggiornare e gestire i propri server di build.
- Su richiesta: l'azione di compilazione è scalabile su richiesta per soddisfare le tue esigenze di compilazione. Paghi soltanto per il tempo di compilazione utilizzato. Per ulteriori informazioni, consulta [Configurazione delle immagini di calcolo e di runtime](#).
- Pronto all'uso: CodeCatalyst include immagini Docker preconfezionate in ambiente di runtime che vengono utilizzate per eseguire tutte le azioni del flusso di lavoro, comprese le azioni di compilazione. Queste immagini sono preconfigurate con strumenti utili per la creazione di applicazioni come e Node.js. AWS CLI È possibile CodeCatalyst configurare l'utilizzo di un'immagine di build fornita da un registro pubblico o privato. Per ulteriori informazioni, consulta [Specificazione delle immagini dell'ambiente di runtime](#).

Alternative all'azione di compilazione

Se utilizzi un'azione di compilazione per distribuire la tua applicazione, prendi in considerazione invece l'utilizzo di un'azione di CodeCatalyst distribuzione. Deploy Actions Esegui una behind-the-scenes configurazione che altrimenti dovresti scrivere manualmente se utilizzi un'azione di compilazione. Per ulteriori informazioni sulle azioni di distribuzione disponibili, consulta. [Elenco delle azioni di distribuzione](#)

Puoi anche utilizzarlo AWS CodeBuild per creare le tue applicazioni. Per ulteriori informazioni, consulta la pagina [Che cos'è CodeBuild?](#)

Aggiungere l'azione di compilazione

Utilizza la procedura seguente per aggiungere un'azione di compilazione al tuo CodeCatalyst flusso di lavoro.

Visual

Per aggiungere un'azione di compilazione utilizzando l'editor visuale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli Visual.
6. Scegli Azioni.
7. In Azioni, scegli Costruisci.
8. Nelle schede Input e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, consulta. [Crea e testa azioni YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e sul valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere un'azione di compilazione utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli YAML.
6. Scegli Azioni.
7. In Azioni, scegli Costruisci.
8. Modifica le proprietà nel codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [Crea e testa azioni YAML](#)
9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Crea la definizione dell'azione

L'azione di compilazione è definita come un insieme di proprietà YAML all'interno del file di definizione del flusso di lavoro. Per informazioni su queste proprietà, [Crea e testa azioni YAML](#) consulta. [Definizione YAML del flusso di lavoro](#)

Visualizzazione dei risultati di un'azione di compilazione

Utilizza le seguenti istruzioni per visualizzare i risultati di un'azione di compilazione, inclusi i log, i report e le variabili generati.

Per visualizzare i risultati di un'azione di compilazione

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
3. Nel diagramma del flusso di lavoro, scegli il nome dell'azione di creazione, ad esempio Crea.
4. Per visualizzare i log per l'esecuzione della build, scegli Logs. Vengono visualizzati i log delle varie fasi di compilazione. È possibile espandere o comprimere i registri in base alle esigenze.
5. Per visualizzare i report di test prodotti dall'azione di compilazione, scegli Report o nel riquadro di navigazione scegli Report. Per ulteriori informazioni, consulta [Tipi di report sulla qualità](#).
6. Per visualizzare la configurazione utilizzata per l'azione di compilazione, scegli Configurazione. Per ulteriori informazioni, consulta [Aggiungere l'azione di compilazione](#).
7. Per visualizzare le variabili utilizzate dall'azione di compilazione, scegliete Variabili. Per ulteriori informazioni, consulta [Utilizzo delle variabili nei flussi di lavoro](#).

Tutorial: caricare artefatti su Amazon S3

[In questo tutorial, imparerai come caricare artefatti in un bucket Amazon S3 utilizzando un flusso di lavoro CodeCatalyst Amazon che include un paio di azioni di compilazione.](#) Queste azioni vengono eseguite in serie all'avvio del flusso di lavoro. La prima azione di compilazione genera due file `Hello.txt` e `Goodbye.txt` li raggruppa in un elemento di compilazione. La seconda azione di compilazione carica l'artefatto su Amazon S3. Configurerai il flusso di lavoro in modo che venga eseguito ogni volta che invii un commit al tuo repository di origine.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un AWS ruolo](#)
- [Fase 2: creare un bucket Amazon S3](#)
- [Fase 3: Creare un archivio di sorgenti](#)
- [Fase 4: Creare un flusso di lavoro](#)
- [Passaggio 5: verifica i risultati](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare, avrai bisogno di:

- È necessario uno CodeCatalyst spazio con un AWS account connesso. Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Nel tuo spazio, hai bisogno di un progetto vuoto chiamato:

```
codecatalyst-artifact-project
```

Usa l'opzione Inizia da zero per creare questo progetto.

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

- Nel tuo progetto, hai bisogno di un CodeCatalyst ambiente chiamato:

```
codecatalyst-artifact-environment
```

Configura questo ambiente come segue:

- Scegli qualsiasi tipo, ad esempio Development.
- Connect il tuo AWS account.
- Per il ruolo IAM predefinito, scegli un ruolo qualsiasi. Specificherai un ruolo diverso in seguito.

Per ulteriori informazioni, consulta [Implementazione in e Account AWS VPCs](#).

Fase 1: Creare un AWS ruolo

In questo passaggio, crei un ruolo AWS IAM che successivamente assegnerai all'azione di compilazione nel tuo flusso di lavoro. Questo ruolo concede all'azione di CodeCatalyst compilazione l'autorizzazione ad accedere al tuo AWS account e scrivere su Amazon S3 dove verrà archiviato il tuo artefatto. Il ruolo è chiamato ruolo Build.

Note

Se hai già un ruolo di costruzione che hai creato per un altro tutorial, puoi utilizzarlo anche per questo tutorial. Assicurati solo che disponga delle autorizzazioni e della politica di fiducia mostrate nella procedura seguente.

Per ulteriori informazioni sui ruoli IAM, consulta i [ruoli IAM](#) nella Guida per l'AWS Identity and Access Management utente.

Per creare un ruolo di costruzione

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - c. Nel riquadro di navigazione, scegli Policy.
 - d. Scegli Create Policy (Crea policy).
 - e. Scegliere la scheda JSON.
 - f. Eliminare il codice esistente.
 - g. Incolla il codice seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
```

```

        "s3:PutObject",
        "s3:ListBucket"
    ],
    "Resource": "*"
}
]
}

```

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-s3-build-policy
```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di costruzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione, cerca **codecatalyst-s3-build-policy** e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).

- h. Per il nome del ruolo, inserisci:

codecatalyst-s3-build-role

- i. Per la descrizione del ruolo, inserisci:

CodeCatalyst build role

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di fiducia e una politica di autorizzazioni.

Fase 2: creare un bucket Amazon S3

In questo passaggio, crei un bucket Amazon S3 in cui verranno caricati gli elementi `Hello.txt` e gli `Goodbye.txt` artefatti.

Come creare un bucket Amazon S3.

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. Nel riquadro principale, scegli Crea bucket.
3. Per il nome del bucket, inserisci:

codecatalyst-artifact-bucket

4. In AWS Region (Regione), scegliere una Regione. Questo tutorial presuppone che tu abbia scelto US West (Oregon) us-west-2. Per informazioni sulle regioni supportate da Amazon S3, consulta gli [endpoint e le quote di Amazon Simple Storage Service](#) nel. Riferimenti generali di AWS
5. Nella parte inferiore della pagina, scegli Crea bucket.
6. Copia il nome del bucket che hai appena creato, ad esempio:

codecatalyst-artifact-bucket

Ora hai creato un bucket chiamato **codecatalyst-artifact-bucket** nella regione us-west-2 degli Stati Uniti occidentali (Oregon).

Fase 3: Creare un archivio di sorgenti

In questo passaggio, creerai un repository di origine in CodeCatalyst. Questo repository viene utilizzato per archiviare il file di definizione del flusso di lavoro del tutorial.

Per ulteriori informazioni sugli archivi di origine, vedere [Creazione di un repository di sorgenti](#)

Per creare un archivio dei sorgenti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto, `codecatalyst-artifact-project`.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Crea repository.
5. Nel nome del repository, inserisci:

`codecatalyst-artifact-source-repository`

6. Scegli Create (Crea).

Ora hai creato un repository chiamato `codecatalyst-artifact-source-repository`

Fase 4: Creare un flusso di lavoro

In questo passaggio, crei un flusso di lavoro costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando si invia una modifica all'archivio di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#)
- Un'azione di compilazione chiamata `GenerateFiles`: al momento del trigger, l'azione `GenerateFiles` crea due file `Hello.txt` e `Goodbye.txt` li impacchetta in un elemento di output chiamato `codecatalystArtifact`
- Un'altra azione di compilazione chiamata `Upload`: al termine dell'azione `GenerateFiles`, l'azione `Upload` esegue il comando `aws s3 sync` per caricare i file nel `codecatalystArtifact` e nel repository di origine nel bucket Amazon S3. AWS CLI Viene fornito preinstallato e preconfigurato sulla piattaforma di CodeCatalyst elaborazione, quindi non è necessario installarlo o configurarlo.

Per ulteriori informazioni sul software preconfezionato sulla piattaforma di elaborazione, consulta CodeCatalyst . [Specificazione delle immagini dell'ambiente di runtime](#) Per ulteriori informazioni sul `aws s3 sync` comando AWS CLI's, consulta [sync](#) nel Command Reference.AWS CLI

Per ulteriori informazioni sull'azione di compilazione, consulta [Creazione con flussi di lavoro](#).

Come creare un flusso di lavoro

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli Crea flusso di lavoro.
3. Elimina il codice di esempio YAML.
4. Aggiungi il seguente codice YAML:

 Note

Nel codice YAML che segue, puoi omettere la sezione se lo `Connections`: desideri. Se ometti questa sezione, devi assicurarti che il ruolo specificato nel campo `Default IAM role` nel tuo ambiente includa le autorizzazioni e le politiche di fiducia descritte in. [Fase 1: Creare un AWS ruolo](#) Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta. [Creazione di un ambiente](#)

```
Name: codecatalyst-artifact-workflow
SchemaVersion: 1.0

Triggers:
  - Type: Push
    Branches:
      - main
Actions:
  GenerateFiles:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        # Create the output files.
        - Run: echo "Hello, World!" > "Hello.txt"
        - Run: echo "Goodbye!" > "Goodbye.txt"
    Outputs:
```

```
Artifacts:
  - Name: codecatalystArtifact
    Files:
      - "**/*"
Upload:
  Identifier: aws/build@v1
  DependsOn:
    - GenerateFiles
  Environment:
    Name: codecatalyst-artifact-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-s3-build-role
  Inputs:
    Artifacts:
      - codecatalystArtifact
  Configuration:
    Steps:
      # Upload the output artifact to the S3 bucket.
      - Run: aws s3 sync . s3://codecatalyst-artifact-bucket
```

Nel codice precedente, sostituisci:

- *codecatalyst-artifact-environment* con il nome dell'ambiente in cui hai creato [Prerequisiti](#).
- *codecatalyst-account-connection* con il nome della connessione all'account in cui hai creato [Prerequisiti](#).
- *codecatalyst-s3-build-role* con il nome del ruolo di build in cui hai creato [Fase 1: Creare un AWS ruolo](#).
- *codecatalyst-artifact-bucket* con il nome dell'Amazon S3 in cui hai creato. [Fase 2: creare un bucket Amazon S3](#)

Per informazioni sulle proprietà di questo file, consulta [Crea e testa azioni YAML](#).

5. (Facoltativo) Scegliete Convalida per assicurarvi che il codice YAML sia valido prima di eseguire il commit.
6. Scegli Applica.
7. Nella finestra di dialogo Conferma flusso di lavoro, inserisci quanto segue:

- a. Per il nome del file Workflow, lasciate il valore predefinito, `codecatalyst-artifact-workflow`.
- b. Per il messaggio di conferma, inserisci:

```
add initial workflow file
```

- c. Per Repository, scegli `codecatalyst-artifact-source-repository`.
- d. Per il nome del ramo, scegli principale.
- e. Scegli Applica.

Ora hai creato un flusso di lavoro. L'esecuzione di un flusso di lavoro viene avviata automaticamente a causa del trigger definito nella parte superiore del flusso di lavoro. In particolare, quando hai eseguito il commit (e inviato) del `codecatalyst-artifact-workflow.yaml` file nell'archivio di origine, il trigger ha avviato l'esecuzione del flusso di lavoro.

Per visualizzare il flusso di lavoro in corso

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il flusso di lavoro che hai appena creato: `codecatalyst-artifact-workflow`
3. Scegli `GenerateFiles` di vedere lo stato di avanzamento della prima azione di costruzione.
4. Scegli `Carica` per vedere lo stato di avanzamento della seconda azione di build.
5. Al termine dell'azione di caricamento, procedi come segue:
 - Se l'esecuzione del flusso di lavoro è riuscita, passate alla procedura successiva.
 - Se l'esecuzione del flusso di lavoro non è riuscita, scegli Registri per risolvere il problema.

Passaggio 5: verifica i risultati

Dopo l'esecuzione del flusso di lavoro, accedi al servizio Amazon S3 e cerca nel tuo *codecatalyst-artifact-bucket* bucket. Ora dovrebbe includere i seguenti file e cartelle:

```
.
|- .aws/
|- .git/
|Goodbye.txt
|Hello.txt
```

`| README.md`

I `Hello.txt` file `Goodbye.txt` e sono stati caricati perché facevano parte dell'`codecatalystArtifact` artefatto. I `README.md` file `.aws/.git/`, e sono stati caricati perché erano presenti nel repository di origine.

Eliminazione

Effettua la pulizia CodeCatalyst ed AWS evita che questi servizi vengano addebitati.

Per pulire CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Eliminare il repository di `codecatalyst-artifact-source-repository` origine.
3. Eliminare il `codecatalyst-artifact-workflow` flusso di lavoro.

Per ripulire AWS

1. Esegui la pulizia in Amazon S3, come segue:
 - a. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
 - b. Elimina i file nel `codecatalyst-artifact-bucket` bucket.
 - c. Eliminare il `codecatalyst-artifact-bucket` bucket.
2. Esegui la pulizia in IAM, come segue:
 - a. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - b. Eliminare il `codecatalyst-s3-build-policy`.
 - c. Eliminare il `codecatalyst-s3-build-role`.

Crea e testa azioni YAML

Di seguito è riportata la definizione YAML delle azioni di compilazione e test. Esiste un riferimento per due azioni perché le loro proprietà YAML sono molto simili.

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Scegliete una proprietà YAML nel codice seguente per vederne una descrizione.

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visuale. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
action-name:
  Identifier: aws/build@v1 | aws/managed-test@v1
  DependsOn:
    - dependent-action-name-1
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
  Caching:
    FileCaching:
      key-name-1:
        Path: file1.txt
      RestoreKeys:
        - restore-key-1
  Inputs:
    Sources:
      - source-name-1
      - source-name-2
    Artifacts:
      - artifact-name
```

Variables:

- Name: *variable-name-1*
Value: *variable-value-1*
- Name: *variable-name-2*
Value: *variable-value-2*

Outputs:Artifacts:

- Name: *output-artifact-1*

Files:

- build-output/artifact-1.jar
- "build-output/build*"

- Name: *output-artifact-2*

Files:

- build-output/artifact-2.1.jar
- build-output/artifact-2.2.jar

Variables:

- *variable-name-1*
- *variable-name-2*

AutoDiscoverReports:Enabled: *true | false*ReportNamePrefix: *AutoDiscovered*IncludePaths:

- ***/**

ExcludePaths:

- *node_modules/cdk/junit.xml*

SuccessCriteria:PassRate: *percent*LineCoverage: *percent*BranchCoverage: *percent*Vulnerabilities:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisBug:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisSecurity:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisQuality:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisFinding:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*

Reports:report-name-1:Format: *format*IncludePaths:

- *"*.xml"*

ExcludePaths:

- *report2.xml*
- *report3.xml*

SuccessCriteria:PassRate: *percent*LineCoverage: *percent*BranchCoverage: *percent*Vulnerabilities:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisBug:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisSecurity:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisQuality:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*StaticAnalysisFinding:Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*Number: *whole-number*Configuration:Container:Registry: *registry*Image: *image*Steps:

- Run: *"step 1"*
- Run: *"step 2"*

Packages:NpmConfiguration:PackageRegistries:

- PackageRepository: *package-repository*

Scopes:

- *"@scope"*

ExportAuthorizationToken: *true | false*

nome dell'azione

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Interfaccia utente corrispondente: scheda di configurazione/nome dell'azione

Identifier

(*action-name*/Identifier)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Da utilizzare `aws/build@v1` per le azioni di compilazione.

Utilizzare `aws/managed-test@v1` per azioni di test.

Interfaccia utente corrispondente: diagramma *aws/build@v1* | *aws/managed-test@v1* del workflow, nome dell'azione, etichetta

DependsOn

(*action-name*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente per l'esecuzione di questa azione.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Ingressi/Dipende da - opzionale

Compute

(*action-name*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*action-name*/Compute/Tipo)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: scheda di configurazione/tipo di calcolo

Fleet

(*action-name*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand:,. Linux.x86-64.Large Linux.x86-64.XLarge Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omesso, l'impostazione predefinita è `Linux.x86-64.Large`

Interfaccia utente corrispondente: scheda di configurazione/parco di calcolo

Timeout

(*action-name*/Timeout)

(Facoltativo)

Specificate il periodo di tempo, in minuti (editor YAML) o ore e minuti (editor visivo), che l'azione può essere eseguita prima CodeCatalyst che termini l'azione. Il minimo è 5 minuti e il massimo è descritto in [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Environment

(*action-name*/Environment)

(Facoltativo)

Specificare l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs e Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*action-name*/Environment/Name)

(Facoltativo)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*action-name*/Environment/Connections)

(Facoltativo)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà `Role` nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: tab/Environment/What la configurazione è attiva *my-environment*? /menu a tre punti/ Cambia ruolo

Name

(*action-name*/Environment/Connections/Name)

(Obbligatorio se incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: tab/Environment/What la configurazione è attiva *my-environment*? /menu a tre punti/ Cambia ruolo

Role

(*action-name*/Environment/Connections/Role)

(Obbligatorio se incluso [Connections](#))

Specificate il nome del ruolo IAM utilizzato da questa azione per accedere e operare in AWS servizi come Amazon S3 e Amazon ECR. Assicurati che questo ruolo venga aggiunto alla tua Account AWS connessione nel tuo spazio. Per aggiungere un ruolo IAM alla connessione di un account, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

Note

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Warning

Limita le autorizzazioni a quelle richieste dalle azioni di compilazione e test. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

Interfaccia utente corrispondente: la configurazione tab/Environment/What è attiva? *my-environment* /menu a tre punti/ Cambia ruolo

Caching

(*action-name*/Caching)

(Facoltativo)

Una sezione in cui è possibile specificare una cache per salvare i file su disco e ripristinarli da tale cache nelle successive esecuzioni del flusso di lavoro.

Per ulteriori informazioni sulla memorizzazione nella cache dei file, vedere. [Memorizzazione nella cache dei file tra le esecuzioni del workflow](#)

Interfaccia utente corrispondente: scheda di configurazione/memorizzazione nella cache dei file - opzionale

FileCaching

(*action-name*/Caching/FileCaching)

(Facoltativo)

Una sezione che specifica la configurazione per una sequenza di cache.

Interfaccia utente corrispondente: memorizzazione nella tab/File cache della configurazione - facoltativa/Aggiungi cache

nome-chiave 1

(*action-name*/Caching/FileCaching/*key-name-1*)

(Facoltativo)

Specificate il nome della proprietà della cache principale. I nomi delle proprietà della cache devono essere univoci all'interno del flusso di lavoro. Per ogni azione possono essere inserite fino a cinque vociFileCaching.

Interfaccia utente corrispondente: tab/File memorizzazione nella cache della configurazione - Opzionale/Aggiungi cache/ Chiave

Path

(*action-name*/Caching/FileCaching/*key-name-1*/Path)

(Facoltativo)

Specificate il percorso associato per la cache.

Interfaccia utente corrispondente: tab/File memorizzazione nella cache della configurazione - Opzionale/Aggiungi cache/percorso

RestoreKeys

(*action-name*/Caching/FileCaching/*key-name-1*/RestoreKeys)

(Facoltativo)

Specificare la chiave di ripristino da utilizzare come riserva quando non è possibile trovare la proprietà della cache principale. I nomi delle chiavi di ripristino devono essere univoci all'interno del flusso di lavoro. Ogni cache può contenere fino a cinque voci RestoreKeys.

Interfaccia utente corrispondente: memorizzazione tab/File nella cache della configurazione - Opzionale/Aggiungi cache/chiavi di ripristino - opzionale

Inputs

(*action-name*/Inputs)

(Facoltativo)

La Inputs sezione definisce i dati necessari a un'azione durante l'esecuzione di un flusso di lavoro.

Note

Sono consentiti un massimo di quattro input (una fonte e tre artefatti) per azione di compilazione o azione di test. Le variabili non vengono conteggiate per questo totale.

Se è necessario fare riferimento a file che risiedono in input diversi (ad esempio una sorgente e un artefatto), l'input di origine è l'input principale e l'artefatto è l'input secondario. I riferimenti ai file negli input secondari richiedono un prefisso speciale per distinguerli dal primario. Per informazioni dettagliate, vedi [Esempio: riferimento a file in più artefatti](#).

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*action-name*/Inputs/Sources)

(Facoltativo)

Specificate le etichette che rappresentano i repository di origine che saranno necessari all'azione. Attualmente, l'unica etichetta supportata è `WorkflowSource`, che rappresenta l'archivio di origine in cui è archiviato il file di definizione del flusso di lavoro.

Se si omette una fonte, è necessario specificare almeno un elemento di input in. `action-name/Inputs/Artifacts`

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: nessuna

Artifacts - input

`(action-name/Inputs/Artifacts)`

(Facoltativo)

Specificate gli artefatti delle azioni precedenti che desiderate fornire come input per questa azione. Questi artefatti devono essere già definiti come artefatti di output nelle azioni precedenti.

Se non si specifica alcun artefatto di input, è necessario specificare almeno un repository di origine in. `action-name/Inputs/Sources`

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Note

Se l'elenco a discesa Artifacts - opzionale non è disponibile (editor visivo) o se si verificano errori durante la convalida di YAML (editor YAML), è possibile che l'azione supporti un solo input. In questo caso, prova a rimuovere l'input sorgente.

Interfaccia utente corrispondente: scheda Ingressi/Artefatti - opzionale

Variables - input

`(action-name/Inputs/Variables)`

(Facoltativo)

Specificate una sequenza di name/value coppie che definisca le variabili di input che desiderate rendere disponibili all'azione. I nomi delle variabili sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle variabili.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Variabili - opzionale

Outputs

(*action-name*/Outputs)

(Facoltativo)

Definisce i dati emessi dall'azione durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Output

Artifacts - output

(*action-name*/Outputs/Artifacts)

(Facoltativo)

Specificate il nome di un artefatto generato dall'azione. I nomi degli Artifact devono essere univoci all'interno di un flusso di lavoro e sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9) e caratteri di sottolineatura (_). Spazi, trattini (-) e altri caratteri speciali non sono consentiti. Non è possibile utilizzare le virgolette per abilitare spazi, trattini e altri caratteri speciali nei nomi degli artefatti di output.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Outputs/Artifacts

Name

(*action-name*/Outputs/Artifacts/Name)

(Richiesto se incluso) [Artifacts - output](#)

Specificate il nome di un artefatto generato dall'azione. I nomi degli Artifact devono essere univoci all'interno di un flusso di lavoro e sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9) e caratteri

di sottolineatura (_). Spazi, trattini (-) e altri caratteri speciali non sono consentiti. Non è possibile utilizzare le virgolette per abilitare spazi, trattini e altri caratteri speciali nei nomi degli artefatti di output.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: restituisce il nome dell'output/build dell'tab/Artifacts/Newartefatto

Files

(*action-name*/Outputs/Artifacts/Files)

(Obbligatorio se incluso) [Artifacts - output](#)

Specificate i file CodeCatalyst inclusi nell'artefatto generato dall'azione. Questi file vengono generati dall'azione del flusso di lavoro durante l'esecuzione e sono disponibili anche nell'archivio di origine. I percorsi dei file possono risiedere in un repository di origine o in un elemento di un'azione precedente e sono relativi al repository di origine o alla radice dell'elemento. È possibile utilizzare modelli a globo per specificare i percorsi. Esempi:

- Per specificare un singolo file che si trova nella radice della posizione di creazione o della posizione del repository di origine, usa `my-file.jar`
- Per specificare un singolo file in una sottodirectory, usa `directory/my-file.jar` o `directory/subdirectory/my-file.jar`
- Per specificare tutti i file, utilizzare `**/*`. Il pattern a `**` glob indica che deve corrispondere a un numero qualsiasi di sottodirectory.
- Per specificare tutti i file e le directory in una directory denominata, usa `directory` `"directory/**/*"` Il pattern a `**` glob indica che deve corrispondere a un numero qualsiasi di sottodirectory.
- Per specificare tutti i file in una directory denominata `directory`, ma non nessuna delle sue sottodirectory, usa `"directory/*"`

Note

Se il percorso del file include uno o più asterischi (*) o un altro carattere speciale, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere.

[Linee guida e convenzioni sulla sintassi](#)

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Note

Potrebbe essere necessario aggiungere un prefisso al percorso del file per indicare in quale elemento o fonte trovarlo. Per ulteriori informazioni, consultare [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

Interfaccia utente corrispondente: output in output/file prodotti tab/Artifacts/New da build

Variables - output

(*action-name*/Outputs/Variables)

(Facoltativo)

Specificate le variabili che desiderate che l'azione esporti in modo che siano disponibili per l'uso da parte delle azioni successive.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Output/Variabili/ Aggiungi variabile

nome-variabile-1

(*action-name*/Outputs/Variables/*variable-name-1*)

(Facoltativo)

Specificate il nome di una variabile che desiderate che l'azione esporti. Questa variabile deve essere già definita nella Steps sezione Inputs o della stessa azione.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: tab/Variables/Add variabile/nome degli output

AutoDiscoverReports

(*action-name*/Outputs/AutoDiscoverReports)

(Facoltativo)

Definisce la configurazione per la funzione di individuazione automatica.

Quando abiliti l'individuazione automatica, tutte le CodeCatalyst ricerche Inputs passano all'azione e tutti i file generati dall'azione stessa, alla ricerca di report di test, copertura del codice e analisi della composizione del software (SCA). Per ogni report trovato, lo CodeCatalyst trasforma in un report. CodeCatalyst Un CodeCatalyst report è un report completamente integrato nel CodeCatalyst servizio e che può essere visualizzato e manipolato tramite la CodeCatalyst console.

Note

Per impostazione predefinita, la funzione di individuazione automatica ispeziona tutti i file. È possibile limitare i file da ispezionare utilizzando le [IncludePaths](#) proprietà o. [ExcludePaths](#)

Interfaccia utente corrispondente: scheda Outputs/Rapports/Auto-discover reports

Enabled

(*action-name*/Outputs/AutoDiscoverReports/Enabled)

(Facoltativo)

Abilita o disabilita la funzione di rilevamento automatico.

I valori validi sono true e false.

Se Enabled viene omissso, l'impostazione predefinita è. true

Interfaccia utente corrispondente: scheda Outputs/Reports/ Auto-discover reports

ReportNamePrefix

(*action-name*/Outputs/AutoDiscoverReports/ReportNamePrefix)

(Richiesto se è incluso e abilitato) [AutoDiscoverReports](#)

Specificate un prefisso che CodeCatalyst precede tutti i report trovati per assegnare un nome ai report associati. CodeCatalyst Ad esempio, se si specifica un prefisso

di `AutoDiscovered` e `CodeCatalyst` rileva automaticamente due rapporti di test `TestSuiteOne.xml` e `TestSuiteTwo.xml`, i `CodeCatalyst` report associati verranno denominati `e.AutoDiscoveredTestSuiteOne` `AutoDiscoveredTestSuiteTwo`

Interfaccia utente corrispondente: scheda `Outputs/Reports/` Nome del prefisso

IncludePaths

(*action-name*/Outputs/AutoDiscoverReports/IncludePaths)

Or

(*action-name*/Outputs/Reports/*report-name-1*/IncludePaths)

(Richiesto se [AutoDiscoverReports](#) è incluso e abilitato o se è incluso) [Reports](#)

Specificate i file e i percorsi dei file da `CodeCatalyst` includere nella ricerca di report non elaborati. Ad esempio, se si specifica `"/test/report/*"`, `CodeCatalyst` cerca la directory nell'intera [immagine di build](#) utilizzata dall'azione per cercare la `/test/report/*` directory. Quando trova quella directory, cerca `CodeCatalyst` i report in quella directory.

Note

Se il percorso del file include uno o più asterischi (*) o altri caratteri speciali, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere. [Linee guida e convenzioni sulla sintassi](#)

Se questa proprietà viene omessa, l'impostazione predefinita è `"**/*"`, ovvero la ricerca include tutti i file in tutti i percorsi.

Note

Per i report configurati manualmente, `IncludePaths` deve essere presente uno schema a globo che corrisponda a un singolo file.

Interfaccia utente corrispondente:

- Percorsi di `tab/Reports/Auto-discover reports/Include/exclude` uscita/percorsi di inclusione

- Gli output tab/Reports/Manually configurano i **report-name-1** rapporti/ /include/escludi percorsi/ Includi percorsi

ExcludePaths

(**action-name**/Outputs/AutoDiscoverReports/ExcludePaths)

Or

(**action-name**/Outputs/Reports/**report-name-1**/ExcludePaths)

(Facoltativo)

Specificate i file e i percorsi dei file da escludere durante la ricerca di report non elaborati.

CodeCatalyst Ad esempio, se si specifica "/test/my-reports/**/*", non CodeCatalyst cercherà i file nella /test/my-reports/ directory. Per ignorare tutti i file in una directory, utilizzate il pattern ****/*** glob.

Note

Se il percorso del file include uno o più asterischi (*) o altri caratteri speciali, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere.

[Linee guida e convenzioni sulla sintassi](#)

Interfaccia utente corrispondente:

- Percorsi di tab/Reports/Auto-discover reports/Include/exclude uscita/percorsi di esclusione
- Gli output tab/Reports/Manually configurano i **report-name-1** report/ /include/escludi percorsi/ Escludi percorsi

SuccessCriteria

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria)

Or

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria)

(Facoltativo)

Specificate i criteri di successo per i report di test, copertura del codice, analisi della composizione del software (SCA) e analisi statica (SA).

Per ulteriori informazioni, consulta [Configurazione dei criteri di successo per i report](#).

Interfaccia utente corrispondente: scheda Output/Rapporti/Criteri di successo

PassRate

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/PassRate)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/PassRate)

(Facoltativo)

Specificate la percentuale di test da superare in un rapporto di prova affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri relativi alla percentuale di superamento vengono applicati solo ai rapporti di prova. Per ulteriori informazioni sui rapporti di prova, vedere [Rapporti dei test](#).

Interfaccia utente corrispondente: tab/Reports/Success criteri di uscita/velocità di passaggio

LineCoverage

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/LineCoverage)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/LineCoverage)

(Facoltativo)

Specificate la percentuale di righe da coprire in un rapporto sulla copertura del codice affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri di copertura delle linee vengono applicati solo ai report sulla copertura del codice. Per ulteriori informazioni sui rapporti sulla copertura del codice, vedere [Report di copertura del codice](#).

Interfaccia utente corrispondente: tab/Reports/Success criteri di uscita/copertura di linea

BranchCoverage

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/BranchCoverage)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/BranchCoverage)

(Facoltativo)

Specificate la percentuale di filiali in un rapporto sulla copertura del codice che deve essere coperta affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri di copertura delle filiali vengono applicati solo ai report sulla copertura del codice. Per ulteriori informazioni sui rapporti sulla copertura del codice, vedere [Report di copertura del codice](#).

Interfaccia utente corrispondente: tab/Reports/Success criteri di uscita/copertura delle filiali

Vulnerabilities

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/Vulnerabilities)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/Vulnerabilities)

(Facoltativo)

Specificate il numero e la gravità massimi di vulnerabilità consentiti nel rapporto SCA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare le vulnerabilità, è necessario specificare:

- La gravità minima delle vulnerabilità da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH,, MEDIUMLOW, INFORMATIONAL.

Ad esempio, se si sceglie HIGH, verranno HIGH CRITICAL conteggiate tutte le vulnerabilità.

- Il numero massimo di vulnerabilità della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri di vulnerabilità vengono applicati solo ai report SCA. Per ulteriori informazioni sui report SCA, vedere. [Rapporti di analisi della composizione del software](#)

Per specificare la severità minima, utilizzare la `Severity` proprietà. Per specificare il numero massimo di vulnerabilità, utilizzare la `Number` proprietà.

Interfaccia utente corrispondente: `tab/Reports/Success` criteri di uscita/vulnerabilità

StaticAnalysisBug

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisBug)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisBug)

(Facoltativo)

Specificate il numero e la gravità massimi di bug consentiti nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare i bug, è necessario specificare:

- La gravità minima dei bug da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH,, MEDIUMLOW, INFORMATIONAL.

Ad esempio, se scegli HIGH, allora HIGH i CRITICAL bug verranno conteggiati.

- Il numero massimo di bug della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri relativi ai bug vengono applicati solo ai report PyLint e ESLint SA. Per ulteriori informazioni sui report SA, vedere [Rapporti di analisi statica](#).

Per specificare la severità minima, utilizzare la `Severity` proprietà. Per specificare il numero massimo di vulnerabilità, utilizzare la `Number` proprietà.

Interfaccia utente corrispondente: `tab/Reports/Success` criteri di uscita/bug

StaticAnalysisSecurity

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisSecurity)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisSecurity)

(Facoltativo)

Specificate il numero e la gravità massimi di vulnerabilità di sicurezza consentite nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare le vulnerabilità di sicurezza, è necessario specificare:

- La gravità minima delle vulnerabilità di sicurezza che si desidera includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Ad esempio, se lo desideri HIGH, verranno HIGH conteggiate le vulnerabilità di CRITICAL sicurezza.

- Il numero massimo di vulnerabilità di sicurezza della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri di vulnerabilità di sicurezza vengono applicati solo ai report PyLint e ESLint SA. Per ulteriori informazioni sui report SA, vedere [Rapporti di analisi statica](#).

Per specificare la severità minima, utilizzare la `Severity` proprietà. Per specificare il numero massimo di vulnerabilità, utilizzare la `Number` proprietà.

Interfaccia utente corrispondente: tab/Reports/Success criteri di uscita/vulnerabilità di sicurezza

StaticAnalysisQuality

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisQuality)

Or

(*action-name*/Outputs/Reports/*report-name-1*/SuccessCriteria/StaticAnalysisQuality)

(Facoltativo)

Specificate il numero e la gravità massimi dei problemi di qualità consentiti nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare i problemi di qualità, è necessario specificare:

- La gravità minima dei problemi di qualità da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Ad esempio, se si sceglie `HIGH`, verranno presi `HIGH` in `CRITICAL` considerazione i problemi di qualità.

- Il numero massimo di problemi di qualità della gravità specificata che si desidera consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri relativi ai problemi di qualità vengono applicati solo PyLint ai report ESLint SA. Per ulteriori informazioni sui report SA, vedere [Rapporti di analisi statica](#).

Per specificare la severità minima, utilizzare la `Severity` proprietà. Per specificare il numero massimo di vulnerabilità, utilizzare la `Number` proprietà.

Interfaccia utente corrispondente: `tab/Reports/Success` criteri di uscita/problemi di qualità

StaticAnalysisFinding

(`action-name`/Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisFinding)

Or

(`action-name`/Outputs/Reports/`report-name-1`/SuccessCriteria/StaticAnalysisFinding)

(Facoltativo)

Specificare il numero e la gravità massimi dei risultati consentiti nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare i risultati, è necessario specificare:

- La gravità minima dei risultati da includere nel conteggio. I valori validi, dal più grave al meno severo, sono: `CRITICAL`, `HIGH`, `MEDIUM`, `LOW`, `INFORMATIONAL`.

Ad esempio, se scegli `HIGH`, allora `HIGH` i `CRITICAL` risultati verranno conteggiati.

- Il numero massimo di risultati della gravità specificata che si desidera consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I risultati si applicano solo ai rapporti SARIF SA. Per ulteriori informazioni sui rapporti SA, vedere [Rapporti di analisi statica](#).

Per specificare la severità minima, utilizzare la `Severity` proprietà. Per specificare il numero massimo di vulnerabilità, utilizzare la `Number` proprietà.

Interfaccia utente corrispondente: tab/Reports/Success criteri di uscita/risultati

Reports

(*action-name*/Outputs/Reports)

(Facoltativo)

Una sezione che specifica la configurazione per i rapporti di prova.

Interfaccia utente corrispondente: scheda Output/Report

nome-rapporto-1

(nome-relazione-1) *action-name* /Outputs/Reports/

(Obbligatorio se incluso) [Reports](#)

Il nome che vuoi dare al CodeCatalyst rapporto che verrà generato dai tuoi report non elaborati.

Interfaccia utente corrispondente: Outputs tab/Reports/Manually configura report/nome del report

Format

(*action-name*/Outputs/Reports/*report-name-1*/Format)

(Obbligatorio se incluso [Reports](#))

Specificate il formato di file che state utilizzando per i report. I valori possibili sono i seguenti.

- Per i rapporti sui test:
 - Per Cucumber JSON, specifica Cucumber (editor visivo) o CUCUMBERJSON (editor YAML).
 - Per JUnit XML, specifica JUnit(editor visivo) o JUNITXML (editor YAML).
 - Per NUnit XML, specificare NUnit(editor visuale) o NUNITXML (editor YAML).
 - Per NUnit 3 XML, specificare NUnit3(editor visuale) o NUNIT3XML (editor YAML).
 - Per Visual Studio TRX, specifica Visual Studio TRX (editor visivo) o VISUALSTUDIOTRX (editor YAML).

- Per TestNg XML, specificare TestNg (editor visivo) o **TESTNGXML** (editor YAML).
- Per i report sulla copertura del codice:
 - Per Clover XML, specifica Clover (editor visivo) o CLOVERXML (editor YAML).
 - Per Cobertura XML, specifica Cobertura (editor visivo) o (editor YAML). COBERTURAXML
 - Per JaCoCo XML, specifica JaCoCo(editor visivo) o (editor YAML). JACOCOXML
 - Per SimpleCov JSON generato da [simplecov](#), [non da simplecov-json](#), [specifica Simplecov](#) (editor visivo) o (editor YAML). SIMPLECOV
- Per i report di analisi della composizione del software (SCA):
 - Per SARIF, specifica SARIF (editor visivo) o SARIFSCA (editor YAML).

Interfaccia utente corrispondente: restituisce i tab/Reports/Manually configure reports/Add/configure **report-name-1** reports//Tipo di report e il formato del report

Configuration

(*action-name*/Configuration)

(Obbligatorio) Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

Container

(*action-name*/Configuration/Container)

(Facoltativo)

Specificate l'immagine Docker, o contenitore, che l'azione utilizza per completare l'elaborazione. È possibile specificare una delle [immagini attive](#) fornite in CodeCatalyst dotazione oppure utilizzare un'immagine personalizzata. Se scegli di utilizzare la tua immagine, questa può risiedere in Amazon ECR, Docker Hub o in un altro registro. Se non specifichi un'immagine Docker, l'azione utilizza una delle immagini attive per l'elaborazione. Per informazioni su quale immagine attiva viene utilizzata per impostazione predefinita, consulta [Immagini attive](#).

Per ulteriori informazioni su come specificare un'immagine Docker personalizzata, consulta [Assegnazione di un'immagine Docker in un ambiente di runtime personalizzato a un'azione](#)

Interfaccia utente corrispondente: immagine Docker dell'ambiente di esecuzione - opzionale

Registry

(*action-name*/Configuration/Container/Registry)

(Obbligatorio se Container incluso)

Specificate il registro in cui è archiviata l'immagine. I valori validi includono:

- CODECATALYST(Editor YAML)

L'immagine è memorizzata nel registro. CodeCatalyst

- Docker Hub (editor visivo) o DockerHub (editor YAML)

L'immagine è memorizzata nel registro delle immagini di Docker Hub.

- Altro registro (editor visivo) o Other (editor YAML)

L'immagine viene memorizzata in un registro di immagini personalizzato. È possibile utilizzare qualsiasi registro disponibile pubblicamente.

- Amazon Elastic Container Registry (editor visivo) o ECR (editor YAML)

L'immagine è archiviata in un repository di immagini di Amazon Elastic Container Registry. Per utilizzare un'immagine in un repository Amazon ECR, questa azione richiede l'accesso ad Amazon ECR. Per abilitare questo accesso, devi creare un [ruolo IAM](#) che includa le seguenti autorizzazioni e una politica di fiducia personalizzata. (Se lo desideri, puoi modificare un ruolo esistente per includere le autorizzazioni e la policy.)

Il ruolo IAM deve includere le seguenti autorizzazioni nella sua politica del ruolo:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

Il ruolo IAM deve includere la seguente politica di fiducia personalizzata:

Per ulteriori informazioni sulla creazione di ruoli IAM, consulta [Creating a role using custom trust policies \(console\)](#) nella IAM User Guide.

Una volta creato il ruolo, è necessario assegnarlo all'azione tramite un ambiente. Per ulteriori informazioni, consulta [Associare un ambiente a un'azione](#).

Interfaccia utente corrispondente: Amazon Elastic Container Registry, Docker Hub e altre opzioni di registro

Image

(*action-name*/Configuration/Container/Image)

(Obbligatorio se Container incluso)

Specifica una delle seguenti proprietà:

- Se utilizzate un CODECATALYST registro, impostate l'immagine su una delle seguenti [immagini attive](#):
 - CodeCatalystLinux_x86_64:2024_03
 - CodeCatalystLinux_x86_64:2022_11
 - CodeCatalystLinux_Arm64:2024_03
 - CodeCatalystLinux_Arm64:2022_11
 - CodeCatalystLinuxLambda_x86_64:2024_03
 - CodeCatalystLinuxLambda_x86_64:2022_11
 - CodeCatalystLinuxLambda_Arm64:2024_03
 - CodeCatalystLinuxLambda_Arm64:2022_11
 - CodeCatalystWindows_x86_64:2022_11
- Se utilizzi un registro Docker Hub, imposta l'immagine sul nome dell'immagine Docker Hub e sul tag opzionale.

Ad esempio: postgres:latest

- Se utilizzi un registro Amazon ECR, imposta l'immagine sull'URI del registro Amazon ECR.

Ad esempio: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo

- Se utilizzi un registro personalizzato, imposta l'immagine sul valore previsto dal registro personalizzato.

Interfaccia utente corrispondente: immagine docker dell'ambiente di runtime (se il registro è **CODECATALYST**), immagine Docker Hub (se il registro è Docker Hub), URL dell'immagine ECR (se il registro è Amazon Elastic Container Registry) e URL dell'immagine (se il registro è Other registry).

Steps

(*action-name*/Configuration/Steps)

(Obbligatorio)

Specificate i comandi di shell da eseguire durante l'operazione di installazione, configurazione ed esecuzione degli strumenti di compilazione.

Ecco un esempio di come creare un progetto npm:

Steps:

- Run: `npm install`
- Run: `npm run build`

Ecco un esempio di come specificare i percorsi dei file:

Steps:

- Run: `cd $ACTION_BUILD_SOURCE_PATH_WorkflowSource/app && cat file2.txt`
- Run: `cd $ACTION_BUILD_SOURCE_PATH_MyBuildArtifact/build-output/ && cat file.txt`

Per ulteriori informazioni sulla specificazione dei percorsi dei file, vedere [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

Interfaccia utente corrispondente: scheda di configurazione/comandi Shell

Packages

(*action-name*/Configuration/Packages)

(Facoltativo)

Una sezione in cui è possibile specificare un archivio di pacchetti utilizzato dall'azione per risolvere le dipendenze. I pacchetti consentono di archiviare e condividere in modo sicuro i pacchetti software utilizzati per lo sviluppo di applicazioni.

Per ulteriori informazioni sui pacchetti, vedere [Pubblica e condividi pacchetti software in CodeCatalyst](#).

Interfaccia utente corrispondente: scheda di configurazione/ Pacchetti

NpmConfiguration

(*action-name*/Configuration/Packages/NpmConfiguration)

(Richiesto se [Packages](#) incluso)

Una sezione che definisce la configurazione per il formato del pacchetto npm. Questa configurazione viene utilizzata da un'azione durante l'esecuzione di un flusso di lavoro.

Per ulteriori informazioni sulla configurazione del pacchetto npm, vedere [Utilizzo di npm](#).

Interfaccia utente corrispondente: Configuration tab/Packages/Add configuration/ npm

PackageRegistries

(*action-name*/Configuration/Packages/NpmConfiguration/PackageRegistries)

(Richiesto se incluso [Packages](#))

Una sezione in cui è possibile definire le proprietà di configurazione di una sequenza di repository di pacchetti.

Interfaccia utente corrispondente: tab/Packages/Add configuration/npm Configurazione/Aggiungi archivio di pacchetti

PackagesRepository

(*action-name*/Configuration/Packages/NpmConfiguration/PackageRegistries/PackagesRepository)

(Richiesto se [Packages](#) incluso)

Specificate il nome dell'archivio dei CodeCatalyst pacchetti che desiderate venga utilizzato dall'azione.

Se specificate più repository predefiniti, l'ultimo repository avrà la priorità.

Per ulteriori informazioni sugli archivi dei pacchetti, vedere. [Archivi di pacchetti](#)

Interfaccia utente corrispondente: archivio dei tab/Packages/Add configuration/npm/Add pacchetti di configurazione/archivio dei pacchetti

Scopes

(*action-name*/Configuration/Packages/NpmConfiguration/PackageRegistries/Scopes)

(Facoltativo)

Specificate una sequenza di ambiti che desiderate definire nel registro dei pacchetti. Quando si definiscono gli ambiti, l'archivio dei pacchetti specificato viene configurato come registro per tutti gli ambiti elencati. Se un pacchetto con l'ambito viene richiesto tramite il client npm, utilizzerà quel repository anziché quello predefinito. Ogni nome di ambito deve avere il prefisso «@».

Se includi gli ambiti di override, l'ultimo repository avrà la priorità.

Se Scopes viene omissso, l'archivio dei pacchetti specificato viene configurato come registro predefinito per tutti i pacchetti utilizzati dall'azione.

[Per ulteriori informazioni sugli ambiti, vedere i pacchetti and Namespace dei pacchetti Scoped.](#)

Interfaccia utente corrispondente: repository/ Scopes dei tab/Packages/Add configuration/npm/Add pacchetti di configurazione - opzionale

ExportAuthorizationToken

(*action-name*/Configuration/Packages/ExportAuthorizationToken)

(Facoltativo)

Abilita o disabilita la funzionalità del token di autorizzazione all'esportazione. Se abilitata, i token di autorizzazione esportati possono essere utilizzati per configurare manualmente un gestore di pacchetti per l'autenticazione con CodeCatalyst gli archivi di pacchetti. È possibile utilizzare il token come variabile di ambiente a cui fare riferimento nelle proprie azioni.

I valori validi sono true e false.

Se ExportAuthorizationToken viene omissso, l'impostazione predefinita è. false

Per ulteriori informazioni sul token di autorizzazione all'esportazione, vedere [Utilizzo dei token di autorizzazione nelle azioni del flusso di lavoro](#).

Interfaccia utente corrispondente: scheda di configurazione/Pacchetti/ Token di autorizzazione all'esportazione

Test con flussi di lavoro

In CodeCatalyst, puoi eseguire test come parte di diverse azioni del flusso di lavoro, come build e test. Tutte queste azioni del flusso di lavoro possono generare report sulla qualità. Un'azione di test è un'azione del flusso di lavoro che produce report di test, copertura del codice, analisi della

composizione del software e analisi statiche. Questi report vengono visualizzati nella CodeCatalyst console.

Argomenti

- [Tipi di report sulla qualità](#)
- [Aggiungere l'azione di test](#)
- [Visualizzazione dei risultati di un'azione di test](#)
- [Ignorare i test non riusciti in un'azione](#)
- [Integrazione con universal-test-runner](#)
- [Configurazione dei report sulla qualità in un'azione](#)
- [Le migliori pratiche per i test](#)
- [Proprietà SARIF supportate](#)

Tipi di report sulla qualità

L'azione CodeCatalyst di test di Amazon supporta i seguenti tipi di report sulla qualità. Per un esempio su come formattare questi report in YAML, consulta. [Rapporti sulla qualità \(esempio YAML\)](#).

Argomenti

- [Rapporti dei test](#)
- [Report di copertura del codice](#)
- [Rapporti di analisi della composizione del software](#)
- [Rapporti di analisi statica](#)

Rapporti dei test

In CodeCatalyst, puoi configurare test unitari, test di integrazione e test di sistema da eseguire durante le build. Quindi CodeCatalyst puoi creare report che contengono i risultati dei tuoi test.

È possibile utilizzare un rapporto di prova per risolvere i problemi relativi ai test. Se disponi di molti rapporti di test relativi a più build, puoi utilizzarli per visualizzare i tassi di errore e ottimizzare le tue build.

Puoi utilizzare i seguenti formati di file dei report di test:

- Cucumber JSON (.json)

- JUnit XML (.xml)
- NUnit XML (.xml)
- NUnit3 XML (.xml)
- TestNg XML (.xml)
- Visual Studio TRX (.trx, .xml)

Report di copertura del codice

Nel CodeCatalyst, puoi generare report sulla copertura del codice per i tuoi test. CodeCatalyst fornisce le seguenti metriche di copertura del codice:

Copertura della linea

Misura il numero di affermazioni coperte dai test. Una dichiarazione è una singola istruzione, esclusi i commenti.

```
line coverage = (total lines covered)/(total number of lines)
```

Copertura del ramo

Misura il numero di rami coperti dai test rispetto a ogni possibile ramo di una struttura di controllo, ad esempio un'istruzione if or.

```
branch coverage = (total branches covered)/(total number of branches)
```

Sono supportati i seguenti formati di file di report di copertura del codice:

- JaCoCo XML (.xml)
- SimpleCov [JSON \(generato da simplecov, non da simplecov-json, .json\)](#)
- Clover XML (versione 3, .xml)
- Copertura XML (.xml)
- LCOV (.info)

Rapporti di analisi della composizione del software

In CodeCatalyst, puoi utilizzare gli strumenti di analisi della composizione del software (SCA) per analizzare i componenti dell'applicazione e verificare eventuali vulnerabilità di sicurezza note. È possibile scoprire e analizzare i report SARIF che descrivono in dettaglio le vulnerabilità con

gravità diverse e i modi per risolverle. I valori di gravità validi, dalla più severa alla meno severa, sono:CRITICAL,,,,, HIGH MEDIUM LOW INFORMATIONAL

Sono supportati i seguenti formati di file di report SCA:

- SARIF (.sarif, .json)

Rapporti di analisi statica

Puoi utilizzare i report di analisi statica (SA) per identificare i difetti del codice a livello di sorgente. Inoltre CodeCatalyst, puoi generare report SA per aiutare a risolvere i problemi del codice prima di distribuirlo. Questi problemi includono bug, vulnerabilità di sicurezza, problemi di qualità e altre vulnerabilità. I valori di gravità validi, dalla più severa alla meno severa, sono:CRITICAL,, HIGHMEDIUM, LOW e. INFORMATIONAL

CodeCatalyst fornisce le seguenti metriche SA:

bug

Identifica una serie di possibili bug trovati nel codice sorgente. Questi bug possono includere problemi relativi alla sicurezza della memoria. Di seguito è riportato un esempio di bug.

```
// The while loop will inadvertently index into array x out-of-bounds
int x[64];
while (int n = 0; n <= 64; n++) {
    x[n] = 0;
}
```

Vulnerabilità di sicurezza

Identifica una serie di possibili vulnerabilità di sicurezza presenti nel codice sorgente. Queste vulnerabilità di sicurezza possono includere problemi come la memorizzazione dei token segreti in testo non crittografato.

Problemi di qualità

Identifica una serie di possibili problemi di qualità riscontrati nel codice sorgente. Questi problemi di qualità possono includere problemi riguardanti le convenzioni di stile. Di seguito è riportato un esempio di problema di qualità.

```
// The function name doesn't adhere to the style convention of camelCase
int SUBTRACT(int x, int y) {
```

```
    return x-y  
}
```

Altre vulnerabilità

Identifica una serie di possibili altre vulnerabilità presenti nel codice sorgente.

CodeCatalyst supporta i seguenti formati di file di report SA:

- PyLint (.py)
- ESLint (.js, .jsx, .ts, .tsx)
- SARIF (.sarif, .json)

Aggiungere l'azione di test

Utilizza la seguente procedura per aggiungere un'azione di test al tuo CodeCatalyst flusso di lavoro.

Visual

Per aggiungere un'azione di test utilizzando l'editor visuale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli Visual.
6. Scegli Azioni.
7. In Azioni, scegli Test.
8. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, vedi. [Crea e testa azioni YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e sul valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.

10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere un'azione di compilazione utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli YAML.
6. Scegli Azioni.
7. In Azioni, scegli Test.
8. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in [Crea e testa azioni YAML](#).
9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Definizione dell'azione di test

L'azione di test è definita come un insieme di proprietà YAML all'interno del file di definizione del flusso di lavoro. Per informazioni su queste proprietà, vedere [Crea e testa azioni YAML](#) in [Definizione YAML del flusso di lavoro](#)

Visualizzazione dei risultati di un'azione di test

Utilizza le seguenti istruzioni per visualizzare i risultati di un'azione di test, inclusi i log, i report e le variabili generati.

Per visualizzare i risultati di un'azione di test

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

2. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
3. Nel diagramma del flusso di lavoro, scegli il nome dell'azione di test, ad esempio Test.
4. Per visualizzare i log generati da un'azione, scegli Registri. Vengono visualizzati i registri delle varie fasi dell'azione. È possibile espandere o comprimere i registri in base alle esigenze.
5. Per visualizzare i report sui test prodotti dall'azione di test, scegli Rapporti o nel riquadro di navigazione scegli Report. Per ulteriori informazioni, consulta [Tipi di report sulla qualità](#).
6. Per visualizzare la configurazione utilizzata per l'azione di test, scegli Configurazione. Per ulteriori informazioni, consulta [Aggiungere l'azione di test](#).
7. Per visualizzare le variabili utilizzate dall'azione di test, scegliete Variabili. Per ulteriori informazioni, consulta [Utilizzo delle variabili nei flussi di lavoro](#).

Ignorare i test non riusciti in un'azione

Se la tua azione ha più di un comando di test, potresti voler consentire l'esecuzione dei comandi di test successivi dell'azione anche se un comando precedente fallisce. Ad esempio, nei seguenti comandi, potresti test2 voler eseguirli sempre, anche se test1 fallisce.

Steps:

- Run: `npm install`
- Run: `npm run test1`
- Run: `npm run test2`

Normalmente, quando un passaggio restituisce un errore, Amazon CodeCatalyst interrompe l'azione del flusso di lavoro e la contrassegna come non riuscita. Puoi consentire ai passaggi di azione di continuare a essere eseguiti reindirizzando l'output dell'errore a. `null` È possibile eseguire questa operazione aggiungendo `2>/dev/null` al comando. Con questa modifica, l'esempio precedente sarebbe simile al seguente.

Steps:

- Run: `npm install`
- Run: `npm run test1 2>/dev/null`
- Run: `npm run test2`

Nel secondo frammento di codice, lo stato del `npm install` comando verrà rispettato, ma qualsiasi errore restituito dal `npm run test1` comando verrà ignorato. Di conseguenza, il comando

viene `npm run test2` eseguito. In questo modo, è possibile visualizzare entrambi i report contemporaneamente indipendentemente dal fatto che si verifichi un errore.

Integrazione con universal-test-runner

Le azioni di test si integrano con lo strumento `universal-test-runner` a riga di comando open source. `universal-test-runner` utilizza il [Test Execution Protocol](#) per eseguire i test per qualsiasi lingua in un determinato framework. `universal-test-runner` supporta i seguenti framework:

- [Gradle](#)
- [Scherzo](#)
- [Maven](#)
- [Pytest](#)
- [.NET](#)

`universal-test-runner` viene installato solo sulle immagini curate per le azioni di test. Se configuri un'azione di test per utilizzare un Docker Hub o Amazon ECR personalizzato, devi installarlo manualmente `universal-test-runner` per abilitare le funzionalità di test avanzate. A tale scopo, installa Node.js (14 o versione successiva) sull'immagine, quindi esegui l'installazione `universal-test-runner` `npm` utilizzando il comando shell. - Run: `npm install -g @aws/universal-test-runner` Per ulteriori informazioni sull'installazione di Node.js nel contenitore tramite i comandi della shell, consulta [Installazione e aggiornamento di Node Version Manager](#).

Per ulteriori informazioni su `universal-test-runner`, consulta [Che cos'è universal-test-runner](#).

Visual

Da utilizzare `universal-test-runner` nell'editor visuale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro.
4. Scegli Modifica.
5. Scegli Visual.
6. Scegli Azioni.
7. In Azioni, scegli Test.

8. Nella scheda Configurazione, completa il campo Comandi Shell aggiornando il codice di esempio con la tua scelta dei framework supportati. Ad esempio, per utilizzare un framework supportato, è necessario utilizzare un Run comando simile al seguente.

```
- Run: run-tests <framework>
```

Se il framework desiderato non è supportato, valuta la possibilità di aggiungere un adattatore o un runner personalizzato. Per una descrizione del campo dei comandi Shell, consulta [Steps](#).

9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Da usare universal-test-runner nell'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro.
4. Scegli Modifica.
5. Scegli YAML.
6. Scegli Azioni.
7. In Azioni, scegli Test.
8. Modifica il codice YAML in base alle tue esigenze. Ad esempio, per utilizzare un framework supportato, è necessario utilizzare un Run comando simile al seguente.

```
Configuration:
  Steps:
    - Run: run-tests <framework>
```

Se il framework desiderato non è supportato, valuta la possibilità di aggiungere un adattatore o un runner personalizzato. Per una descrizione della proprietà Steps, vedere [Steps](#).

9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Configurazione dei report sulla qualità in un'azione

Questa sezione descrive come configurare un rapporto sulla qualità in un'azione.

Argomenti

- [Rilevamento automatico e report manuali](#)
- [Configurazione dei criteri di successo per i report](#)
- [Rapporti sulla qualità \(esempio YAML\).](#)

Rilevamento automatico e report manuali

Quando l'individuazione automatica è abilitata, CodeCatalyst cerca tutti gli input passati all'azione e tutti i file generati dall'azione stessa, cercando report di test, copertura del codice, analisi della composizione del software (SCA) e analisi statica (SA). È possibile visualizzare e manipolare ciascuno di questi report in. CodeCatalyst

È inoltre possibile configurare manualmente i report da generare. Puoi specificare il tipo di rapporto che desideri generare e il formato del file. Per ulteriori informazioni, consulta [Tipi di report sulla qualità](#).

Configurazione dei criteri di successo per i report

È possibile impostare i valori che determinano i criteri di successo per un rapporto di test, copertura del codice, analisi della composizione del software (SCA) o analisi statica (SA).

I criteri di successo sono soglie che determinano se un report ha esito positivo o negativo. CodeCatalyst genera innanzitutto il rapporto, che può essere un rapporto di test, di copertura del codice, SCA o SA, e quindi applica i criteri di successo ai report generati. Quindi mostra se i criteri di successo sono stati soddisfatti e in che misura. Se un rapporto non soddisfa i criteri di successo specificati, l'azione CodeCatalyst che ha specificato i criteri di successo ha esito negativo.

Ad esempio, quando si impostano i criteri di successo per il rapporto SCA, i valori di vulnerabilità validi, dal più grave al meno grave, sono: CRITICAL,, HIGHMEDIUM,LOW. INFORMATIONAL Se si impostano i criteri per la scansione di una vulnerabilità per livello di HIGH gravità, il rapporto avrà esito negativo se è presente almeno una vulnerabilità con HIGH gravità o nessuna vulnerabilità con HIGH gravità, ma almeno una vulnerabilità con un livello di gravità più elevato, ad esempio una vulnerabilità per gravità. CRITICAL

Se non si specificano criteri di successo, allora:

- Il CodeCatalyst rapporto generato in base ai report non elaborati non mostrerà criteri di successo.
- I criteri di successo non verranno utilizzati per determinare se l'azione del flusso di lavoro associata ha esito positivo o negativo.

Visual

Per configurare i criteri di successo

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli un flusso di lavoro contenente un'azione che generi un report. Questo è il rapporto per il quale desideri applicare i criteri di successo. È possibile filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
3. Scegli Modifica.
4. Scegli Visual.
5. Nel diagramma del flusso di lavoro, scegli l'azione che hai configurato per generare CodeCatalyst report.
6. Seleziona la scheda Outputs (Output).
7. In Individuazione automatica dei report o in Configura manualmente i report, scegli Criteri di successo.

Vengono visualizzati i criteri di successo. A seconda delle selezioni precedenti, è possibile visualizzare alcune o tutte le seguenti opzioni:

Frequenza di superamento

Specificare la percentuale di test in un rapporto di prova che deve essere superato affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri relativi alla percentuale di superamento vengono applicati solo ai rapporti di prova. Per ulteriori informazioni sui rapporti di prova, vedere [Rapporti dei test](#).

Copertura della linea

Specificate la percentuale di righe in un rapporto sulla copertura del codice che devono essere coperte affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri di

copertura delle linee vengono applicati solo ai report sulla copertura del codice. Per ulteriori informazioni sui rapporti sulla copertura del codice, vedere [Report di copertura del codice](#).

Copertura delle filiali

Specificate la percentuale di filiali in un rapporto sulla copertura del codice che deve essere coperta affinché il CodeCatalyst rapporto associato venga contrassegnato come approvato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri di copertura delle filiali vengono applicati solo ai report sulla copertura del codice. Per ulteriori informazioni sui rapporti sulla copertura del codice, vedere [Report di copertura del codice](#).

Vulnerabilità (SCA)

Specificare il numero e la gravità massimi di vulnerabilità consentiti nel rapporto SCA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare le vulnerabilità, è necessario specificare:

- La gravità minima delle vulnerabilità da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Ad esempio, se si sceglie HIGH, verranno HIGH CRITICAL conteggiate tutte le vulnerabilità.

- Il numero massimo di vulnerabilità della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri di vulnerabilità vengono applicati solo ai report SCA. Per ulteriori informazioni sui report SCA, vedere. [Rapporti di analisi della composizione del software](#)

Bug

Specificate il numero e la gravità massimi di bug consentiti nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare i bug, è necessario specificare:

- La gravità minima dei bug da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Ad esempio, se scegli HIGH, allora HIGH i CRITICAL bug verranno conteggiati.

- Il numero massimo di bug della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri relativi ai bug vengono applicati solo PyLint ai report ESLint SA. Per ulteriori informazioni sui report SA, vedere [Rapporti di analisi statica](#).

Vulnerabilità di sicurezza

Specificare il numero e la gravità massimi di vulnerabilità di sicurezza consentite nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare le vulnerabilità di sicurezza, è necessario specificare:

- La gravità minima delle vulnerabilità di sicurezza da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Ad esempio, se lo desideri HIGH, verranno HIGH conteggiate le vulnerabilità di CRITICAL sicurezza.

- Il numero massimo di vulnerabilità di sicurezza della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri di vulnerabilità di sicurezza vengono applicati solo ai report PyLint e ESLint SA. Per ulteriori informazioni sui report SA, vedere [Rapporti di analisi statica](#).

Problemi di qualità

Specificare il numero e la gravità massimi dei problemi di qualità consentiti nel rapporto SA affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. Per specificare i problemi di qualità, è necessario specificare:

- La gravità minima dei problemi di qualità da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Ad esempio, se si sceglie HIGH, verranno presi HIGH in CRITICAL considerazione i problemi di qualità.

- Il numero massimo di problemi di qualità della gravità specificata che si desidera consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri relativi ai problemi di qualità vengono applicati solo PyLint ai report ESLint SA. Per ulteriori informazioni sui report SA, vedere [Rapporti di analisi statica](#).

8. Scegli Applica.
9. Esegui il flusso di lavoro per CodeCatalyst applicare criteri di successo ai report non elaborati e rigenera i CodeCatalyst report associati includendo le informazioni sui criteri di successo. Per ulteriori informazioni, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

YAML

Per configurare i criteri di successo

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli un flusso di lavoro contenente un'azione che generi un report. Questo è il rapporto per il quale desideri applicare i criteri di successo. È possibile filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
3. Scegli Modifica.
4. Scegli YAML.
5. Nel diagramma del flusso di lavoro, scegli l'azione che hai configurato per generare report. CodeCatalyst
6. Nel riquadro dei dettagli, scegli la scheda Output.
7. Nell'azione, nella AutoDiscoverReports sezione o nella Reports sezione, aggiungi una SuccessCriteri proprietà, insieme a PassRate, LineCoverage, BranchCoverage, Vulnerabilities StaticAnalysisBugStaticAnalysisSecurity, e StaticAnalysisQuality proprietà.

Per una spiegazione di ciascuna di queste proprietà, consulta [il Crea e testa azioni YAML](#).

8. Scegli Applica.
9. Esegui il flusso di lavoro per CodeCatalyst applicare criteri di successo ai report non elaborati e rigenera i CodeCatalyst report associati includendo le informazioni sui criteri di

successo. Per ulteriori informazioni sull'avvio di un flusso di lavoro, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Rapporti sulla qualità (esempio YAML).

L'esempio seguente mostra come configurare manualmente quattro report: un rapporto di test, un rapporto sulla copertura del codice, un rapporto di analisi della composizione del software e un rapporto di analisi statico.

```
Reports:
  MyTestReport:
    Format: JUNITXML
    IncludePaths:
      - "*.xml"
    ExcludePaths:
      - report1.xml
    SuccessCriteria:
      PassRate: 90
  MyCoverageReport:
    Format: CLOVERXML
    IncludePaths:
      - output/coverage/jest/clover.xml
    SuccessCriteria:
      LineCoverage: 75
      BranchCoverage: 75
  MySCARReport:
    Format: SARIFSCA
    IncludePaths:
      - output/sca/reports.xml
    SuccessCriteria:
      Vulnerabilities:
        Number: 5
        Severity: HIGH
  MySARReport:
    Format: ESLINTJSON
    IncludePaths:
      - output/static/eslint.xml
    SuccessCriteria:
      StaticAnalysisBug:
        Number: 10
        Severity: MEDIUM
      StaticAnalysisSecurity:
```

```
Number: 5
Severity: CRITICAL
StaticAnalysisQuality:
Number: 0
Severity: INFORMATIONAL
```

Le migliori pratiche per i test

Quando si utilizzano le funzionalità di test fornite da CodeCatalyst, si consiglia di seguire queste best practice.

Argomenti

- [Rilevamento automatico](#)
- [Criteri di successo](#)
- [Includi/escludi percorsi](#)

Rilevamento automatico

Durante la configurazione delle azioni CodeCatalyst, l'individuazione automatica consente di scoprire automaticamente gli output di vari strumenti, come i report dei JUnit test, e di generare CodeCatalyst report pertinenti da essi. L'individuazione automatica aiuta a garantire che i report continuino a essere generati anche se i nomi o i percorsi degli output rilevati cambiano. Quando vengono aggiunti nuovi file, li CodeCatalyst rileva automaticamente e produce report pertinenti. Tuttavia, se si utilizza l'individuazione automatica, è importante tenere conto di alcuni dei seguenti aspetti di questa funzionalità:

- Quando attivi l'individuazione automatica nella tua azione, tutti i report dello stesso tipo scoperti automaticamente condivideranno gli stessi criteri di successo. Ad esempio, un criterio condiviso come la frequenza minima di superamento si applicherebbe a tutti i report dei test scoperti automaticamente. Se hai bisogno di criteri diversi per report dello stesso tipo, devi configurare in modo esplicito ciascuno di questi report.
- L'individuazione automatica può anche trovare i report prodotti dalle dipendenze e, se sono configurati criteri di successo, l'azione su questi report potrebbe fallire. Questo problema può essere risolto aggiornando la configurazione del percorso di esclusione.
- Non è garantito che l'individuazione automatica produca sempre lo stesso elenco di report, poiché analizza l'azione in fase di esecuzione. Nel caso in cui si desideri che venga sempre prodotto un determinato report, è necessario configurare i report in modo esplicito. Ad esempio, se i test

smettessero di essere eseguiti come parte della build, il framework di test non produrrebbe alcun output e, di conseguenza, non verrebbe prodotto alcun rapporto di test e l'azione potrebbe avere successo. Se vuoi che il successo della tua azione dipenda da quel particolare test, devi configurare esplicitamente quel rapporto.

Tip

Quando inizi a lavorare su un progetto nuovo o esistente, usa l'individuazione automatica per l'intera directory del progetto (include**/*). Ciò richiama la generazione di report su tutti i file del progetto, inclusi quelli all'interno delle sottodirectory.

Per ulteriori informazioni, consulta [Configurazione dei report sulla qualità in un'azione](#).

Criteri di successo

È possibile applicare soglie di qualità ai report configurando criteri di successo. Ad esempio, se due report sulla copertura del codice sono stati scoperti automaticamente, uno con una copertura di linea dell'80% e l'altro con una copertura di linea del 60%, sono disponibili le seguenti opzioni:

- Imposta i criteri di successo del rilevamento automatico per la copertura di linea all'80%. Ciò comporterebbe l'approvazione del primo rapporto e il fallimento del secondo rapporto, con conseguente fallimento dell'azione complessiva. Per sbloccare il flusso di lavoro, aggiungi nuovi test al progetto fino a quando la copertura di linea per il secondo rapporto non supera l'80%.
- Imposta i criteri di successo del rilevamento automatico per la copertura di linea al 60%. Ciò comporterebbe l'approvazione di entrambi i report, il che comporterebbe il successo dell'azione. Potresti quindi lavorare per aumentare la copertura del codice nel secondo rapporto. Tuttavia, con questo approccio, non è possibile garantire che la copertura del primo rapporto non scenda al di sotto dell'80%.
- Configura in modo esplicito uno o entrambi i report utilizzando l'editor visivo o aggiungendo una sezione e un percorso YAML espliciti per ogni report. Ciò consentirebbe di configurare criteri di successo separati e nomi personalizzati per ogni rapporto. Tuttavia, con questo approccio, l'azione potrebbe fallire se i percorsi dei report cambiano.

Per ulteriori informazioni, consulta [Configurazione dei criteri di successo per i report](#).

Includi/escludi percorsi

Quando si esaminano i risultati delle azioni, è possibile modificare l'elenco dei report generati CodeCatalyst mediante la configurazione `IncludePaths` `ExcludePaths`.

- Utilizzato `IncludePaths` per specificare i file e i percorsi dei file CodeCatalyst da includere durante la ricerca di report. Ad esempio, se si specifica `/test/report/*`, CodeCatalyst cerca l'intera immagine di build utilizzata dall'azione che cerca la `/test/report/` directory. Quando trova quella directory, cerca CodeCatalyst i report in quella directory.

Note

Per i report configurati manualmente, `IncludePaths` deve essere presente uno schema a globo che corrisponda a un singolo file.

- `ExcludePaths` Da utilizzare per specificare i file e i percorsi dei file che si CodeCatalyst desidera escludere durante la ricerca di report. Ad esempio, se si specifica `/test/reports/**/*`, non CodeCatalyst cercherà i file nella `/test/reports/` directory. Per ignorare tutti i file in una directory, utilizzate il pattern `**/*` glob.

Di seguito sono riportati alcuni esempi di possibili pattern a glob.

Pattern	Description
<code>*.*</code>	Corrisponde a tutti i nomi di oggetti nella directory corrente che contengono un punto
<code>*.xml</code>	Corrisponde a tutti i nomi degli oggetti nella directory corrente che terminano con <code>.xml</code>
<code>*.{xml,txt}</code>	Corrisponde a tutti i nomi degli oggetti nella directory corrente che terminano con <code>.xml</code> o <code>.txt</code>
<code>**/*.xml</code>	Corrisponde ai nomi degli oggetti in tutte le directory che terminano con <code>.xml</code>

Pattern	Description
<code>testFolder</code>	Corrisponde a un oggetto chiamato <code>testFolder</code> , trattandolo come un file
<code>testFolder/*</code>	Corrisponde agli oggetti in un livello della sottocartella <code>testFolder</code> , ad esempio <code>testFolder/file.xml</code>
<code>testFolder/*/*</code>	Corrisponde agli oggetti in due livelli della sottocartella <code>testFolder</code> , ad esempio <code>testFolder/reportsFolder/file.xml</code>
<code>testFolder/**</code>	Individua la sottocartella <code>testFolder</code> , insieme ai file all'interno di <code>testFolder</code> , ad esempio <code>testFolder/file.xml</code> e <code>testFolder/otherFolder/file.xml</code>

CodeCatalyst interpreta i pattern globulari come segue:

- Il carattere slash (/) separa le directory nei percorsi dei file.
- L'asterisco (*) individua zero o più caratteri di un componente del nome entro i limiti della cartella.
- Un doppio asterisco (**) corrisponde a zero o più caratteri di un componente del nome in tutte le directory.

Note

`ExcludePaths` ha la precedenza su `IncludePaths`. Se entrambi `ExcludePaths` e `IncludePaths` includono la stessa cartella, tale cartella non viene analizzata per individuare eventuali report.

Proprietà SARIF supportate

Lo Static Analysis Results Interchange Format (SARIF) è un formato di file di output disponibile nell'analisi della composizione del software (SCA) e nei report di analisi statica in Amazon. CodeCatalyst L'esempio seguente mostra come configurare manualmente SARIF in un report di analisi statica:

```
Reports:
MySAReport:
Format: SARIFSA
IncludePaths:
  - output/sa_report.json
SuccessCriteria:
  StaticAnalysisFinding:
    Number: 25
    Severity: HIGH
```

CodeCatalyst supporta le seguenti proprietà SARIF che possono essere utilizzate per ottimizzare la visualizzazione dei risultati dell'analisi nei report.

Argomenti

- [Oggetto sarifLog](#)
- [Oggetto run](#)
- [Oggetto toolComponent](#)
- [Oggetto reportingDescriptor](#)
- [Oggetto result](#)
- [Oggetto location](#)
- [Oggetto physicalLocation](#)
- [Oggetto logicalLocation](#)
- [Oggetto fix](#)

Oggetto **sarifLog**

Name	Richiesto	Descrizione
<code>\$schema</code>	Sì	L'URI dello schema JSON SARIF per la versione 2.1.0.
<code>version</code>	Sì	CodeCatalyst supporta solo la versione SARIF 2.1.0.
<code>runs[]</code>	Sì	Un file SARIF contiene una matrice di una o più esecuzioni, ognuna delle quali rappresenta una singola esecuzione dello strumento di analisi.

Oggetto **run**

Name	Richiesto	Descrizione
<code>tool.driver</code>	Sì	Un <code>toolComponent</code> oggetto che descrive lo strumento di analisi.
<code>tool.name</code>	No	Una proprietà che indica il nome dello strumento utilizzato per eseguire l'analisi.
<code>results[]</code>	Sì	I risultati dello strumento di analisi visualizzati su CodeCatalyst.

Oggetto **toolComponent**

Name	Richiesto	Descrizione
name	Sì	Il nome dello strumento di analisi.
properties.artifactScanned	No	Numero totale di artefatti analizzati dallo strumento.
rules[]	Sì	Una serie di reporting Descriptor oggetti che rappresentano regole. In base a queste regole, lo strumento di analisi rileva problemi nel codice analizzato.

Oggetto **reportingDescriptor**

Name	Richiesto	Descrizione
id	Sì	L'identificatore univoco della regola utilizzata per fare riferimento a un risultato. Lunghezza massima: 1.024 caratteri
name	No	Il nome visualizzato della regola. Lunghezza massima: 1.024 caratteri
shortDescription.text	No	Una breve descrizione della regola.

Name	Richiesto	Descrizione
		Lunghezza massima: 3.000 caratteri
<code>fullDescription.text</code>	No	Una descrizione completa della regola. Lunghezza massima: 3.000 caratteri
<code>helpUri</code>	No	Una stringa che può essere localizzata per contenere l'URI assoluto della documentazione principale della regola. Lunghezza massima: 3.000 caratteri
<code>properties.unscore</code>	No	Un contrassegno che indica se il risultato della scansione è stato valutato.
<code>properties.score.severity</code>	No	Un set fisso di stringhe che specificano il livello di gravità del risultato. Lunghezza massima: 1.024 caratteri
<code>properties.cvssv3_baseSeverity</code>	No	Una valutazione di gravità qualitativa del Common Vulnerability Scoring System v3.1.
<code>properties.cvssv3_baseScore</code>	No	Un punteggio di base CVSS v3 compreso tra 0,0 e 10,0.

Name	Richiesto	Descrizione
<code>properties.cvssv2_severity</code>	No	Se i valori CVSS v3 non sono disponibili, cerca i valori CVSS v2. CodeCatalyst
<code>properties.cvssv2_score</code>	No	Un punteggio di base CVSS v2 compreso tra 0,0 e 10,0.
<code>properties.severity</code>	No	Un insieme fisso di stringhe che specificano il livello di gravità del risultato. Lunghezza massima: 1.024 caratteri
<code>defaultConfiguration.level</code>	No	La severità predefinita di una regola.

Oggetto **result**

Name	Richiesto	Descrizione
<code>ruleId</code>	Sì	L'identificatore univoco della regola utilizzata per fare riferimento a un risultato. Lunghezza massima: 1.024 caratteri
<code>ruleIndex</code>	Sì	L'indice della regola associata nel componente <code>rules[]</code> utensile.
<code>message.text</code>	Sì	Un messaggio che descrive il risultato e visualizza il messaggio per ogni risultato.

Name	Richiesto	Descrizione
		Lunghezza massima: 3.000 caratteri
rank	No	Un valore compreso tra 0,0 e 100,0 inclusi che rappresenta la priorità o l'importanza del risultato. Il valore di questa scala è 0,0 come priorità più bassa e 100,0 come priorità più alta.
level	No	La gravità del risultato. Lunghezza massima: 1.024 caratteri
properties.unscore	No	Un contrassegno che indica se il risultato della scansione è stato valutato.
properties.score.severity	No	Un set fisso di stringhe che specificano il livello di gravità del risultato. Lunghezza massima: 1.024 caratteri
properties.cvssv3_baseSeverity	No	Una valutazione di gravità qualitativa del Common Vulnerability Scoring System v3.1.
properties.cvssv3_baseScore	No	Un punteggio di base CVSS v3 compreso tra 0,0 e 10,0.

Name	Richiesto	Descrizione
<code>properties.cvssv2_severity</code>	No	Se i valori CVSS v3 non sono disponibili, cerca i valori CVSS v2. CodeCatalyst
<code>properties.cvssv2_score</code>	No	Un punteggio di base CVSS v2 compreso tra 0,0 e 10,0.
<code>properties.severity</code>	No	Un insieme fisso di stringhe che specificano il livello di gravità del risultato. Lunghezza massima: 1.024 caratteri
<code>locations[]</code>	Sì	L'insieme di posizioni in cui è stato rilevato il risultato. È necessario includere solo una posizione, a meno che il problema non possa essere corretto solo apportando una modifica in ogni posizione specificata. CodeCatalyst utilizza il primo valore nell'array di posizione per annotare il risultato. Numero massimo di location oggetti: 10
<code>relatedLocations[]</code>	No	Un elenco di riferimenti a località aggiuntivi inclusi nel risultato. Numero massimo di location oggetti: 50

Name	Richiesto	Descrizione
<code>fixes[]</code>	No	Una serie di <code>fix</code> oggetti che rappresentano la raccomandazione fornita dallo strumento di scansione. CodeCatalyst utilizza la prima raccomandazione dell' <code>fixesarray</code> .

Oggetto **location**

Name	Richiesto	Descrizione
<code>physicalLocation</code>	Sì	Identifica l'artefatto e la regione.
<code>logicalLocations[]</code>	No	L'insieme di posizioni descritte per nome senza riferimento all'artefatto.

Oggetto **physicalLocation**

Name	Richiesto	Descrizione
<code>artifactLocation.uri</code>	Sì	L'URI che indica la posizione di un artefatto, in genere un file nel repository o generato durante una compilazione.
<code>fileLocation.uri</code>	No	L'URI di riserva che indica la posizione del file. Viene utilizzato se i risultati <code>artifactLocation.uri</code> sono vuoti.

Name	Richiesto	Descrizione
<code>region.startLine</code>	Sì	Il numero di riga del primo carattere nella regione.
<code>region.startColumn</code>	Sì	Il numero di colonna del primo carattere nella regione.
<code>region.endLine</code>	Sì	Il numero di riga dell'ultimo carattere nella regione.
<code>region.endColumn</code>	Sì	Il numero di colonna dell'ultimo carattere nella regione.

Oggetto **logicalLocation**

Name	Richiesto	Description
<code>fullyQualifiedname</code>	No	Informazioni aggiuntive che descrivono la posizione del risultato. Lunghezza massima: 1.024 caratteri

Oggetto **fix**

Name	Richiesto	Description
<code>description.text</code>	No	Un messaggio che mostra un consiglio per ogni risultato. Lunghezza massima: 3.000 caratteri

Name	Richiesto	Description
<code>artifactChanges.[0].artifactLocation.uri</code>	No	L'URI che indica la posizione dell'elemento che deve essere aggiornato.

Implementazione con flussi di lavoro

Utilizzando [CodeCatalyst i flussi](#) di lavoro, puoi distribuire applicazioni e altre risorse per vari obiettivi come Amazon ECS e altro ancora AWS Lambda.

Come posso distribuire un'applicazione?

Per distribuire un'applicazione o una risorsa CodeCatalyst, è innanzitutto necessario creare un flusso di lavoro e quindi specificare un'azione di distribuzione al suo interno. Un'azione di distribuzione è un elemento costitutivo del flusso di lavoro che definisce cosa si desidera distribuire, dove distribuirlo e come lo si desidera (ad esempio, utilizzando uno schema). blue/green Puoi aggiungere un'azione di distribuzione al tuo flusso di lavoro utilizzando l'editor visivo della CodeCatalyst console o l'editor YAML.

I passaggi di alto livello per distribuire un'applicazione o una risorsa sono i seguenti.

Per distribuire un'applicazione (attività di alto livello)

1. Nel CodeCatalyst progetto, aggiungi il codice sorgente per un'applicazione che desideri distribuire. Per ulteriori informazioni, consulta [Memorizzazione del codice sorgente nei repository per un progetto in CodeCatalyst](#).
2. Nel CodeCatalyst progetto, aggiungi un ambiente che definisce la destinazione Account AWS e l'opzione Amazon Virtual Private Cloud (VPC) su cui desideri effettuare la distribuzione. Per ulteriori informazioni, consulta [Implementazione in e Account AWS VPCs](#).
3. Nel tuo CodeCatalyst progetto, crei un flusso di lavoro. Il flusso di lavoro è il luogo in cui definisci come creare, testare e distribuire l'applicazione. Per ulteriori informazioni, consulta [Guida introduttiva ai flussi di lavoro](#).
4. Nel flusso di lavoro, aggiungi un trigger, un'azione di creazione e, facoltativamente, un'azione di test. Per ulteriori informazioni, consultare [L'avvio di un flusso di lavoro viene eseguito](#)

[automaticamente utilizzando i trigger](#), [Aggiungere l'azione di compilazione](#) e [Aggiungere l'azione di test](#).

5. Nel flusso di lavoro, aggiungi un'azione di distribuzione. Puoi scegliere tra diverse azioni CodeCatalyst di distribuzione fornite alla tua applicazione verso destinazioni diverse, come Amazon ECS. (Puoi anche utilizzare un'azione di compilazione o un' GitHub azione per distribuire la tua applicazione. Per ulteriori informazioni sull'azione di compilazione e sulle GitHub azioni, consulta [Alternative all'implementazione delle azioni](#).)
6. Il flusso di lavoro viene avviato manualmente o automaticamente tramite un trigger. Il flusso di lavoro esegue le azioni di compilazione, test e distribuzione in sequenza per distribuire l'applicazione e le risorse sulla destinazione. Per ulteriori informazioni, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Elenco delle azioni di distribuzione

Sono disponibili le seguenti azioni di distribuzione:

- [Distribuisci CloudFormation stack](#): questa azione crea uno CloudFormation stack in AWS base a uno o AWS Serverless Application Model più CloudFormation modelli forniti dall'utente. Per ulteriori informazioni, consulta [Implementazione di uno stack CloudFormation](#).
- Distribuzione su Amazon ECS: questa azione registra un file di [definizione delle attività fornito](#) dall'utente. Per ulteriori informazioni, consulta [Distribuzione su Amazon ECS con un flusso di lavoro](#).
- Distribuzione su cluster Kubernetes: questa azione distribuisce un'applicazione in un cluster Amazon Elastic Kubernetes Service. Per ulteriori informazioni, consulta [Implementazione su Amazon EKS con un flusso di lavoro](#).
- AWS CDK [deploy](#): questa azione distribuisce un'app in AWS CDK AWS. Per ulteriori informazioni, consulta [Implementazione di un' AWS CDK app con un flusso di lavoro](#).

Note

Esistono altre CodeCatalyst azioni che possono distribuire risorse; tuttavia, non sono considerate azioni di distribuzione perché le relative informazioni sulla distribuzione non vengono visualizzate nella pagina Ambienti. Per ulteriori informazioni sulla pagina Ambienti e sulla visualizzazione delle distribuzioni, consulta e. [Implementazione in e Account AWS VPCs Visualizzazione delle informazioni sulla distribuzione](#)

Vantaggi delle azioni di implementazione

L'utilizzo delle azioni di distribuzione all'interno di un flusso di lavoro offre i seguenti vantaggi:

- **Cronologia di implementazione:** visualizza una cronologia delle distribuzioni per facilitare la gestione e la comunicazione delle modifiche al software distribuito.
- **Tracciabilità:** monitora lo stato delle tue implementazioni tramite la CodeCatalyst console e scopri quando e dove è stata distribuita ogni revisione dell'applicazione.
- **Rollback:** ripristina automaticamente le distribuzioni in caso di errori. Puoi anche configurare gli allarmi per attivare i rollback di distribuzione.
- **Monitoraggio:** monitora l'avanzamento della distribuzione nelle varie fasi del flusso di lavoro.
- **Integrazione con altre CodeCatalyst funzionalità:** archivia il codice sorgente e quindi crealo, testalo e distribuisilo, il tutto da un'unica applicazione.

Alternative all'implementazione delle azioni

Non è necessario utilizzare le azioni di distribuzione, sebbene siano consigliate perché offrono i vantaggi descritti nella sezione precedente. [Puoi invece utilizzare le seguenti azioni: CodeCatalyst](#)

- Un'azione di costruzione.

In genere, si utilizzano le azioni di compilazione se si desidera eseguire la distribuzione su una destinazione per la quale non esiste un'azione di distribuzione corrispondente o se si desidera un maggiore controllo sulla procedura di distribuzione. Per ulteriori informazioni sull'utilizzo delle azioni di compilazione per distribuire risorse, consulta [Creazione con flussi di lavoro](#)

- Un'GitHub azione.

È possibile utilizzare un'[GitHub azione](#) all'interno CodeCatalyst di un flusso di lavoro per distribuire applicazioni e risorse (anziché un' CodeCatalystazione). Per informazioni su come utilizzare GitHub Actions all'interno di un CodeCatalyst flusso di lavoro, consulta [Integrazione con GitHub Actions](#)

Puoi anche utilizzare i seguenti AWS servizi per distribuire l'applicazione, se non desideri utilizzare un CodeCatalyst flusso di lavoro per farlo:

- AWS CodeDeploy — vedi [Cos'è CodeDeploy?](#)
- AWS CodeBuild e AWS CodePipeline — vedi [Cos'è AWS CodeBuild?](#) e [cos'è AWS CodePipeline?](#)

- CloudFormation — vedi [Cos'è CloudFormation?](#)

Utilizzo CodeDeploy CodeBuild e CodePipeline CloudFormation servizi per implementazioni aziendali complesse.

Argomenti

- [Distribuzione su Amazon ECS con un flusso di lavoro](#)
- [Implementazione su Amazon EKS con un flusso di lavoro](#)
- [Implementazione di uno stack CloudFormation](#)
- [Implementazione di un' AWS CDK app con un flusso di lavoro](#)
- [Avvio di un' AWS CDK app con un flusso di lavoro](#)
- [Pubblicazione di file su Amazon S3 con un flusso di lavoro](#)
- [Implementazione in e Account AWS VPCs](#)
- [Visualizzazione dell'URL dell'app nel diagramma del flusso di lavoro](#)
- [Rimozione di un obiettivo di distribuzione](#)
- [Monitoraggio dello stato di distribuzione tramite commit](#)
- [Visualizzazione dei log di distribuzione](#)
- [Visualizzazione delle informazioni sulla distribuzione](#)

Distribuzione su Amazon ECS con un flusso di lavoro

Questa sezione descrive come distribuire un'applicazione containerizzata in un cluster Amazon Elastic Container Service utilizzando un flusso di lavoro. CodeCatalyst A tale scopo, devi aggiungere l'azione Deploy to Amazon ECS al tuo flusso di lavoro. Questa azione registra un file di [definizione delle attività fornito](#) dall'utente. Al momento della registrazione, la definizione dell'attività viene istanziata dal [servizio Amazon ECS](#) in esecuzione nel cluster [Amazon ECS](#). «Istanziare una definizione di attività» equivale a distribuire un'applicazione in Amazon ECS.

Per utilizzare questa azione, è necessario disporre di un cluster Amazon ECS, di un servizio e di un file di definizione delle attività.

Per ulteriori informazioni su Amazon ECS, consulta la Amazon Elastic Container Service Developer Guide.

Tip

Per un tutorial che mostra come utilizzare l'azione Deploy to Amazon ECS, consulta [Tutorial: distribuire un'applicazione su Amazon ECS](#)

Tip

Per un esempio funzionante dell'azione Deploy to Amazon ECS, crea un progetto con l'API Node.js con AWS Fargate o l'API Java con blueprint. AWS Fargate Per ulteriori informazioni, consulta [Creare un progetto con un blueprint](#).

Argomenti

- [Immagine di runtime utilizzata dall'azione «Deploy to Amazon ECS»](#)
- [Tutorial: distribuire un'applicazione su Amazon ECS](#)
- [Aggiungere l'azione «Deploy to Amazon ECS»](#)
- [Variabili «Distribuisce su Amazon ECS»](#)
- [Azione «Distribuisce su Amazon ECS» YAML](#)

Immagine di runtime utilizzata dall'azione «Deploy to Amazon ECS»

L'azione Deploy to Amazon ECS viene eseguita su un'immagine di [novembre 2022](#). Per ulteriori informazioni, consulta [Immagini attive](#).

Tutorial: distribuire un'applicazione su Amazon ECS

In questo tutorial, imparerai come distribuire un'applicazione serverless in Amazon Elastic Container Service (Amazon ECS) utilizzando un flusso di lavoro, Amazon ECS e alcuni altri servizi. AWS L'applicazione distribuita è un semplice sito Web Hello World costruito su un'immagine Docker del server Web Apache. Il tutorial illustra il lavoro di preparazione richiesto, ad esempio la configurazione di un cluster, e quindi descrive come creare un flusso di lavoro per creare e distribuire l'applicazione.

Tip

Invece di seguire questo tutorial, puoi utilizzare un blueprint che esegua una configurazione completa di Amazon ECS per te. Dovrai utilizzare l'API Node.js con AWS Fargate o l'API

Java con blueprint. AWS Fargate Per ulteriori informazioni, consulta [Creare un progetto con un blueprint](#).

Argomenti

- [Prerequisiti](#)
- [Passaggio 1: configurare un AWS utente e AWS CloudShell](#)
- [Fase 2: distribuire un'applicazione segnaposto in Amazon ECS](#)
- [Fase 3: creare un repository di immagini Amazon ECR](#)
- [Fase 4: Creare ruoli AWS](#)
- [Fase 5: Aggiungere AWS ruoli a CodeCatalyst](#)
- [Fase 6: Creare un archivio di sorgenti](#)
- [Passaggio 7: Aggiungere i file sorgente](#)
- [Fase 8: Creare ed eseguire un flusso di lavoro](#)
- [Passaggio 9: apporta una modifica ai file sorgente](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare:

- È necessario uno CodeCatalyst spazio con un AWS account connesso. Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Nel tuo spazio, hai bisogno di un progetto vuoto chiamato:

```
codecatalyst-ecs-project
```

Usa l'opzione Inizia da zero per creare questo progetto.

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

- Nel tuo progetto, hai bisogno di un CodeCatalyst ambiente chiamato:

```
codecatalyst-ecs-environment
```

Configura questo ambiente come segue:

- Scegliete qualsiasi tipo, ad esempio Non di produzione.
- Connect il tuo AWS account.
- Per il ruolo IAM predefinito, scegli un ruolo qualsiasi. Specificherai un ruolo diverso in seguito.

Per ulteriori informazioni, consulta [Implementazione in e Account AWS VPCs](#).

Passaggio 1: configurare un AWS utente e AWS CloudShell

Il primo passaggio di questo tutorial consiste nel creare un utente in AWS IAM Identity Center e avviare un' AWS CloudShell istanza come tale utente. Per la durata di questo tutorial, CloudShell è il tuo computer di sviluppo ed è dove configuri AWS risorse e servizi. Elimina questo utente dopo aver completato il tutorial.

Note

Non usate il vostro utente root per questo tutorial. È necessario creare un utente separato, altrimenti potrebbero verificarsi problemi durante l'esecuzione di azioni nella AWS Command Line Interface (CLI) in un secondo momento.

Per ulteriori informazioni sugli utenti di IAM Identity Center e CloudShell, consulta la Guida per l'AWS IAM Identity Center utente e la Guida per AWS CloudShell l'utente.

Per creare un utente IAM Identity Center

1. Accedi a Console di gestione AWS e apri la AWS IAM Identity Center console all'indirizzo <https://console.aws.amazon.com/singlesignon/>.

Note

Assicurati di accedere utilizzando Account AWS il dispositivo connesso al tuo CodeCatalyst spazio. Puoi verificare quale account è connesso accedendo al tuo spazio e scegliendo la scheda Account AWS. Per ulteriori informazioni, consulta [Creare uno spazio](#).

2. Nel riquadro di navigazione, scegli Users (Utenti), quindi scegli Add user (Aggiungi utente).

3. In Nome utente, inserisci:

CodeCatalystECSUser

4. In Password, scegli Genera una password monouso da condividere con questo utente.
5. In Indirizzo e-mail e Conferma indirizzo e-mail, inserisci un indirizzo e-mail che non esista già in IAM Identity Center.
6. In Nome e cognome, inserisci:

CodeCatalystECSUser

7. In Nome visualizzato, mantieni il nome generato automaticamente:

CodeCatalystECSUser CodeCatalystECSUser

8. Scegli Next (Successivo).
9. Nella pagina Aggiungi utente ai gruppi, scegli Avanti.
10. Nella pagina Rivedi e aggiungi utente, esamina le informazioni e scegli Aggiungi utente.

Viene visualizzata una finestra di dialogo con password monouso.

11. Scegli Copia, quindi incolla le informazioni di accesso, inclusi l'URL del portale di AWS accesso e la password monouso.
12. Scegli Chiudi.

Per creare un set di autorizzazioni

Assegnerai questo set di autorizzazioni a un secondo momento. **CodeCatalystECSUser**

1. Nel riquadro di navigazione, scegli Set di autorizzazioni, quindi scegli Crea set di autorizzazioni.
2. Scegli Set di autorizzazioni predefinito, quindi seleziona AdministratorAccess. Questa politica fornisce autorizzazioni complete a tutti. Servizi AWS
3. Scegli Next (Successivo).
4. In Nome del set di autorizzazioni, inserisci:

CodeCatalystECSPermissionSet

5. Scegli Next (Successivo).

6. Nella pagina Rivedi e crea, esamina le informazioni e scegli Crea.

Per assegnare il set di autorizzazioni a CodeCatalyst ECSUser

1. Nel riquadro di navigazione, scegli Account AWS, quindi seleziona la casella di controllo accanto a Account AWS quella a cui hai attualmente effettuato l'accesso.
2. Scegli Assegna utenti o gruppi.
3. Scegli la scheda Users (Utenti);
4. Seleziona la casella di controllo accanto a CodeCatalystECSUser
5. Scegli Next (Successivo).
6. Seleziona la casella di controllo accanto a CodeCatalystECSPermissionSet.
7. Scegli Next (Successivo).
8. Controlla le informazioni e scegli Invia.

Ora le hai assegnate CodeCatalystECSUser e CodeCatalystECSPermissionSet Account AWS le hai assegnate insieme.

Per disconnetterti e accedere nuovamente come CodeCatalyst ECSUser

1. Prima di disconnetterti, assicurati di avere l'URL del portale di AWS accesso e il nome utente e la password monouso per CodeCatalystECSUser. In precedenza avresti dovuto copiare queste informazioni in un editor di testo.

 Note

Se non disponi di queste informazioni, vai alla pagina dei CodeCatalystECSUser dettagli in IAM Identity Center, scegli Reimposta password, Genera una password monouso [...] e Reimposta nuovamente la password per visualizzare le informazioni sullo schermo.

2. Uscire da AWS.
3. Incolla l'URL del portale di AWS accesso nella barra degli indirizzi del browser.
4. Accedi con il nome utente e la password monouso per CodeCatalystECSUser.
5. In Nuova password, inserisci una password e scegli Imposta nuova password.

Sullo schermo viene visualizzata una Account AWS casella.

6. Scegli Account AWS, quindi scegli il nome Account AWS a cui hai assegnato `l'CodeCatalystECSUser` e il set di autorizzazioni.
7. Accanto a `CodeCatalystECSPermissionSet`, scegli Console di gestione.

Console di gestione AWS Appare. Ora hai effettuato l'accesso `CodeCatalystECSUser` con le autorizzazioni appropriate.

Per avviare un'istanza AWS CloudShell

1. Quindi `CodeCatalystECSUser`, nella barra di navigazione in alto, scegli l' AWS icona



).

Console di gestione AWS Viene visualizzata la pagina principale di.

2. Nella barra di navigazione in alto, scegli l' AWS CloudShell icona



).

CloudShell si apre. Attendi che l' CloudShell ambiente venga creato.

Note

Se non vedi l' CloudShell icona, assicurati di trovarti in una [regione supportata da CloudShell](#). Questo tutorial presuppone che ti trovi nella regione degli Stati Uniti occidentali (Oregon).

Per verificare che sia installato AWS CLI

1. Nel CloudShell terminale, inserisci:

```
aws --version
```

2. Verifica che venga visualizzata una versione.

AWS CLI È già configurato per l'utente corrente `CodeCatalystECSUser`, quindi non è necessario configurare AWS CLI chiavi e credenziali, come avviene normalmente.

Fase 2: distribuire un'applicazione segnaposto in Amazon ECS

In questa sezione, distribuisce manualmente un'applicazione segnaposto in Amazon ECS. Questa applicazione segnaposto verrà sostituita dall'applicazione Hello World distribuita dal tuo flusso di lavoro. L'applicazione segnaposto è Apache Web Server.

Per ulteriori informazioni su Amazon ECS, consulta la Amazon Elastic Container Service Developer Guide.

Completa la seguente serie di procedure per distribuire l'applicazione placeholder.

Per creare il ruolo di esecuzione dell'attività

Questo ruolo concede ad Amazon ECS l'AWS Fargate autorizzazione a effettuare chiamate API per tuo conto.

1. Crea una politica di fiducia:
 - a. In AWS CloudShell, inserisci il seguente comando:

```
cat > codecatalyst-ecs-trust-policy.json
```

Nel terminale viene visualizzato un prompt lampeggiante. CloudShell

- b. Immettete il seguente codice al prompt:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- c. Posizionate il cursore dopo l'ultima parentesi riccia (). }
 - d. **Enter**Premete e poi **Ctrl+d** per salvare il file e uscire da cat.
2. Crea un ruolo di esecuzione dell'attività:

```
aws iam create-role \  
    --role-name codecatalyst-ecs-task-execution-role \  
    --assume-role-policy-document file:///codecatalyst-ecs-trust-policy.json
```

3. Allega la AmazonECSTaskExecutionRolePolicy policy AWS gestita al ruolo:

```
aws iam attach-role-policy \  
    --role-name codecatalyst-ecs-task-execution-role \  
    --policy-arn arn:aws:iam::aws:policy/service-role/  
AmazonECSTaskExecutionRolePolicy
```

4. Visualizza i dettagli del ruolo:

```
aws iam get-role \  
    --role-name codecatalyst-ecs-task-execution-role
```

5. Nota il "Arn": valore del ruolo, ad esempio `arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role`. Avrai bisogno di questo Amazon Resource Name (ARN) in un secondo momento.

Come creare un cluster Amazon ECS

Questo cluster conterrà l'applicazione placeholder Apache e, successivamente, l'applicazione Hello World.

1. Ad esempio CodeCatalystECSUser AWS CloudShell, crea un cluster vuoto:

```
aws ecs create-cluster --cluster-name codecatalyst-ecs-cluster
```

2. (Facoltativo) Verifica che il cluster sia stato creato correttamente:

```
aws ecs list-clusters
```

L'ARN del `codecatalyst-ecs-cluster` cluster dovrebbe apparire nell'elenco, a indicare che la creazione è avvenuta con successo.

Per creare un file di definizione delle attività

Il file di definizione dell'attività indica di eseguire l'immagine Docker del [server Web Apache 2.4](#) (`httpd:2.4`) da cui viene estratta. DockerHub

1. Ad esempio CodeCatalystECSUser AWS CloudShell, crea un file di definizione dell'attività:

```
cat > taskdef.json
```

2. Incolla il codice seguente al prompt:

```
{
  "executionRoleArn": "arn:aws:iam::111122223333:role/codecatalyst-ecs-task-  
execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": "httpd:2.4",
      "essential": true,
      "portMappings": [
        {
          "hostPort": 80,
          "protocol": "tcp",
          "containerPort": 80
        }
      ]
    }
  ],
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "cpu": "256",
  "family": "codecatalyst-ecs-task-def",
  "memory": "512",
  "networkMode": "awsvpc"
}
```

Nel codice precedente, sostituisci *arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role*

con l'ARN del ruolo di esecuzione dell'attività in cui hai annotato. [Per creare il ruolo di esecuzione dell'attività](#)

3. Posiziona il cursore dopo l'ultima parentesi riccia (). }
4. **Enter**Premete e poi **Ctrl+d** per salvare il file e uscire da cat.

Per registrare il file di definizione delle attività con Amazon ECS

1. Ad esempio CodeCatalystECSUser AWS CloudShell, registra la definizione dell'attività:

```
aws ecs register-task-definition \  
  --cli-input-json file://taskdef.json
```

2. (Facoltativo) Verifica che la definizione dell'attività sia stata registrata:

```
aws ecs list-task-definitions
```

La definizione dell'codecatalyst-ecs-task-defattività dovrebbe apparire nell'elenco.

Per creare il servizio Amazon ECS

Il servizio Amazon ECS esegue le attività (e i contenitori Docker associati) dell'applicazione placeholder Apache e, successivamente, dell'applicazione Hello World.

1. QuindiCodeCatalystECSUser, passa alla console Amazon Elastic Container Service se non l'hai già fatto.
2. Scegli il cluster che hai creato in precedenza,codecatalyst-ecs-cluster.
3. Nella scheda Servizi, scegli Crea.
4. Nella pagina Crea, procedi come segue:

- a. Mantieni tutte le impostazioni predefinite ad eccezione di quelle elencate di seguito.
- b. In Launch Type (Tipo di lancio), scegli FARGATE.
- c. In Definizione dell'attività, nell'elenco a discesa Famiglia, scegli:

codecatalyst-ecs-task-def

- d. Per il nome del servizio, inserisci:

codecatalyst-ecs-service

- e. Per Attività desiderate, inserisci:

3

In questo tutorial, ogni attività avvia un singolo contenitore Docker.

- f. Espandi la sezione Rete.
- g. Per VPC, scegli un VPC qualsiasi.
- h. Per le sottoreti, scegli una sottorete.

 Note

Specificate solo una sottorete. Questo è tutto ciò che serve per questo tutorial.

 Note

Se non disponi di un VPC e di una sottorete, creali. Consulta [Crea un VPC](#) e [Crea una sottorete nel tuo VPC nella Amazon VPC User Guide](#).

- i. Per Gruppo di sicurezza, scegli Crea un nuovo gruppo di sicurezza, quindi procedi come segue:
 - i. Per il nome del gruppo di sicurezza, inserisci:

```
codecatalyst-ecs-security-group
```
 - ii. Per la descrizione del gruppo di sicurezza, inserisci:

```
CodeCatalyst ECS security group
```
 - iii. Scegli Aggiungi regola. Per Tipo, scegli HTTP e per Source, scegli Anywhere.
 - j. In basso, scegli Crea.
 - k. Attendi che il servizio venga creato. Ciò può richiedere alcuni minuti.
5. Scegli la scheda Attività, quindi scegli il pulsante Aggiorna. Verifica che la colonna Ultimo stato di tutte e tre le attività sia impostata su In esecuzione.

(Facoltativo) Per verificare che l'applicazione segnaposto Apache sia in esecuzione

1. Nella scheda Attività, scegli una delle tre attività.
2. Nel campo IP pubblico, scegli Apri indirizzo.

Viene visualizzata una *It Works!* pagina. Ciò indica che il servizio Amazon ECS ha avviato con successo un'attività che ha avviato un contenitore Docker con l'immagine Apache.

A questo punto del tutorial, hai distribuito manualmente un cluster Amazon ECS, un servizio e una definizione di attività, oltre a un'applicazione segnaposto Apache. Con tutti questi elementi a disposizione, ora sei pronto per creare un flusso di lavoro che sostituirà l'applicazione segnaposto Apache con l'applicazione Hello World del tutorial.

Fase 3: creare un repository di immagini Amazon ECR

In questa sezione, crei un repository di immagini privato in Amazon Elastic Container Registry (Amazon ECR). Questo repository memorizza l'immagine Docker del tutorial che sostituirà l'immagine segnaposto Apache che hai distribuito in precedenza.

Per ulteriori informazioni su Amazon ECR, consulta la *Amazon Elastic Container Registry User Guide*.

Per creare un archivio di immagini in Amazon ECR

1. Ad esempio `CodeCatalystECSUser` AWS CloudShell, crea un repository vuoto in Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-ecs-image-repo
```

2. Visualizza i dettagli del repository Amazon ECR:

```
aws ecr describe-repositories \
    --repository-names codecatalyst-ecs-image-repo
```

3. Nota il `"repositoryUri"`: valore, ad esempio, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

Ne avrai bisogno in seguito quando aggiungerai il repository al tuo flusso di lavoro.

Fase 4: Creare ruoli AWS

In questa sezione, creerai i ruoli AWS IAM di cui il tuo CodeCatalyst flusso di lavoro avrà bisogno per funzionare. Questi ruoli sono:

- Ruolo di creazione: concede all'azione di CodeCatalyst compilazione (nel flusso di lavoro) l'autorizzazione ad accedere al tuo AWS account e scrivere su Amazon ECR e Amazon. EC2
- Ruolo di distribuzione: concede all'azione CodeCatalyst Deploy to ECS (nel flusso di lavoro) l'autorizzazione ad accedere al tuo account AWS , ad Amazon ECS e ad alcuni altri servizi. AWS

[Per ulteriori informazioni sui ruoli IAM, consulta i ruoli IAM nella Guida per l'utente.AWS Identity and Access Management](#)

Note

Per risparmiare tempo, puoi creare un singolo ruolo, chiamato `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo, anziché i due ruoli elencati in precedenza. Per ulteriori informazioni, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName*ruolo per il tuo account e il tuo spazio](#). Tieni presente che il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo ha autorizzazioni molto ampie che possono rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante. Questo tutorial presuppone che stiate creando i due ruoli elencati in precedenza.

Per creare i ruoli di compilazione e distribuzione, puoi utilizzare il Console di gestione AWS o il. AWS CLI

Console di gestione AWS

Per creare i ruoli di compilazione e distribuzione, completa la seguente serie di procedure.

Per creare un ruolo di costruzione

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.

- c. Nel riquadro di navigazione, scegli Policy.
- d. Scegli Create Policy (Crea policy).
- e. Scegliere la scheda JSON.
- f. Eliminare il codice esistente.
- g. Incolla il codice seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"

```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next: Review (Successivo: Rivedi).
- j. In Nome, inserisci:

codecatalyst-ecs-build-policy

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di costruzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione `codecatalyst-ecs-build-policy`, cerca e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per il nome del ruolo, inserisci:

`codecatalyst-ecs-build-role`

- i. Per la descrizione del ruolo, inserisci:

`CodeCatalyst ECS build role`

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di autorizzazioni e una politica di fiducia.

3. Ottieni l'ARN del ruolo di costruzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (`codecatalyst-ecs-build-role`).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN. perché sarà necessaria in seguito.

Per creare un ruolo di distribuzione

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - c. Nel riquadro di navigazione, scegli Policy.
 - d. Scegliere Create Policy (Crea policy).
 - e. Scegliere la scheda JSON.
 - f. Eliminare il codice esistente.
 - g. Incolla il codice seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
```



```

        "codedeploy:GetApplication",
        "codedeploy:GetDeployment",
        "codedeploy:GetDeploymentGroup",
        "codedeploy:ListApplications",
        "codedeploy:ListDeploymentGroups",
        "codedeploy:ListDeployments",
        "codedeploy:StopDeployment",
        "codedeploy:GetDeploymentTarget",
        "codedeploy:ListDeploymentTargets",
        "codedeploy:GetDeploymentConfig",
        "codedeploy:GetApplicationRevision",
        "codedeploy:RegisterApplicationRevision",
        "codedeploy:BatchGetApplicationRevisions",
        "codedeploy:BatchGetDeploymentGroups",
        "codedeploy:BatchGetDeployments",
        "codedeploy:BatchGetApplications",
        "codedeploy:ListApplicationRevisions",
        "codedeploy:ListDeploymentConfigs",
        "codedeploy:ContinueDeployment"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }, { "Action": [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": { "StringLike": { "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "codedeploy.amazonaws.com"
    ]
    }
  }
}
]]
}

```

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizzate il carattere jolly nell'informativa sulla politica delle risorse. È quindi possibile definire l'ambito della politica inserendo il nome della risorsa una volta che è disponibile.

```
"Resource": "*"

```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-ecs-deploy-policy

```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di distribuzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione `codecatalyst-ecs-deploy-policy`, cerca e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per il nome del ruolo, inserisci:

```
codecatalyst-ecs-deploy-role

```

- i. Per la descrizione del ruolo, inserisci:

```
CodeCatalyst ECS deploy role

```

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di distribuzione con una politica di fiducia.

3. Ottenere l'ARN del ruolo di distribuzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.

- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (`codecatalyst-ecs-deploy-role`).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN, perché sarà necessaria in seguito.

AWS CLI

Per creare i ruoli di compilazione e distribuzione, completa la seguente serie di procedure.

Per creare una politica di fiducia per entrambi i ruoli

Ad esempio `CodeCatalystECSUser` AWS CloudShell, crea un file di criteri di fiducia:

1. Crea il file:

```
cat > codecatalyst-ecs-trust-policy.json
```

2. Al prompt del terminale, incolla il seguente codice:
3. Posizionate il cursore dopo l'ultima parentesi riccia (`()`). `}`
4. **Enter** Premete e poi **Ctrl+d** per salvare il file e uscire da `cat`.

Per creare la politica di costruzione e costruire il ruolo

1. Crea la politica di compilazione:
 - a. Quindi `CodeCatalystECSUser` AWS CloudShell, crea un file di policy di compilazione:

```
cat > codecatalyst-ecs-build-policy.json
```

- b. Al prompt, inserisci il seguente codice:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Effect": "Allow",
        "Action": [
            "ecr:*",
            "ec2:*"
        ],
        "Resource": "*"
    }
}

```

- c. Posizionate il cursore dopo l'ultima parentesi riccia (). }
- d. **Enter**Premete e poi **Ctrl+d** per salvare il file e uscire da cat.

2. Aggiungi la politica di compilazione a AWS:

```

aws iam create-policy \
  --policy-name codecatalyst-ecs-build-policy \
  --policy-document file://codecatalyst-ecs-build-policy.json

```

3. Nell'output del comando, annota il "arn": valore, ad esempio `arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy`. Questo ARN ti servirà più tardi.
4. Crea il ruolo di costruzione e allega ad esso la policy di fiducia:

```

aws iam create-role \
  --role-name codecatalyst-ecs-build-role \
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json

```

5. Allega la policy di compilazione al ruolo di costruzione:

```

aws iam attach-role-policy \
  --role-name codecatalyst-ecs-build-role \
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy

```

Dove *arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy* viene sostituito dall'ARN della politica di compilazione che hai notato in precedenza.

6. Visualizza i dettagli del ruolo di compilazione:

```

aws iam get-role \
  --role-name codecatalyst-ecs-build-role

```

7. Nota il "Arn": valore del ruolo, ad esempio `arn:aws:iam::111122223333:role/codecatalyst-ecs-build-role`. Questo ARN ti servirà più tardi.

Per creare la politica di distribuzione e il ruolo di distribuzione

1. Crea una politica di distribuzione:
 - a. In AWS CloudShell, crea un file di policy di distribuzione:

```
cat > codecatalyst-ecs-deploy-policy.json
```

- b. Al prompt, inserisci il codice seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
```

```

        "codedeploy:GetApplication",
        "codedeploy:GetDeployment",
        "codedeploy:GetDeploymentGroup",
        "codedeploy:ListApplications",
        "codedeploy:ListDeploymentGroups",
        "codedeploy:ListDeployments",
        "codedeploy:StopDeployment",
        "codedeploy:GetDeploymentTarget",
        "codedeploy:ListDeploymentTargets",
        "codedeploy:GetDeploymentConfig",
        "codedeploy:GetApplicationRevision",
        "codedeploy:RegisterApplicationRevision",
        "codedeploy:BatchGetApplicationRevisions",
        "codedeploy:BatchGetDeploymentGroups",
        "codedeploy:BatchGetDeployments",
        "codedeploy:BatchGetApplications",
        "codedeploy:ListApplicationRevisions",
        "codedeploy:ListDeploymentConfigs",
        "codedeploy:ContinueDeployment"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }, { "Action": [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": { "StringLike": { "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "codedeploy.amazonaws.com"
    ] } }
  }
}
]]
}

```

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizzate il carattere jolly nell'informativa sulla politica delle risorse, quindi definite la politica inserendo il nome della risorsa dopo che è stata resa disponibile.

```
"Resource": "*" }
```

- c. Posizionate il cursore dopo l'ultima parentesi riccia (`)`. `}`
- d. **Enter**Premete e poi **Ctrl+d** per salvare il file e uscire da `cat`.

2. Aggiungi la policy di distribuzione a AWS:

```
aws iam create-policy \
  --policy-name codecatalyst-ecs-deploy-policy \
  --policy-document file://codecatalyst-ecs-deploy-policy.json
```

- 3. Nell'output del comando, annota il `"arn"`: valore della policy di distribuzione, ad esempio. `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` Questo ARN ti servirà più tardi.
- 4. Crea il ruolo di distribuzione e allega ad esso la policy di fiducia:

```
aws iam create-role \
  --role-name codecatalyst-ecs-deploy-role \
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

- 5. Allega la policy di distribuzione al ruolo di distribuzione, dove `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` viene sostituita con l'ARN della policy di distribuzione indicata in precedenza.

```
aws iam attach-role-policy \
  --role-name codecatalyst-ecs-deploy-role \
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy
```

- 6. Visualizza i dettagli del ruolo di distribuzione:

```
aws iam get-role \
  --role-name codecatalyst-ecs-deploy-role
```

- 7. Nota il `"Arn"`: valore del ruolo, `arn:aws:iam::111122223333:role/codecatalyst-ecs-deploy-role` ad esempio. Questo ARN ti servirà più tardi.

Fase 5: Aggiungere AWS ruoli a CodeCatalyst

In questo passaggio, aggiungi build role (`codecatalyst-ecs-build-role`) e deploy role (`codecatalyst-ecs-deploy-role`) alla connessione dell' CodeCatalyst account nel tuo spazio.

Per aggiungere ruoli, build e deploy alla connessione del tuo account

1. Nel CodeCatalyst, accedi al tuo spazio.
2. Seleziona AWS accounts (Account). Viene visualizzato un elenco di connessioni all'account.
3. Scegli la connessione all'account che rappresenta l' AWS account in cui hai creato i ruoli di compilazione e distribuzione.
4. Scegli Gestisci ruoli dalla console AWS di gestione.

Viene visualizzata la pagina Aggiungi ruolo IAM CodeCatalyst allo spazio Amazon. Potrebbe essere necessario effettuare il login per accedere alla pagina.

5. Seleziona Aggiungi un ruolo esistente che hai creato in IAM.

Viene visualizzato un elenco a discesa. L'elenco mostra tutti i ruoli IAM con una policy di fiducia che include i principali `codecatalyst-runner.amazonaws.com` di `codecatalyst.amazonaws.com` servizio.

6. Nell'elenco a discesa **`codecatalyst-ecs-build-role`**, scegli e scegli Aggiungi ruolo.

Note

Se vedi `The security token included in the request is invalid`, potrebbe essere perché non disponi delle autorizzazioni giuste. Per risolvere questo problema, esci e accedi nuovamente con l' AWS account che hai usato quando hai creato il tuo CodeCatalyst spazio. AWS

7. Scegli Aggiungi ruolo IAM, scegli Aggiungi un ruolo esistente che hai creato in IAM e, nell'elenco a discesa, scegli. `codecatalyst-ecs-deploy-role` Scegliere Add role (Aggiungi ruolo).

Ora hai aggiunto i ruoli di creazione e distribuzione al tuo spazio.

8. Copia il valore del nome CodeCatalyst visualizzato di Amazon. Questo valore ti servirà in seguito, durante la creazione del tuo flusso di lavoro.

Fase 6: Creare un archivio di sorgenti

In questo passaggio, creerai un repository di sorgenti in CodeCatalyst. Questo repository memorizza i file sorgente del tutorial, come il file di definizione delle attività.

Per ulteriori informazioni sui repository dei sorgenti, consulta [Creazione di un repository di sorgenti](#)

Per creare un archivio di sorgenti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto, `codecatalyst-ecs-project`.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Crea repository.
5. Nel nome del repository, inserisci:

```
codecatalyst-ecs-source-repository
```

6. Scegli Create (Crea).

Passaggio 7: Aggiungere i file sorgente

In questa sezione, aggiungi i file sorgente di Hello World al tuo CodeCatalyst repository, `codecatalyst-ecs-source-repository`. Sono costituiti da:

- Un `index.html` file: visualizza un messaggio Hello World nel browser.
- Un Dockerfile: descrive l'immagine di base da utilizzare per l'immagine Docker e i comandi Docker da applicarvi.
- Un `taskdef.json` file: definisce l'immagine Docker da utilizzare per l'avvio di attività nel cluster.

La struttura delle cartelle è la seguente:

```
.  
|- public-html  
|   |- index.html  
|- Dockerfile  
|- taskdef.json
```

 Note

Le seguenti istruzioni mostrano come aggiungere i file usando la CodeCatalyst console, ma puoi usare Git se preferisci. Per informazioni dettagliate, vedi [Clonazione di un repository di sorgenti](#).

Argomenti

- [index.html](#)
- [Dockerfile](#)
- [taskdef.json](#)

index.html

Il `index.html` file visualizza un messaggio Hello World nel browser.

Per aggiungere il file `index.html`

1. Nella CodeCatalyst console, vai al tuo repository di origine, `codecatalyst-ecs-source-repository`.
2. In File, scegli Crea file.
3. Per Nome file, inserisci:

```
public-html/index.html
```

 Important

Assicurati di includere il `public-html/` prefisso per creare una cartella con lo stesso nome. `index.html` Si prevede che si trovi in questa cartella.

4. Nella casella di testo, inserisci il seguente codice:

```
<html>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: black;
```

```
        text-align: center;
        color: white;
        font-family: Arial, Helvetica, sans-serif;
    }
</style>
</head>
<body>
    <h1>Hello World</h1>
</body>
</html>
```

5. Scegliete Commit, quindi scegliete nuovamente Commit.

`index.html` Viene aggiunto al tuo repository in una `public-html` cartella.

Dockerfile

Il Dockerfile descrive l'immagine Docker di base da utilizzare e i comandi Docker da applicarvi. [Per ulteriori informazioni sul Dockerfile, consulta il Dockerfile Reference.](#)

Il Dockerfile specificato qui indica di utilizzare l'immagine di base di Apache 2.4 (`httpd`). Include anche istruzioni per copiare un file sorgente chiamato `index.html` in una cartella sul server Apache che serve le pagine Web. L'`EXPOSE`istruzione nel Dockerfile indica a Docker che il contenitore è in ascolto sulla porta 80.

Per aggiungere il Dockerfile

1. Nel tuo repository di origine, scegli Crea file.
2. Per Nome file, inserisci:

Dockerfile

Non includere un'estensione di file.

Important

Il Dockerfile deve risiedere nella cartella principale del repository. Il `Docker build` comando del flusso di lavoro si aspetta che sia presente.

3. Nella casella di testo, inserisci il seguente codice:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

4. Scegliete Commit, quindi scegliete nuovamente Commit.

Il Dockerfile viene aggiunto al tuo repository.

taskdef.json

Il `taskdef.json` file che aggiungi in questo passaggio è lo stesso di quello già specificato con la seguente differenza: [Fase 2: distribuire un'applicazione segnaposto in Amazon ECS](#)

Invece di specificare un nome di immagine Docker codificato nel `image:` campo (`httpd:2.4`), la definizione dell'attività qui utilizza un paio di variabili per indicare l'immagine: `and`. `$REPOSITORY_URI $IMAGE_TAG` Queste variabili verranno sostituite con valori reali generati dall'azione di creazione del flusso di lavoro quando si esegue il flusso di lavoro in una fase successiva.

Per informazioni dettagliate sui parametri di definizione delle attività, consulta [Parametri di definizione delle attività](#) nella Amazon Elastic Container Service Developer Guide.

Per aggiungere il file `taskdef.json`

1. Nel tuo repository di origine, scegli Crea file.
2. Per Nome file, inserisci:

taskdef.json

3. Nella casella di testo, inserisci il seguente codice:

```
{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      # The $REPOSITORY_URI and $IMAGE_TAG variables will be replaced
      # by the workflow at build time (see the build action in the
      # workflow)
```

```
        "image": $REPOSITORY_URI:$IMAGE_TAG,
        "essential": true,
        "portMappings": [
            {
                "hostPort": 80,
                "protocol": "tcp",
                "containerPort": 80
            }
        ]
    },
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "networkMode": "awsvpc",
    "cpu": "256",
    "memory": "512",
    "family": "codecatalyst-ecs-task-def"
}
```

Nel codice precedente, sostituisci

`arn:aws:iam::account_ID:role/codecatalyst-ecs-task-execution-role`

con l'ARN del ruolo di esecuzione dell'attività in cui hai annotato. [Per creare il ruolo di esecuzione dell'attività](#)

4. Scegliete Commit, quindi scegliete nuovamente Commit.

Il `taskdef.json` file viene aggiunto al tuo repository.

Fase 8: Creare ed eseguire un flusso di lavoro

In questa fase, crei un flusso di lavoro che prende i tuoi file sorgente, li crea in un'immagine Docker e quindi distribuisce l'immagine nel tuo cluster Amazon ECS. Questa implementazione sostituisce l'applicazione placeholder Apache esistente.

Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

- Un'azione di compilazione (`BuildBackend`): all'attivazione, l'azione crea l'immagine Docker utilizzando il `Dockerfile` e la invia ad Amazon ECR. L'azione di compilazione aggiorna inoltre il `taskdef.json` con il valore di `image` campo corretto, quindi crea un elemento di output di questo file. Questo artefatto viene utilizzato come input per l'azione di distribuzione, che viene successiva.

Per ulteriori informazioni sull'azione di compilazione, consulta. [Creazione con flussi di lavoro](#)

- Un'azione di distribuzione (`DeployToECS`): al termine dell'azione di creazione, l'azione di distribuzione cerca l'artefatto di output generato dall'azione di compilazione (`TaskDefArtifact`), ne trova il `taskdef.json` contenuto e lo registra con il servizio Amazon ECS. Il servizio segue quindi le istruzioni contenute nel `taskdef.json` file per eseguire tre attività Amazon ECS e i contenitori Hello World Docker associati all'interno del cluster Amazon ECS.

Come creare un flusso di lavoro

1. Nella CodeCatalyst console, nel pannello di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli Crea flusso di lavoro.
3. Per Source repository, scegli `codecatalyst-ecs-source-repository`.
4. Per Branch, scegli `main`.
5. Scegli Create (Crea).
6. Eliminate il codice di esempio YAML.
7. Aggiungi il seguente codice YAML:

Note

Nel codice YAML che segue, puoi omettere le sezioni se lo `Connections:` desideri. Se ometti queste sezioni, devi assicurarti che il ruolo specificato nel campo `Default IAM role` nel tuo ambiente includa le autorizzazioni e le policy di fiducia di entrambi i ruoli descritte in. [Fase 5: Aggiungere AWS ruoli a CodeCatalyst](#) Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta. [Creazione di un ambiente](#)

```
Name: codecatalyst-ecs-workflow
SchemaVersion: 1.0
```

```

Triggers:
- Type: PUSH
  Branches:
    - main
Actions:
  BuildBackend:
    Identifier: aws/build@v1
    Environment:
      Name: codecatalyst-ecs-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-ecs-build-role
    Inputs:
      Sources:
        - WorkflowSource
      Variables:
        - Name: REPOSITORY_URI
          Value: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo
        - Name: IMAGE_TAG
          Value: ${WorkflowSource.CommitId}
    Configuration:
      Steps:
        #pre_build:
          - Run: echo Logging in to Amazon ECR...
          - Run: aws --version
          - Run: aws ecr get-login-password --region us-west-2 | docker login --username AWS --password-stdin 111122223333.dkr.ecr.us-west-2.amazonaws.com
        #build:
          - Run: echo Build started on `date`
          - Run: echo Building the Docker image...
          - Run: docker build -t $REPOSITORY_URI:latest .
          - Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
        #post_build:
          - Run: echo Build completed on `date`
          - Run: echo Pushing the Docker images...
          - Run: docker push $REPOSITORY_URI:latest
          - Run: docker push $REPOSITORY_URI:$IMAGE_TAG
          # Replace the variables in taskdef.json
          - Run: find taskdef.json -type f | xargs sed -i "s|\\$REPOSITORY_URI|$REPOSITORY_URI|g"
          - Run: find taskdef.json -type f | xargs sed -i "s|\\$IMAGE_TAG|$IMAGE_TAG|g"
          - Run: cat taskdef.json

```

```

    # The output artifact will be a zip file that contains a task definition
    file.
    Outputs:
      Artifacts:
        - Name: TaskDefArtifact
          Files:
            - taskdef.json
    DeployToECS:
    DependsOn:
      - BuildBackend
    Identifier: aws/ecs-deploy@v1
    Environment:
      Name: codecatalyst-ecs-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-ecs-deploy-role
    Inputs:
      Sources: []
      Artifacts:
        - TaskDefArtifact
    Configuration:
      region: us-west-2
      cluster: codecatalyst-ecs-cluster
      service: codecatalyst-ecs-service
      task-definition: taskdef.json

```

Nel codice precedente, sostituisci:

- Entrambe le istanze hanno *codecatalyst-ecs-environment* il nome dell'ambiente in cui avete creato. [Prerequisiti](#)
- Entrambe le istanze *codecatalyst-account-connection* con il nome visualizzato della connessione all'account. Il nome visualizzato potrebbe essere un numero. Per ulteriori informazioni, consulta [Fase 5: Aggiungere AWS ruoli a CodeCatalyst](#).
- *codecatalyst-ecs-build-role* con il nome del ruolo di costruzione in cui hai creato [Fase 4: Creare ruoli AWS](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo* (nella Value: proprietà) con l'URI del repository Amazon ECR in cui hai creato. [Fase 3: creare un repository di immagini Amazon ECR](#)

- `111122223333.dkr.ecr.us-west-2.amazonaws.com`(nel Run: `aws ecr` comando) con l'URI del repository Amazon ECR senza il suffisso dell'immagine (). `/codecatalyst-ecs-image-repo`
- `codecatalyst-ecs-deploy-role` con il nome del ruolo di distribuzione in cui hai creato.

[Fase 4: Creare ruoli AWS](#)

- Entrambe le istanze `us-west-2` con il tuo codice AWS regionale. Per un elenco dei codici regionali, consulta [Endpoint regionali](#) in. Riferimenti generali di AWS

Note

Se hai deciso di non creare ruoli, build and deploy, sostituisci «`codecatalyst-ecs-build-role`» `codecatalyst-ecs-deploy-role` con il nome del `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo. Per ulteriori informazioni su questo ruolo, consulta [Fase 4: Creare ruoli AWS](#).

Tip

Invece di utilizzare i sed comandi `find` e mostrati nel codice del flusso di lavoro precedente per aggiornare il repository e il nome dell'immagine, puoi utilizzare l'azione di definizione delle attività di Render Amazon ECS per questo scopo. Per ulteriori informazioni, consulta [Modifica della definizione di un'attività Amazon ECS](#).

8. (Facoltativo) Scegli Convalida per assicurarti che il codice YAML sia valido prima di eseguire il commit.
9. Scegli Applica.
10. Nella finestra di dialogo Conferma flusso di lavoro, inserisci quanto segue:
 - a. Per il messaggio di conferma, rimuovi il testo e inserisci:

Add first workflow

- b. Per Repository, scegli `codecatalyst-ecs-source-repository`.
- c. Per il nome del ramo, scegli principale.
- d. Scegli Applica.

Ora hai creato un flusso di lavoro. L'esecuzione di un flusso di lavoro viene avviata automaticamente a causa del trigger definito nella parte superiore del flusso di lavoro. In particolare, quando hai eseguito il commit (e inviato) del `workflow.yaml` file nell'archivio di origine, il trigger ha avviato l'esecuzione del flusso di lavoro.

Per visualizzare l'avanzamento del flusso di lavoro, esegui

1. Nel pannello di navigazione della CodeCatalyst console, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il flusso di lavoro che hai appena creato, `codecatalyst-ecs-workflow`
3. Scegli BuildBackenddi vedere lo stato di avanzamento della costruzione.
4. Scegli DeployToECS per vedere lo stato di avanzamento della distribuzione.

Per ulteriori informazioni sulla visualizzazione dei dettagli dell'esecuzione, consulta [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

Per verificare la distribuzione

1. Apri la console Amazon ECS classic all'indirizzo <https://console.aws.amazon.com/ecs/>.
2. Scegli il tuo cluster, `codecatalyst-ecs-cluster`.
3. Selezionare la scheda Tasks (Attività).
4. Scegli una delle tre attività.
5. Nel campo IP pubblico, scegli Apri indirizzo.

Nel browser viene visualizzata una pagina «Hello World» che indica che il servizio Amazon ECS ha distribuito correttamente l'applicazione.

Passaggio 9: apporta una modifica ai file sorgente

In questa sezione, apporti una modifica al `index.html` file nel tuo repository di origine. Questa modifica fa sì che il flusso di lavoro crei una nuova immagine Docker, la tagghi con un ID di commit, la invii ad Amazon ECR e la distribuisca in Amazon ECS.

Per modificare il file index.html

1. Nella CodeCatalyst console, nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine e quindi scegli il tuo repository, `codecatalyst-ecs-source-repository`
2. Scegli `public-html` e quindi `index.html`.

Vengono visualizzati i contenuti di `index.html`

3. Scegli Modifica.
4. Nella riga 14, modifica il `Hello World` testo in `Tutorial complete!`.
5. Scegli Commetti, quindi scegli di nuovo Commetti.

Il commit provoca l'avvio di un nuovo flusso di lavoro.

6. (Facoltativo) Vai alla pagina principale del tuo repository di origine, scegli Visualizza i commit, quindi annota l'ID di commit per la `index.html` modifica.
7. Guarda lo stato di avanzamento della distribuzione:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - b. Scegli `codecatalyst-ecs-workflow` di visualizzare l'ultima esecuzione.
 - c. Scegli BuildBackended DeployToECS per vedere lo stato di avanzamento del flusso di lavoro.
8. Verifica che l'applicazione sia stata aggiornata, come segue:
 - a. Apri la console Amazon ECS classic all'indirizzo <https://console.aws.amazon.com/ecs/>.
 - b. Scegli il tuo cluster, `codecatalyst-ecs-cluster`.
 - c. Selezionare la scheda Tasks (Attività).
 - d. Scegli una delle tre attività.
 - e. Nel campo IP pubblico, scegli Apri indirizzo.

Viene visualizzata una `Tutorial complete!` pagina.

9. (Facoltativo) In AWS, passa alla console Amazon ECR e verifica che la nuova immagine Docker sia stata etichettata con l'ID di commit del passaggio 6.

Eliminazione

Pulisci i file e i servizi utilizzati in questo tutorial per evitare che ti vengano addebitati dei costi.

Nel Console di gestione AWS, pulisci in questo ordine:

1. In Amazon ECS, esegui le seguenti operazioni:
 - a. Eliminarecodecatalyst-ecs-service.
 - b. Eliminarecodecatalyst-ecs-cluster.
 - c. Annullare la registrazionecodecatalyst-ecs-task-definition.
2. In Amazon ECR, eliminacodecatalyst-ecs-image-repo.
3. In Amazon EC2, eliminacodecatalyst-ecs-security-group.
4. In IAM Identity Center, elimina:
 - a. CodeCatalystECSUser
 - b. CodeCatalystECSPermissionSet

Nella CodeCatalyst console, esegui la pulizia come segue:

1. Eliminarecodecatalyst-ecs-workflow.
2. Eliminarecodecatalyst-ecs-environment.
3. Eliminarecodecatalyst-ecs-source-repository.
4. Eliminarecodecatalyst-ecs-project.

In questo tutorial, hai imparato a distribuire un'applicazione su un servizio Amazon ECS utilizzando un CodeCatalyst flusso di lavoro e un'azione Deploy to Amazon ECS.

Aggiungere l'azione «Deploy to Amazon ECS»

Utilizza le seguenti istruzioni per aggiungere l'azione Deploy to Amazon ECS al tuo flusso di lavoro.

Visual

Per aggiungere l'azione «Deploy to Amazon ECS» utilizzando l'editor visuale

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione Deploy to Amazon ECS ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Deploy to Amazon ECS. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprirne il riquadro di configurazione.
10. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, consulta. [Azione «Distribuisci su Amazon ECS» YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere l'azione «Deploy to Amazon ECS» utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione Deploy to Amazon ECS ed esegui una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Deploy to Amazon ECS. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprirne il riquadro di configurazione.
- 10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [Azione «Distribuisce su Amazon ECS» YAML](#)
- 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
- 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Variabili «Distribuisce su Amazon ECS»

L'azione Deploy to Amazon ECS produce e imposta le seguenti variabili in fase di esecuzione. Queste sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

Chiave	Valore
cluster	Il nome del cluster Amazon ECS su cui è stato distribuito durante l'esecuzione del flusso di lavoro. Ad esempio: <code>codecatalyst-ecs-cluster</code>
piattaforma di distribuzione	Il nome della piattaforma di distribuzione. Codificato in. <code>AWS:ECS</code>
service	Il nome del servizio Amazon ECS su cui è stato distribuito durante l'esecuzione del flusso di lavoro. Ad esempio: <code>codecatalyst-ecs-service</code>
task-definition-arn	L'Amazon Resource Name (ARN) della definizione dell'attività registrata durante l'esecuzione del flusso di lavoro. Ad esempio: <code>arn:aws:ecs:us-west-2:111122223333:task-definition/codecatalyst-task-def:8</code> L'8esempio precedente indica la revisione che è stata registrata.
url di distribuzione	Un collegamento alla scheda Eventi della console Amazon ECS, in cui è possibile visualizzare i dettagli della distribuzione di Amazon ECS associata all'esecuzione del flusso di lavoro. Ad esempio: <code>https://console.aws.amazon.com/ecs/home?regio</code>

Chiave	Valore
	n=us-west-2#/clusters/codecatalyst-ecs-cluster/services/codecatalyst-ecs-service/events
region	Il codice regionale su Regione AWS cui è stato distribuito durante l'esecuzione del flusso di lavoro. Ad esempio: us-west-2

Azione «Distribuisci su Amazon ECS» YAML

Di seguito è riportata la definizione YAML dell'azione Deploy to Amazon ECS. Per informazioni su come utilizzare questa azione, consulta [Distribuzione su Amazon ECS con un flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSDeployAction\_nn:
  Identifier: aws/ecs-deploy@v1
  DependsOn:
    - build-action
```



```

Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
Timeout: timeout-minutes
Environment:
  Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
Inputs:
  # Specify a source or an artifact, but not both.
  Sources:
    - source-name-1
  Artifacts:
    - task-definition-artifact
Configuration:
  region: us-east-1
  cluster: ecs-cluster
  service: ecs-service
  task-definition: task-definition-path
  force-new-deployment: false|true
  codedeploy-appspec: app-spec-file-path
  codedeploy-application: application-name
  codedeploy-deployment-group: deployment-group-name
  codedeploy-deployment-description: deployment-description

```

ECSDeployAction

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: `ECSDeployAction_nn`.

Interfaccia utente corrispondente: scheda di configurazione/nome visualizzato dell'azione

Identifier

(*ECSDeployAction*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: aws/ecs-deploy@v1.

Interfaccia utente corrispondente: Workflow diagram/ ECSDeploy Action_NN/ aws/ecs-deploy @v1 label

DependsOn

(*ECSDeployAction*/DependsOn)

(Facoltativo)

Specificare un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente per eseguire questa azione.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Input/Dipende da - opzionale

Compute

(*ECSDeployAction*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*ECSDeployAction*/Compute/Type)

(Richiesto [Compute](#) se incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: configurazione tab/Advanced : opzionale/tipo di calcolo

Fleet

(*ECSDeployAction*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand:,. Linux.x86-64.Large Linux.x86-64.XLarge Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere. [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omesso, l'impostazione predefinita è. Linux.x86-64.Large

Interfaccia utente corrispondente: Configurazione tab/Advanced : parco opzionale/di calcolo

Timeout

(*ECSDeployAction*/Timeout)

(Facoltativo)

Specificate il periodo di tempo, in minuti (editor YAML) o in ore e minuti (editor visivo), che l'azione può essere eseguita prima che termini l'azione. CodeCatalyst Il minimo è 5 minuti e il massimo è descritto in. [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Environment

(*ECSDeployAction*/Environment)

(Obbligatorio)

Specificare l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs e Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*ECSDeployAction*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*ECSDeployAction*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' '/ connessione all'account Environment/account/role AWS

Name

(*ECSDeployAction*/Environment/Connections/Name)

(Richiesto se incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' '/ connessione all'account Environment/account/role AWS

Role

(*ECSDeployAction*/Environment/Connections/Role)

(Richiesto se incluso [Connections](#))

Specificare il nome del ruolo IAM a cui l'azione Deploy to Amazon ECS utilizza per accedere. AWS Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

- La seguente politica di autorizzazione:

 Warning

Limita le autorizzazioni a quelle mostrate nella seguente politica. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
```

```

        "codedeploy:CreateApplication",
        "codedeploy:CreateDeployment",
        "codedeploy:CreateDeploymentGroup",
        "codedeploy:GetApplication",
        "codedeploy:GetDeployment",
        "codedeploy:GetDeploymentGroup",
        "codedeploy:ListApplications",
        "codedeploy:ListDeploymentGroups",
        "codedeploy:ListDeployments",
        "codedeploy:StopDeployment",
        "codedeploy:GetDeploymentTarget",
        "codedeploy:ListDeploymentTargets",
        "codedeploy:GetDeploymentConfig",
        "codedeploy:GetApplicationRevision",
        "codedeploy:RegisterApplicationRevision",
        "codedeploy:BatchGetApplicationRevisions",
        "codedeploy:BatchGetDeploymentGroups",
        "codedeploy:BatchGetDeployments",
        "codedeploy:BatchGetApplications",
        "codedeploy:ListApplicationRevisions",
        "codedeploy:ListDeploymentConfigs",
        "codedeploy:ContinueDeployment"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }, { "Action": [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": { "StringLike": { "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "codedeploy.amazonaws.com"
    ]
    }
  }
}
}]
}

```

Note

La prima volta che il ruolo viene utilizzato, utilizza il seguente carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"

```

- La seguente politica di fiducia personalizzata:

Note

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' / Ruolo Environment/account/role

Inputs

(*ECSDeployAction*/Inputs)

(Facoltativo)

La Inputs sezione definisce i dati ECSDeployAction necessari durante l'esecuzione di un flusso di lavoro.

 Note

È consentito un solo input (fonte o artefatto) per azione Deploy to Amazon ECS.

Interfaccia utente corrispondente: scheda Input

Sources

(*ECSDeployAction*/Inputs/Sources)

(Richiesto se il file di definizione delle attività è archiviato in un repository di origine)

Se il file di definizione delle attività è archiviato in un repository di origine, specifica l'etichetta di tale repository di origine. Attualmente, l'unica etichetta supportata è `WorkflowSource`

Se il file di definizione dell'attività non è contenuto in un repository di origine, deve risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*ECSDeployAction*/Inputs/Artifacts)

(Richiesto se il file di definizione dell'attività è memorizzato in un elemento [di output di un'azione precedente](#))

Se il file di definizione dell'attività che desideri distribuire è contenuto in un artefatto generato da un'azione precedente, specifica l'artefatto qui. Se il file di definizione dell'attività non è contenuto in un artefatto, deve risiedere nel repository di origine.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda di configurazione/artefatti - opzionale

Configuration

(*ECSDeployAction*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

region

(Configuration/region)

(Obbligatorio)

Specificare la AWS regione in cui risiedono il cluster e il servizio Amazon ECS. Per un elenco dei codici regionali, consulta [Endpoint regionali](#) nel. Riferimenti generali di AWS

Interfaccia utente corrispondente: scheda/regione di configurazione

cluster

(*ECSDeployAction*/Configuration/cluster)

(Obbligatorio)

Specificare il nome di un cluster Amazon ECS esistente. L'azione Deploy to Amazon ECS distribuirà la tua applicazione containerizzata come attività in questo cluster. Per ulteriori informazioni sui cluster Amazon ECS, consulta [Clusters](#) nella Amazon Elastic Container Service Developer Guide.

Interfaccia utente corrispondente: scheda Configurazione/Cluster

service

(*ECSDeployAction*/Configuration/service)

(Obbligatorio)

Specificare il nome di un servizio Amazon ECS esistente che creerà un'istanza del file di definizione delle attività. Questo servizio deve risiedere nel cluster specificato nel campo. `cluster` Per ulteriori informazioni sui servizi Amazon ECS, consulta i servizi [Amazon ECS](#) nella Amazon Elastic Container Service Developer Guide.

Interfaccia utente corrispondente: scheda di configurazione/servizio

task-definition

(*ECSDeployAction*/Configuration/task-definition)

(Obbligatorio)

Specificare il percorso di un file di definizione delle attività esistente. Se il file si trova nel repository di origine, il percorso è relativo alla cartella principale del repository di origine. Se il file si trova in un elemento di un'azione precedente del flusso di lavoro, il percorso è relativo alla cartella principale dell'elemento. Per ulteriori informazioni sui file di definizione delle attività, consulta [Definizioni delle attività](#) nella Amazon Elastic Container Service Developer Guide.

Interfaccia utente corrispondente: scheda di configurazione/definizione dell'attività

force-new-deployment

(*ECSDeployAction*/Configuration/force-new-deployment)

(Obbligatorio)

Se abilitato, il servizio Amazon ECS è in grado di avviare nuove distribuzioni senza modifiche alla definizione del servizio. Forzare una distribuzione fa sì che il servizio interrompa tutte le attività attualmente in esecuzione e avvii nuove attività. Per ulteriori informazioni su come forzare nuove distribuzioni, consulta [Updating a service](#) nella Amazon Elastic Container Service Developer Guide.

Impostazione predefinita: false

Interfaccia utente corrispondente: scheda Configurazione/ Forza una nuova distribuzione del servizio

codedeploy-appspec

(*ECSDeployAction*/Configuration/codedeploy-appspec)

(Obbligatorio se hai configurato il servizio Amazon ECS per utilizzare le blue/green distribuzioni, altrimenti omettilo)

Specificare il nome e il percorso di un file di specifiche CodeDeploy dell'applicazione () esistente. AppSpec Questo file deve risiedere nella directory principale del repository di CodeCatalyst origine. Per ulteriori informazioni sui AppSpec file, consultate i file [delle specifiche CodeDeploy dell'applicazione \(AppSpec\) nella Guida](#) per l'AWS CodeDeploy utente.

Note

Fornisci CodeDeploy informazioni solo se hai configurato il servizio Amazon ECS per eseguire blue/green distribuzioni. Per le distribuzioni di aggiornamento in sequenza (impostazione predefinita), ometti le informazioni. CodeDeploy Per ulteriori informazioni sulle

distribuzioni di Amazon ECS, consulta i tipi di [distribuzione di Amazon ECS nella](#) Amazon Elastic Container Service Developer Guide.

 Note

I CodeDeploy campi possono essere nascosti nell'editor visuale. Per farli apparire, vedi [Perché mancano CodeDeploy dei campi nell'editor visivo?](#).

Interfaccia utente corrispondente: scheda Configurazione/ CodeDeploy AppSpec

codedeploy-application

(*ECSDeployAction*/Configuration/codedeploy-application)

(Richiesto se codedeploy-appspec incluso)

Specificate il nome di un' CodeDeploy applicazione esistente. Per ulteriori informazioni sulle CodeDeploy applicazioni, vedere [Lavorare con le applicazioni CodeDeploy nella](#) Guida per l'AWS CodeDeploy utente.

Interfaccia utente corrispondente: scheda di configurazione/applicazione CodeDeploy

codedeploy-deployment-group

(*ECSDeployAction*/Configuration/codedeploy-deployment-group)

(Richiesto se codedeploy-appspec incluso)

Specificare il nome di un gruppo CodeDeploy di distribuzione esistente. Per ulteriori informazioni sui gruppi di CodeDeploy distribuzione, vedere [Lavorare con i gruppi di distribuzione CodeDeploy nella](#) Guida per l'AWS CodeDeploy utente.

Interfaccia utente corrispondente: scheda di configurazione/gruppo CodeDeploy di distribuzione

codedeploy-deployment-description

(*ECSDeployAction*/Configuration/codedeploy-deployment-description)

(Facoltativo)

Specificare una descrizione della distribuzione che verrà creata con questa azione. Per ulteriori informazioni, consulta [Lavorare con le distribuzioni CodeDeploy nella Guida](#) per l'AWS CodeDeploy utente.

Interfaccia utente corrispondente: scheda di configurazione/descrizione della distribuzione CodeDeploy

Implementazione su Amazon EKS con un flusso di lavoro

Tip

Per un tutorial che mostra come utilizzare l'azione del cluster Deploy to Kubernetes, consulta. [Tutorial: distribuzione di un'applicazione su Amazon EKS](#)

Questa sezione descrive come distribuire un'applicazione containerizzata in un cluster Kubernetes utilizzando un flusso di lavoro. CodeCatalyst A tale scopo, è necessario aggiungere l'azione Deploy to Kubernetes cluster al flusso di lavoro. Questa azione distribuisce la tua applicazione in un cluster Kubernetes che hai configurato in Amazon Elastic Kubernetes Service (EKS) utilizzando uno o più file manifest Kubernetes. Per [deployment.yaml](#) un [Tutorial: distribuzione di un'applicazione su Amazon EKS](#) manifesto di esempio, consulta in.

[Per ulteriori informazioni su Kubernetes, consulta la documentazione di Kubernetes.](#)

Per ulteriori informazioni su Amazon EKS, consulta [Cos'è Amazon EKS?](#) nella Guida per l'utente di Amazon EKS.

Argomenti

- [Come funziona l'azione «Deploy to Kubernetes cluster»](#)
- [Immagine di runtime utilizzata dall'azione «Deploy to Amazon EKS»](#)
- [Tutorial: distribuzione di un'applicazione su Amazon EKS](#)
- [Aggiungere l'azione «Deploy to Kubernetes cluster»](#)
- [Variabili «Distribuisci su cluster Kubernetes»](#)
- [Azione «Distribuisci su cluster Kubernetes» YAML](#)

Come funziona l'azione «Deploy to Kubernetes cluster»

Il cluster Deploy to Kubernetes funziona come segue:

1. In fase di esecuzione, l'azione installa l'utilità Kubernetes sulla macchina di elaborazione su cui è in esecuzione l'azione CodeCatalyst. L'azione si configura in modo da puntare al cluster Amazon EKS fornito al momento della configurazione dell'azione. L'utilità è necessaria per eseguire il `kubectl apply` comando, quindi.
2. L'azione esegue il `kubectl apply -f my-manifest.yaml` comando, che esegue le istruzioni `my-manifest.yaml` per distribuire l'applicazione come set di contenitori e pod nel cluster configurato. Per ulteriori informazioni su questo comando, consulta l'argomento [kubectl apply](#) nella documentazione di riferimento di Kubernetes.

Immagine di runtime utilizzata dall'azione «Deploy to Amazon EKS»

L'azione Deploy to Amazon EKS viene eseguita su un'[immagine di novembre 2022](#). Per ulteriori informazioni, consulta [Immagini attive](#).

Tutorial: distribuzione di un'applicazione su Amazon EKS

In questo tutorial, imparerai come distribuire un'applicazione containerizzata in Amazon Elastic Kubernetes Service utilizzando un flusso di lavoro Amazon, CodeCatalyst Amazon EKS e alcuni altri servizi. AWS L'applicazione distribuita è un semplice «Hello, World!» sito web costruito su un'immagine Docker del server web Apache. Il tutorial illustra il lavoro di preparazione richiesto, ad esempio la configurazione di una macchina di sviluppo e di un cluster Amazon EKS, quindi descrive come creare un flusso di lavoro per creare l'applicazione e distribuirla nel cluster.

Una volta completata la distribuzione iniziale, il tutorial ti spiega come apportare una modifica all'origine dell'applicazione. Questa modifica fa sì che una nuova immagine Docker venga creata e inserita nel tuo archivio di immagini Docker con nuove informazioni di revisione. La nuova revisione dell'immagine Docker viene quindi distribuita in Amazon EKS.

Tip

Invece di seguire questo tutorial, puoi utilizzare un blueprint che esegua una configurazione completa di Amazon EKS per te. Dovrai utilizzare il blueprint EKS App Deployment. Per ulteriori informazioni, consulta [Creare un progetto con un blueprint](#).

Argomenti

- [Prerequisiti](#)
- [Passaggio 1: configura la tua macchina di sviluppo](#)

- [Fase 2: creare un cluster Amazon EKS](#)
- [Fase 3: creare un repository di immagini Amazon ECR](#)
- [Passaggio 4: Aggiungere i file sorgente](#)
- [Fase 5: Creare AWS ruoli](#)
- [Fase 6: Aggiungere AWS ruoli a CodeCatalyst](#)
- [Fase 7: Aggiornare il ConfigMap](#)
- [Passaggio 8: creare ed eseguire un flusso di lavoro](#)
- [Passaggio 9: apporta una modifica ai file sorgente](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare questo tutorial:

- È necessario uno CodeCatalyst spazio Amazon con un AWS account connesso. Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Nel tuo spazio, hai bisogno di un progetto vuoto chiamato:

```
codecatalyst-eks-project
```

Usa l'opzione Inizia da zero per creare questo progetto.

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

- Nel tuo progetto, hai bisogno di un repository di CodeCatalyst sorgenti vuoto chiamato:

```
codecatalyst-eks-source-repository
```

Per ulteriori informazioni, consulta [Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst](#).

- Nel tuo progetto, hai bisogno di un ambiente CodeCatalyst CI/CD (non un ambiente di sviluppo) chiamato:

```
codecatalyst-eks-environment
```

Configura questo ambiente come segue:

- Scegli qualsiasi tipo, ad esempio Non di produzione.
- Connect il tuo AWS account.
- Per il ruolo IAM predefinito, scegli un ruolo qualsiasi. Specificherai un ruolo diverso in seguito.

Per ulteriori informazioni, consulta [Implementazione in e Account AWS VPCs](#).

Passaggio 1: configura la tua macchina di sviluppo

Il primo passo di questo tutorial è configurare una macchina di sviluppo con alcuni strumenti che utilizzerai durante questo tutorial. Questi strumenti sono:

- l'`eksctl` utilità: per la creazione di cluster
- l'`kubectl` utilità: un prerequisito per `eksctl`
- il AWS CLI — anche un prerequisito per `eksctl`

Puoi installare questi strumenti sulla tua macchina di sviluppo esistente, se ne hai una, oppure puoi utilizzare un ambiente di CodeCatalyst sviluppo, basato su cloud. Il vantaggio di un ambiente di CodeCatalyst sviluppo è che è facile da avviare e smontare ed è integrato con altri CodeCatalyst servizi, consentendoti di completare questo tutorial in meno passaggi.

Questo tutorial presuppone che utilizzerai un ambiente di CodeCatalyst sviluppo.

Le seguenti istruzioni descrivono un modo rapido per avviare un ambiente di CodeCatalyst sviluppo e configurarlo con gli strumenti richiesti, ma se desideri istruzioni dettagliate, consulta:

- [Creazione di un ambiente di sviluppo](#) in questa guida.
- [Installazione di kubectl nella Guida](#) per l'utente di Amazon EKS.
- [Installazione o aggiornamento di eksctl nella Amazon EKS](#) User Guide.
- [Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l'utente](#).AWS Command Line Interface

Per avviare un ambiente di sviluppo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo progetto, `codecatalyst-eks-project`.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

4. Scegli il nome del tuo repository di origine, `codecatalyst-eks-source-repository`
5. In alto scegli Create Dev Environment, quindi scegli AWS Cloud9 (nel browser).
6. Assicurati che le opzioni Work in existing branch e main siano selezionate, quindi scegli Crea.

Il tuo ambiente di sviluppo viene avviato in una nuova scheda del browser e il tuo repository (`codecatalyst-eks-source-repository`) viene clonato al suo interno.

Per installare e configurare kubectl

1. Nel terminale Dev Environment, inserisci:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
```

2. Inserisci:

```
chmod +x ./kubectl
```

3. Inserisci:

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
```

4. Inserisci:

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

5. Inserisci:

```
kubectl version --short --client
```

6. Verifica che venga visualizzata una versione.

Ora hai installato `kubectl`.

Per installare e configurare eksctl

Note

`eksctl` non è strettamente necessario perché è possibile `kubectl` utilizzarlo al suo posto. Tuttavia, `eksctl` ha il vantaggio di automatizzare gran parte della configurazione del cluster ed è quindi lo strumento consigliato per questo tutorial.

1. Nel terminale Dev Environment, inserisci:

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

2. Inserisci:

```
sudo cp /tmp/eksctl /usr/bin
```

3. Inserisci:

```
eksctl version
```

4. Verifica che venga visualizzata una versione.

Ora hai installato `eksctl`.

Per verificare che AWS CLI sia installato

1. Nel terminale Dev Environment, inserisci:

```
aws --version
```

2. Verificate che appaia una versione per verificare che AWS CLI sia installata.

Completa le procedure rimanenti per configurarlo AWS CLI con le autorizzazioni necessarie per l'accesso AWS.

Per configurare il AWS CLI

È necessario configurarlo AWS CLI con chiavi di accesso e un token di sessione per consentirgli di accedere ai AWS servizi. Le seguenti istruzioni forniscono un modo rapido per configurare le chiavi e il token, ma se desideri istruzioni dettagliate, consulta [Configurazione del AWS CLI](#) nella Guida per l'AWS Command Line Interface utente.

1. Crea un utente IAM Identity Center, come segue:

- a. Accedi a Console di gestione AWS e apri la AWS IAM Identity Center console all'indirizzo <https://console.aws.amazon.com/singlesignon/>.

(Potresti dover scegliere Abilita se non hai mai effettuato l'accesso a IAM Identity Center prima.)

 Note

Assicurati di accedere utilizzando Account AWS il dispositivo connesso al tuo CodeCatalyst spazio. Puoi verificare quale account è connesso accedendo al tuo spazio e scegliendo la scheda Account AWS. Per ulteriori informazioni, consulta [Creare uno spazio](#).

- b. Nel riquadro di navigazione, scegli Users (Utenti), quindi scegli Add user (Aggiungi utente).
- c. In Nome utente, inserisci:

codecatalyst-eks-user

- d. In Password, scegli Genera una password monouso da condividere con questo utente.
- e. In Indirizzo e-mail e Conferma indirizzo e-mail, inserisci un indirizzo e-mail che non esista già in IAM Identity Center.
- f. In Nome, inserisci:

codecatalyst-eks-user

- g. In Cognome, inserisci:

codecatalyst-eks-user

- h. In Nome visualizzato, mantieni:

codecatalyst-eks-user codecatalyst-eks-user

- i. Scegli Next (Successivo).
- j. Nella pagina Aggiungi utente ai gruppi, scegli Avanti.
- k. Nella pagina Rivedi e aggiungi utente, esamina le informazioni e scegli Aggiungi utente.

Viene visualizzata una finestra di dialogo con password monouso.

- l. Scegli Copia, quindi incolla le informazioni di accesso in un file di testo. Le informazioni di accesso sono costituite dall'URL del portale di AWS accesso, da un nome utente e da una password monouso.
- m. Scegli Chiudi.

2. Crea un set di autorizzazioni, come segue:

- a. Nel riquadro di navigazione, scegli Set di autorizzazioni, quindi scegli Crea set di autorizzazioni.
- b. Scegli Set di autorizzazioni predefinito, quindi seleziona AdministratorAccess. Questa politica fornisce autorizzazioni complete a tutti. Servizi AWS
- c. Scegli Next (Successivo).
- d. In Nome del set di autorizzazioni, rimuovi AdministratorAccess e inserisci:

codecatalyst-eks-permission-set

- e. Scegli Next (Successivo).
 - f. Nella pagina Rivedi e crea, esamina le informazioni e scegli Crea.
3. Assegna il set di autorizzazioni a `codecatalyst-eks-user`, come segue:
- a. Nel riquadro di navigazione, scegli Account AWS, quindi seleziona la casella di controllo accanto a Account AWS quella a cui hai attualmente effettuato l'accesso.
 - b. Scegli Assegna utenti o gruppi.
 - c. Scegli la scheda Users (Utenti);
 - d. Seleziona la casella di controllo accanto a `codecatalyst-eks-user`
 - e. Scegli Next (Successivo).
 - f. Seleziona la casella di controllo accanto a `codecatalyst-eks-permission-set`.
 - g. Scegli Next (Successivo).
 - h. Controlla le informazioni e scegli Invia.

Ora le hai assegnate `codecatalyst-eks-user` e `codecatalyst-eks-permission-set` Account AWS le hai assegnate insieme.

4. Le chiavi `codecatalyst-eks-user` di accesso e il token di sessione di Ottain, come segue:

- a. Assicurati di avere l'URL del portale di AWS accesso e il nome utente e la password monouso per `codecatalyst-eks-user`. In precedenza avreste dovuto copiare queste informazioni in un editor di testo.

 Note

Se non disponi di queste informazioni, vai alla pagina dei `codecatalyst-eks-user` dettagli in IAM Identity Center, scegli Reimposta password, Genera una password monouso [...] e Reimposta nuovamente la password per visualizzare le informazioni sullo schermo.

- b. Esci da AWS.
- c. Incolla l'URL del portale di AWS accesso nella barra degli indirizzi del browser.
- d. Accedi con:

- Nome utente:

`codecatalyst-eks-user`

- Password:

one-time-password

- e. In Imposta nuova password, inserisci una nuova password e scegli Imposta nuova password.

Sullo schermo viene visualizzata una Account AWS casella.

- f. Scegli Account AWS, quindi scegli il nome Account AWS a cui hai assegnato l'`codecatalyst-eks-user` utente e il set di autorizzazioni.
- g. Accanto `codecatalyst-eks-permission-set`, scegli Riga di comando o Accesso programmatico.
- h. Copia i comandi al centro della pagina. Sono simili ai seguenti:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
```

```
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... *session-token* dov'è una lunga stringa casuale.

5. Aggiungi le chiavi di accesso e il token di sessione a AWS CLI, come segue:

- a. Tornate al vostro ambiente di CodeCatalyst sviluppo.
- b. Al prompt del terminale, incolla i comandi che hai copiato. Premere Invio.

Ora lo hai configurato AWS CLI con chiavi di accesso e un token di sessione. Ora puoi AWS CLI utilizzarlo per completare le attività richieste da questo tutorial.

 Important

Se in qualsiasi momento durante questo tutorial visualizzi messaggi simili a:
Unable to locate credentials. You can configure credentials by running "aws configure".

O:

ExpiredToken: The security token included in the request is expired

... è perché la tua AWS CLI sessione è scaduta. In questo caso, non eseguire il `aws configure` comando. Utilizzate invece le istruzioni riportate nel passaggio 4 di questa procedura, che inizia con, `Obtain codecatalyst-eks-user's access key and session token` per aggiornare la sessione.

Fase 2: creare un cluster Amazon EKS

In questa sezione, crei un cluster in Amazon EKS. Le istruzioni riportate di seguito descrivono un modo rapido per creare il cluster utilizzando `eksctl`, ma se desideri istruzioni dettagliate, consulta:

- [Guida introduttiva a eksctl](#) nella Amazon EKS User Guide

or

- Guida [introduttiva alla console e AWS CLI](#) nella Amazon EKS User Guide (questo argomento fornisce `kubectl` istruzioni per la creazione del cluster)

 Note

I [cluster privati](#) non sono supportati dall' CodeCatalyst integrazione con Amazon EKS.

Prima di iniziare

Assicurati di aver completato le seguenti attività sulla tua macchina di sviluppo:

- Installata l'`eksctl` utilità.
- Installata l'`kubectl` utilità.
- L'ha installata AWS CLI e configurata con chiavi di accesso e un token di sessione.

Per informazioni su come completare queste attività, vedere [Passaggio 1: configura la tua macchina di sviluppo](#).

Come creare un cluster

 Important

Non utilizzare l'interfaccia utente del servizio Amazon EKS per creare il cluster perché il cluster non verrà configurato correttamente. Usa l'`eksctl` utilità, come descritto nei passaggi seguenti.

1. Vai al tuo ambiente di sviluppo.
2. Crea un cluster e dei nodi:

```
eksctl create cluster --name codecatalyst-eks-cluster --region us-west-2
```

Dove:

- *codecatalyst-eks-cluster* viene sostituito con il nome che vuoi assegnare al cluster.
- *us-west-2* viene sostituito dalla tua regione.

Dopo 10-20 minuti, viene visualizzato un messaggio simile al seguente:

```
EKS cluster "codecatalyst-eks-cluster" in "us-west-2" region is ready
```

 Note

Vedrai più `waiting for CloudFormation stack` messaggi durante la AWS creazione del cluster. Si tratta di un comportamento normale.

3. Verifica che il cluster sia stato creato correttamente:

```
kubectl cluster-info
```

Verrà visualizzato un messaggio simile al seguente, che indica l'avvenuta creazione del cluster:

```
Kubernetes master is running at https://long-string.gr7.us-west-2.eks.amazonaws.com
CoreDNS is running at https://long-string.gr7.us-west-2.eks.amazonaws.com/api/v1/
namespaces/kube-system/services/kube-dns:dns/proxy
```

Fase 3: creare un repository di immagini Amazon ECR

In questa sezione, crei un repository di immagini privato in Amazon Elastic Container Registry (Amazon ECR). Questo repository memorizza l'immagine Docker per il tutorial.

Per ulteriori informazioni su Amazon ECR, consulta la [Amazon Elastic Container Registry User Guide](#).

Per creare un archivio di immagini in Amazon ECR

1. Accedi al tuo ambiente di sviluppo.
2. Crea un repository vuoto in Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-eks-image-repo
```

Sostituisci *codecatalyst-eks-image-repo* con il nome che desideri assegnare al repository Amazon ECR.

Questo tutorial presuppone che tu abbia dato un nome al tuo repository. `codecatalyst-eks-image-repo`

3. Visualizza i dettagli del repository Amazon ECR:


```
aws ecr describe-repositories \
  --repository-names codecatalyst-eks-image-repo
```

4. Nota il “repositoryUri”: valore, ad esempio, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo`

Ne avrai bisogno in seguito quando aggiungerai il repository al tuo flusso di lavoro.

Passaggio 4: Aggiungere i file sorgente

In questa sezione, aggiungete i file sorgente dell'applicazione al vostro repository di origine (`codecatalyst-eks-source-repository`). Sono costituiti da:

- Un `index.html` file: mostra un messaggio «Hello, World!» messaggio nel browser.
- Un `Dockerfile`: descrive l'immagine di base da utilizzare per l'immagine Docker e i comandi Docker da applicarvi.
- Un `deployment.yaml` file: il manifesto di Kubernetes che definisce il servizio e la distribuzione Kubernetes.

La struttura delle cartelle è la seguente:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
    | |– index.html
  |– Dockerfile
```

Argomenti

- [index.html](#)
- [Dockerfile](#)
- [deployment.yaml](#)

index.html

Il `index.html` file mostra un messaggio «Hello, World!» messaggio nel browser.

Per aggiungere il file index.html

1. Vai al tuo ambiente di sviluppo.
2. In `codecatalyst-eks-source-repository`, crea una cartella chiamata `public-html`.
3. In `public-html`, crea un file chiamato `index.html` con il seguente contenuto:

```
<html>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: black;
        text-align: center;
        color: white;
        font-family: Arial, Helvetica, sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

4. Alla riga di comando del terminale, digitate:

```
cd /projects/codecatalyst-eks-source-repository
```

5. Aggiungi, esegui il commit e invia:

```
git add .
git commit -m "add public-html/index.html"
git push
```

`index.html` Viene aggiunto al tuo repository in una `public-html` cartella.

Dockerfile

Il Dockerfile descrive l'immagine Docker di base da utilizzare e i comandi Docker da applicarvi. [Per ulteriori informazioni sul Dockerfile, consulta il Dockerfile Reference.](#)

Il Dockerfile specificato qui indica di utilizzare l'immagine di base di Apache 2.4 (`httpd`). Include anche istruzioni per copiare un file sorgente chiamato `index.html` in una cartella sul server Apache che serve le pagine Web. L'istruzione `EXPOSE` nel Dockerfile indica a Docker che il contenitore è in ascolto sulla porta 80.

Per aggiungere il Dockerfile

1. In `incodecatalyst-eks-source-repository`, crea un file chiamato `Dockerfile` con il seguente contenuto:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

Non includere un'estensione di file.

 Important

Il Dockerfile deve risiedere nella cartella principale del repository. Il comando `Docker build` del flusso di lavoro si aspetta che sia presente.

2. Aggiungi, esegui il commit e invia:

```
git add .
git commit -m "add Dockerfile"
git push
```

Il Dockerfile viene aggiunto al tuo repository.

`deployment.yaml`

In questa sezione, aggiungi un file al tuo repository. `deployment.yaml` Il `deployment.yaml` file è un manifesto di Kubernetes che definisce due tipi o tipi di risorse Kubernetes da eseguire: un «servizio» e una «distribuzione».

- Il «servizio» implementa un sistema di bilanciamento del carico in Amazon EC2. Il sistema di bilanciamento del carico ti fornisce un URL pubblico con accesso a Internet e una porta standard (porta 80) che puoi utilizzare per accedere a «Hello, World!» applicazione.

- La «distribuzione» utilizza tre pod e ogni pod conterrà un contenitore Docker con il comando «Hello, World!» applicazione. I tre pod vengono distribuiti sui nodi creati al momento della creazione del cluster.

Il manifesto di questo tutorial è breve; tuttavia, un manifesto può includere qualsiasi tipo di risorsa Kubernetes, come pod, job, ingress e policy di rete. Inoltre, puoi utilizzare più file manifest se la distribuzione è complessa.

Per aggiungere un file deployment.yaml

1. In codecatalyst-eks-source-repository, crea una cartella chiamata. Kubernetes
2. In/Kubernetes, crea un file chiamato deployment.yaml con il seguente contenuto:

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-app
spec:
  type: LoadBalancer
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  labels:
    app: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
```

```
labels:
  app: my-app
spec:
  containers:
  - name: codecatalyst-eks-container
    # The $REPOSITORY_URI and $IMAGE_TAG placeholders will be replaced by
    actual values supplied by the build action in your workflow
    image: $REPOSITORY_URI:$IMAGE_TAG
    ports:
    - containerPort: 80
```

3. Aggiungi, esegui il commit e invia:

```
git add .
git commit -m "add Kubernetes/deployment.yaml"
git push
```

Il `deployment.yaml` file viene aggiunto al tuo repository in una cartella chiamata `Kubernetes`.

Ora hai aggiunto tutti i tuoi file sorgente.

Prenditi un momento per ricontrollare il tuo lavoro e assicurati di aver inserito tutti i file nelle cartelle corrette. La struttura delle cartelle è la seguente:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
    | |– index.html
  |– Dockerfile
```

Fase 5: Creare AWS ruoli

In questa sezione, creerai i ruoli AWS IAM di cui il tuo CodeCatalyst flusso di lavoro avrà bisogno per funzionare. Questi ruoli sono:

- Ruolo di creazione: concede all'azione di CodeCatalyst compilazione (nel flusso di lavoro) l'autorizzazione ad accedere al tuo AWS account e scrivere su Amazon ECR e Amazon. EC2
- Ruolo di distribuzione: concede all'azione del cluster CodeCatalyst Deploy to Kubernetes (nel flusso di lavoro) l'autorizzazione ad accedere al tuo account e ad Amazon EKS. AWS

[Per ulteriori informazioni sui ruoli IAM, consulta i ruoli IAM nella Guida per l'utente.AWS Identity and Access Management](#)

Note

Per risparmiare tempo, puoi creare un singolo ruolo, chiamato `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo, anziché i due ruoli elencati in precedenza. Per ulteriori informazioni, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo ha autorizzazioni molto ampie che possono rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante. Questo tutorial presuppone che stiate creando i due ruoli elencati in precedenza.

Per creare i ruoli di compilazione e distribuzione, completa la seguente serie di procedure.

1. Per creare una politica di fiducia per entrambi i ruoli

1. Vai al tuo ambiente di sviluppo.
2. Nella Cloud9-*long-string* directory, crea un file chiamato `codecatalyst-eks-trust-policy.json` con il seguente contenuto:

2. Per creare la politica di compilazione per il ruolo di compilazione

- Nella Cloud9-*long-string* directory, create un file chiamato `codecatalyst-eks-build-policy.json` con il seguente contenuto:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
```

```

        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}

```

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizzate il carattere jolly nell'informativa sulla politica delle risorse, quindi definite la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

3. Per creare la politica di distribuzione per il ruolo di distribuzione

- Nella Cloud9-*long-string* directory, create un file chiamato `codecatalyst-eks-deploy-policy.json` con il seguente contenuto:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}

```

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizzate il carattere jolly nell'informativa sulla politica delle risorse, quindi definite la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

Ora hai aggiunto tre documenti di policy al tuo Dev Environment. La struttura delle cartelle ora è simile a questa:

```
|– Cloud9-long-string
  |– .c9
  |– codecatalyst-eks-source-repository
    |– Kubernetes
    |– public-html
    |– Dockerfile
  codecatalyst-eks-build-policy.json
  codecatalyst-eks-deploy-policy.json
  codecatalyst-eks-trust-policy.json
```

4. Per aggiungere la politica di compilazione a AWS

1. Nel terminale Dev Environment, inserisci:

```
cd /projects
```

2. Inserisci:

```
aws iam create-policy \
  --policy-name codecatalyst-eks-build-policy \
  --policy-document file://codecatalyst-eks-build-policy.json
```

3. Premere Invio.
4. Nell'output del comando, annota il "arn": valore, ad esempio `arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy`. Questo ARN ti servirà più tardi.

5. Per aggiungere la politica di distribuzione a AWS

1. Inserisci:

```
aws iam create-policy \  
  --policy-name codecatalyst-eks-deploy-policy \  
  --policy-document file://codecatalyst-eks-deploy-policy.json
```

2. Premere Invio.

3. Nell'output del comando, annota il "arn": valore della policy di distribuzione, `arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy` ad esempio. Questo ARN ti servirà più tardi.

6. Per creare il ruolo di costruzione

1. Inserisci:

```
aws iam create-role \  
  --role-name codecatalyst-eks-build-role \  
  --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Premere Invio.

3. Inserisci:

```
aws iam attach-role-policy \  
  --role-name codecatalyst-eks-build-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy
```

Dove *arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy* viene sostituito dall'ARN della politica di compilazione che hai notato in precedenza.

4. Premere Invio.

5. Alla riga di comando del terminale, inserisci:

```
aws iam get-role \  
  --role-name codecatalyst-eks-build-role
```

6. Premere Invio.

7. Annotate il "Arn": valore del ruolo, `arn:aws:iam::111122223333:role/codecatalyst-eks-build-role` ad esempio. Questo ARN ti servirà più tardi.

7. Per creare il ruolo di distribuzione

1. Inserisci:

```
aws iam create-role \  
    --role-name codecatalyst-eks-deploy-role \  
    --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Premere Invio.

3. Inserisci:

```
aws iam attach-role-policy \  
    --role-name codecatalyst-eks-deploy-role \  
    --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy
```

Dove *arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy* viene sostituito dall'ARN della politica di distribuzione indicata in precedenza.

4. Premere Invio.

5. Inserisci:

```
aws iam get-role \  
    --role-name codecatalyst-eks-deploy-role
```

6. Premere Invio.

7. Nota il "Arn": valore del ruolo, ad esempio `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role`. Questo ARN ti servirà più tardi.

Ora hai creato i ruoli di compilazione e distribuzione e li hai annotati. ARNs

Fase 6: Aggiungere AWS ruoli a CodeCatalyst

In questo passaggio, aggiungi build role (`codecatalyst-eks-build-role`) e deploy role (`codecatalyst-eks-deploy-role`) a Account AWS quello che hai collegato al tuo spazio. In questo modo i ruoli sono disponibili per l'uso nel flusso di lavoro.

Per aggiungere ruoli di creazione e distribuzione al tuo Account AWS

1. Nella CodeCatalyst console, accedi al tuo spazio.
2. In alto, scegli Impostazioni.
3. Nel riquadro di navigazione, scegli AWS Account. Viene visualizzato un elenco di account.
4. Nella colonna Amazon CodeCatalyst display name, copia il nome visualizzato del Account AWS luogo in cui hai creato i ruoli di build e deploy. (Potrebbe essere un numero). Questo valore ti servirà in seguito, durante la creazione del flusso di lavoro.
5. Scegli il nome visualizzato.
6. Scegli Gestisci ruoli dalla console AWS di gestione.

Viene visualizzata la pagina Aggiungi ruolo IAM CodeCatalyst allo spazio Amazon. Potrebbe essere necessario effettuare il login per accedere alla pagina.

7. Seleziona Aggiungi un ruolo esistente che hai creato in IAM.

Viene visualizzato un elenco a discesa. L'elenco mostra i ruoli di compilazione e distribuzione e tutti gli altri ruoli IAM con una policy di fiducia che include i `codecatalyst-runner.amazonaws.com` principali di servizio. `codecatalyst.amazonaws.com`

8. Dall'elenco a discesa, aggiungi:
 - `codecatalyst-eks-build-role`
 - `codecatalyst-eks-deploy-role`

Note

Se vedi `The security token included in the request is invalid`, potrebbe essere perché non disponi delle autorizzazioni giuste. Per risolvere questo problema, esci e accedi nuovamente con l' AWS account che hai usato quando hai creato il tuo CodeCatalyst spazio. AWS

9. Torna alla CodeCatalyst console e aggiorna la pagina.

I ruoli di compilazione e distribuzione dovrebbero ora apparire nei ruoli IAM.

Questi ruoli sono ora disponibili per l'uso nei flussi CodeCatalyst di lavoro.

Fase 7: Aggiornare il ConfigMap

Devi aggiungere il ruolo di distribuzione che hai creato nel ConfigMap file Kubernetes [Fase 5: Creare AWS ruoli](#) per consentire all'azione del cluster Deploy to Kubernetes (nel tuo flusso di lavoro) la possibilità di accedere e interagire con il cluster. È possibile utilizzare o eseguire questa attività.

```
eksctl kubectl
```

Per configurare il file Kubernetes ConfigMap usando eksctl

- Nel terminale Dev Environment, inserisci:

```
eksctl create iamidentitymapping --cluster codecatalyst-eks-cluster --  
arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role --group  
system:masters --username codecatalyst-eks-deploy-role --region us-west-2
```

Dove:

- *codecatalyst-eks-cluster* viene sostituito con il nome del cluster Amazon EKS.
- *arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role* viene sostituito dall'ARN del ruolo di distribuzione in cui è stato creato. [Fase 5: Creare AWS ruoli](#)
- *codecatalyst-eks-deploy-role* (accanto a --username) viene sostituito dal nome del ruolo di distribuzione in cui è stato creato. [Fase 5: Creare AWS ruoli](#)

Note

Se hai deciso di non creare un ruolo di distribuzione, sostituiscilo *codecatalyst-eks-deploy-role* con il nome del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo. Per ulteriori informazioni su questo ruolo, consulta [Fase 5: Creare AWS ruoli](#).

- *us-west-2* viene sostituito con la tua regione.

Per i dettagli su questo comando, consulta [Gestire gli utenti e i ruoli IAM](#).

Viene visualizzato un messaggio simile al seguente:

```
2023-06-09 00:58:29 [#] checking arn arn:aws:iam::111122223333:role/codecatalyst-  
eks-deploy-role against entries in the auth ConfigMap
```

```
2023-06-09 00:58:29 [#] adding identity "arn:aws:iam::111122223333:role/
codecatalyst-eks-deploy-role" to auth ConfigMap
```

Per configurare il file Kubernetes ConfigMap usando kubectl

1. Nel terminale Dev Environment, inserisci:

```
kubectl edit configmap -n kube-system aws-auth
```

Il ConfigMap file appare sullo schermo.

2. Aggiungi il testo in corsivo rosso:

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file
# will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
      rolearn: arn:aws:iam::111122223333:role/eksctl-codecatalyst-eks-cluster-n-
NodeInstanceRole-16BC456ME6YR5
      username: system:node:{{EC2PrivateDNSName}}
    - groups:
      - system:masters
      rolearn: arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role
      username: codecatalyst-eks-deploy-role
  mapUsers: |
    []
kind: ConfigMap
metadata:
  creationTimestamp: "2023-06-08T19:04:39Z"
  managedFields:
  ...
```

Dove:

- `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role` viene sostituito dall'ARN del ruolo di distribuzione in cui è stato creato. [Fase 5: Creare AWS ruoli](#)
- `codecatalyst-eks-deploy-role` (accanto a `username:`) viene sostituito dal nome del ruolo di distribuzione in cui è stato creato. [Fase 5: Creare AWS ruoli](#)

 Note

Se hai deciso di non creare un ruolo di distribuzione, sostituiscilo `codecatalyst-eks-deploy-role` con il nome del `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo. Per ulteriori informazioni su questo ruolo, consulta [Fase 5: Creare AWS ruoli](#).

Per i dettagli, consulta [Abilitazione dell'accesso principale IAM al tuo cluster](#) nella Guida per l'utente di Amazon EKS.

Ora hai assegnato al ruolo di distribuzione e, per estensione, all'azione `Deploy to Amazon EKS`, `system:masters` le autorizzazioni per il tuo cluster Kubernetes.

Passaggio 8: creare ed eseguire un flusso di lavoro

In questo passaggio, crei un flusso di lavoro che prende i tuoi file sorgente, li crea in un'immagine Docker e quindi distribuisce l'immagine in tree pod nel tuo cluster Amazon EKS.

Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione di compilazione (`BuildBackend`): all'attivazione, l'azione crea l'immagine Docker utilizzando il Dockerfile e la invia ad Amazon ECR. L'azione di compilazione aggiorna anche `$IMAGE_TAG` le variabili `$REPOSITORY_URI` and nel `deployment.yaml` file con i valori corretti, quindi crea un elemento di output di questo file e di qualsiasi altro file nella cartella. Kubernetes In questo tutorial, l'unico file nella Kubernetes cartella è, `deployment.yaml` ma è possibile includerne altri. L'artefatto viene utilizzato come input per l'azione di distribuzione, che viene successiva.

Per ulteriori informazioni sull'azione di creazione, consulta. [Creazione con flussi di lavoro](#)

- A deploy action (DeployToEKS): al termine dell'azione di compilazione, l'azione deploy cerca l'artefatto di output generato dall'azione di compilazione (Manifests) e trova il `deployment.yaml` file al suo interno. L'azione segue quindi le istruzioni contenute nel `deployment.yaml` file per eseguire tre pod, ciascuno contenente un singolo «Hello, World!» Contenitore Docker: all'interno del tuo cluster Amazon EKS.

Come creare un flusso di lavoro

1. Vai alla console. CodeCatalyst
2. Vai al tuo progetto (`codecatalyst-eks-project`).
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli Crea flusso di lavoro.
5. Per Source repository, scegli `codecatalyst-eks-source-repository`.
6. Per Branch, scegli `main`.
7. Scegli Create (Crea).
8. Elimina il codice di esempio YAML.
9. Aggiungi il seguente codice YAML per creare un nuovo file di definizione del flusso di lavoro:

 Note

Per ulteriori informazioni sul file di definizione del flusso di lavoro, vedere. [Definizione YAML del flusso di lavoro](#)

 Note

Nel codice YAML che segue, puoi omettere le `Connections`: sezioni se lo desideri. Se ometti queste sezioni, devi assicurarti che il ruolo specificato nel campo Default IAM role nel tuo ambiente includa le autorizzazioni e le policy di fiducia di entrambi i ruoli descritte in. [Fase 6: Aggiungere AWS ruoli a CodeCatalyst](#) Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta. [Creazione di un ambiente](#)

Name: codecatalyst-eks-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH

Branches:

- main

Actions:

BuildBackend:

Identifier: aws/build@v1

Environment:

Name: *codecatalyst-eks-environment*

Connections:

- Name: *codecatalyst-account-connection*
- Role: *codecatalyst-eks-build-role*

Inputs:

Sources:

- WorkflowSource

Variables:

- Name: REPOSITORY_URI

Value: *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo*

- Name: IMAGE_TAG

Value: \${WorkflowSource.CommitId}

Configuration:

Steps:

#pre_build:

- Run: echo Logging in to Amazon ECR...
- Run: aws --version
- Run: aws ecr get-login-password --region *us-west-2* | docker login --username AWS --password-stdin *111122223333.dkr.ecr.us-west-2.amazonaws.com*

#build:

- Run: echo Build started on `date`
- Run: echo Building the Docker image...
- Run: docker build -t \$REPOSITORY_URI:latest .
- Run: docker tag \$REPOSITORY_URI:latest \$REPOSITORY_URI:\$IMAGE_TAG

#post_build:

- Run: echo Build completed on `date`
- Run: echo Pushing the Docker images...
- Run: docker push \$REPOSITORY_URI:latest
- Run: docker push \$REPOSITORY_URI:\$IMAGE_TAG
- # Replace the variables in deployment.yaml


```

- Run: find Kubernetes/ -type f | xargs sed -i "s|\\$REPOSITORY_URI|
$REPOSITORY_URI|g"
- Run: find Kubernetes/ -type f | xargs sed -i "s|\\$IMAGE_TAG|$IMAGE_TAG|g"
- Run: cat Kubernetes/*
# The output artifact will be a zip file that contains Kubernetes manifest
files.
Outputs:
  Artifacts:
    - Name: Manifests
      Files:
        - "Kubernetes/*"
DeployToEKS:
  DependsOn:
    - BuildBackend
Identifier: aws/kubernetes-deploy@v1
Environment:
  Name: codecatalyst-eks-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-eks-deploy-role
Inputs:
  Artifacts:
    - Manifests
Configuration:
  Namespace: default
  Region: us-west-2
  Cluster: codecatalyst-eks-cluster
  Manifests: Kubernetes/

```

Nel codice precedente, sostituisci:

- Entrambe le istanze hanno *codecatalyst-eks-environment* il nome dell'ambiente in cui avete creato. [Prerequisiti](#)
- Entrambe le istanze *codecatalyst-account-connection* con il nome visualizzato della connessione all'account. Il nome visualizzato potrebbe essere un numero. Per ulteriori informazioni, consulta [Fase 6: Aggiungere AWS ruoli a CodeCatalyst](#).
- *codecatalyst-eks-build-role* con il nome del ruolo di costruzione in cui hai creato [Fase 5: Creare AWS ruoli](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo* (nella Value: proprietà) con l'URI del repository Amazon ECR in cui hai creato. [Fase 3: creare un repository di immagini Amazon ECR](#)

- `111122223333.dkr.ecr.us-west-2.amazonaws.com`(nel Run: `aws ecr` comando) con l'URI del repository Amazon ECR senza il suffisso dell'immagine (). `/codecatalyst-eks-image-repo`
- `codecatalyst-eks-deploy-role` con il nome del ruolo di distribuzione in cui hai creato. [Fase 5: Creare AWS ruoli](#)
- Entrambe le istanze `us-west-2` con il tuo codice AWS regionale. Per un elenco dei codici regionali, consulta [Endpoint regionali](#) in. Riferimenti generali di AWS

Note

Se hai deciso di non creare ruoli, build and deploy, sostituisci «`codecatalyst-eks-build-role`» `codecatalyst-eks-deploy-role` con il nome del `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo. Per ulteriori informazioni su questo ruolo, consulta [Fase 5: Creare AWS ruoli](#).

10. (Facoltativo) Scegli Convalida per assicurarti che il codice YAML sia valido prima di eseguire il commit.
11. Scegli Applica.
12. Nella finestra di dialogo Conferma flusso di lavoro, inserisci quanto segue:
 - a. Per il messaggio di conferma, rimuovi il testo e inserisci:

Add first workflow
 - b. Per Repository, scegli `codecatalyst-eks-source-repository`.
 - c. Per il nome del ramo, scegli principale.
 - d. Scegli Applica.

Ora hai creato un flusso di lavoro. L'esecuzione di un flusso di lavoro viene avviata automaticamente a causa del trigger definito nella parte superiore del flusso di lavoro. In particolare, quando hai eseguito il commit (e inviato) del `workflow.yaml` file nell'archivio di origine, il trigger ha avviato l'esecuzione del flusso di lavoro.

Per visualizzare l'avanzamento del flusso di lavoro, esegui

1. Nel pannello di navigazione della CodeCatalyst console, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il flusso di lavoro che hai appena creato, `codecatalyst-eks-workflow`.
3. Scegli BuildBackenddi vedere lo stato di avanzamento della costruzione.
4. Scegli DeployToEKS per vedere lo stato di avanzamento dell'implementazione.

Per ulteriori informazioni sulla visualizzazione dei dettagli dell'esecuzione, consulta [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

Per verificare la distribuzione

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. A sinistra, in basso, scegli Load Balancers.
3. Seleziona il load balancer che è stato creato come parte della tua implementazione Kubernetes. Se non sei sicuro di quale bilanciamento del carico scegliere, cerca i seguenti tag nella scheda Tag:
 - `kubernetes.io/service-name`
 - `kubernetes.io/cluster/ekstutorialcluster`
4. Con il bilanciamento del carico corretto selezionato, scegli la scheda Descrizione.
5. Copia e incolla il valore del nome DNS nella barra degli indirizzi del browser.

Il messaggio «Hello, World!» nel browser viene visualizzata una pagina Web che indica che l'applicazione è stata distribuita correttamente.

Passaggio 9: apporta una modifica ai file sorgente

In questa sezione, apporti una modifica al `index.html` file nel tuo repository di origine. Questa modifica fa sì che il flusso di lavoro crei una nuova immagine Docker, la tagghi con un ID di commit, la invii ad Amazon ECR e la distribuisca in Amazon ECS.

Per modificare il file `index.html`

1. Vai al tuo ambiente di sviluppo.
2. Al prompt del terminale, passa al repository dei sorgenti:

```
cd /projects/codecatalyst-eks-source-repository
```

3. Recupera le ultime modifiche al flusso di lavoro:

```
git pull
```

4. Aprire `codecatalyst-eks-source-repository/public-html/index.html`.
5. Nella riga 14, modifica il `Hello, World!` testo in `Tutorial complete!`.
6. Aggiungi, conferma e invia:

```
git add .  
git commit -m "update index.html title"  
git push
```

L'esecuzione di un flusso di lavoro viene avviata automaticamente.

7. (Facoltativo) Inserisci:

```
git show HEAD
```

Annota l'ID di commit per la `index.html` modifica. Questo ID di commit verrà taggato nell'immagine Docker che verrà distribuita dall'esecuzione del flusso di lavoro appena avviata.

8. Guarda lo stato di avanzamento della distribuzione:
 - a. Nella CodeCatalyst console, nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - b. Scegli `codecatalyst-eks-workflow` di visualizzare l'ultima esecuzione.
 - c. Scegliete `BuildBackended DeployToEKS` per vedere lo stato di avanzamento del flusso di lavoro.
9. Verifica che l'applicazione sia stata aggiornata, come segue:
 - a. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
 - b. A sinistra, in basso, scegli Load Balancers.
 - c. Seleziona il load balancer che è stato creato come parte della tua implementazione Kubernetes.
 - d. Copia e incolla il valore del nome DNS nella barra degli indirizzi del browser.

Il «Tutorial completo!» Nel browser viene visualizzata una pagina Web che indica che è stata implementata correttamente una nuova revisione dell'applicazione.

10. (Facoltativo) In AWS, passa alla console Amazon ECR e verifica che la nuova immagine Docker sia stata etichettata con l'ID di commit del passaggio 7 di questa procedura.

Eliminazione

È necessario ripulire l'ambiente in modo da non addebitare inutilmente i costi delle risorse di archiviazione e di elaborazione utilizzate in questo tutorial.

Per eliminare

1. Elimina il tuo cluster:

- Nel terminale Dev Environment, inserisci:

```
eksctl delete cluster --region=us-west-2 --name=codecatalyst-eks-cluster
```

Dove:

- *us-west-2* viene sostituito dalla tua regione.
- *codecatalyst-eks-cluster* viene sostituito dal nome del cluster creato.

Dopo 5-10 minuti, il cluster e le risorse associate vengono eliminati, inclusi, a titolo esemplificativo, CloudFormation stack, gruppi di nodi (in Amazon EC2) e sistemi di bilanciamento del carico.

Important

Se il `eksctl delete cluster` comando non funziona, potrebbe essere necessario aggiornare le credenziali o le credenziali. AWS `kubectl` Se non sei sicuro di quali credenziali aggiornare, aggiorna prima le credenziali. AWS Per aggiornare le credenziali, consulta. AWS [Come posso correggere gli errori «Impossibile individuare le credenziali» e "ExpiredToken"?](#) Per aggiornare le `kubectl` credenziali, consulta. [Come posso correggere gli errori «Impossibile connettersi al server»?](#)

2. Nella AWS console, pulisci come segue:
 1. In Amazon ECR, elimina `codecatalyst-eks-image-repo`.
 2. In IAM Identity Center, elimina:
 - a. `codecatalyst-eks-user`
 - b. `codecatalyst-eks-permission-set`
 3. In IAM, elimina:
 - `codecatalyst-eks-build-role`
 - `codecatalyst-eks-deploy-role`
 - `codecatalyst-eks-build-policy`
 - `codecatalyst-eks-deploy-policy`
3. Nella CodeCatalyst console, pulisci come segue:
 1. Elimina `codecatalyst-eks-workflow`.
 2. Elimina `codecatalyst-eks-environment`.
 3. Elimina `codecatalyst-eks-source-repository`.
 4. Elimina il tuo ambiente di sviluppo.
 5. Elimina `codecatalyst-eks-project`.

In questo tutorial, hai imparato a distribuire un'applicazione su un servizio Amazon EKS utilizzando un CodeCatalyst flusso di lavoro e un'azione cluster Deploy to Kubernetes.

Aggiungere l'azione «Deploy to Kubernetes cluster»

Utilizza le seguenti istruzioni per aggiungere l'azione Deploy to Kubernetes cluster al tuo flusso di lavoro.

Prima di iniziare

Prima di aggiungere l'azione Deploy to Kubernetes cluster al flusso di lavoro, devi aver preparato quanto segue:

Tip

Per configurare rapidamente questi prerequisiti, segui le istruzioni riportate in. [Tutorial: distribuzione di un'applicazione su Amazon EKS](#)

- Un cluster Kubernetes in Amazon EKS. Per informazioni sui cluster, consulta i cluster [Amazon EKS nella Guida per l'utente](#) di Amazon EKS.
- Almeno un Dockerfile che descrive come assemblare l'applicazione in un'immagine Docker. [Per ulteriori informazioni su Dockerfiles, consulta il riferimento a Dockerfile.](#)
- Almeno un file manifest Kubernetes, chiamato file di configurazione o configurazione nella documentazione di Kubernetes. Per ulteriori informazioni, consulta [Gestione](#) delle risorse nella documentazione di Kubernetes.
- Un ruolo IAM che offre all'azione del cluster Deploy to Kubernetes la possibilità di accedere e interagire con il cluster Amazon EKS. Per ulteriori informazioni, consulta l'argomento [Role](#) nella [Azione «Distribuisci su cluster Kubernetes» YAML](#).

Dopo aver creato questo ruolo, devi aggiungerlo a:

- Il tuo file Kubernetes ConfigMap . Per informazioni su come aggiungere un ruolo a un ConfigMap file, consulta la sezione [Abilitazione dell'accesso principale IAM al cluster](#) nella Amazon EKS User Guide.
- CodeCatalyst. Per informazioni su come aggiungere un ruolo IAM a CodeCatalyst, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).
- Uno CodeCatalyst spazio, un progetto e un ambiente. Lo spazio e l'ambiente devono essere entrambi collegati all' AWS account in cui verrà distribuita l'applicazione. Per ulteriori informazioni, consultare [Creare uno spazio](#), [Creare un progetto vuoto in Amazon CodeCatalyst](#) e [Implementazione in e Account AWS VPCs](#).
- Un repository di sorgenti supportato da. CodeCatalyst Il repository memorizza i file sorgente dell'applicazione, i Dockerfile e i manifesti di Kubernetes. Per ulteriori informazioni, consulta [Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst](#).

Visual

Per aggiungere l'azione «Distribuisci al cluster Kubernetes» utilizzando l'editor visivo

1. Apri CodeCatalyst la console [all'](#)indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.

5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione Deploy to Kubernetes cluster ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Deploy to Kubernetes cluster. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
- 10. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, consulta. [Azione «Distribuisci su cluster Kubernetes» YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
- 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
- 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere l'azione «Deploy to Kubernetes cluster» utilizzando l'editor YAML

1. Apri CodeCatalyst la console [all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.

6. Scegli YAML.
 7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
 8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
 9. Cerca l'azione Deploy to Kubernetes cluster ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.
- Or
- Scegli Deploy to Kubernetes cluster. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [Azione «Distribuisci su cluster Kubernetes» YAML](#)
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Variabili «Distribuisci su cluster Kubernetes»

L'azione cluster Deploy to Kubernetes produce e imposta le seguenti variabili in fase di esecuzione. Queste sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

Chiave	Valore
cluster	L'Amazon.com Resource Name (ARN) del cluster Amazon EKS su cui è stato distribuito durante l'esecuzione del flusso di lavoro.

Chiave	Valore
	Ad esempio: <code>arn:aws:eks:us-west-2:111122223333:cluster/codecatalyst-eks-cluster</code>
piattaforma di distribuzione	Il nome della piattaforma di distribuzione. Codificato in. <code>AWS:EKS</code>
metadati	Riservata. Metadati in formato JSON relativi al cluster distribuito durante l'esecuzione del flusso di lavoro.
namespace	Lo spazio dei nomi Kubernetes in cui è stato distribuito il cluster. Ad esempio: <code>default</code>
risorse	Riservata. Metadati in formato JSON relativi alle risorse distribuite durante l'esecuzione del flusso di lavoro.
server	Il nome dell'endpoint del server API che puoi utilizzare per comunicare con il cluster utilizzando strumenti di gestione come <code>kubectl</code> . Per ulteriori informazioni sull'endpoint del servizio API, consulta Amazon EKS Cluster Endpoint Access Control nella Amazon EKS User Guide. Ad esempio: <code>https://random-string.gr7.us-west-2.eks.amazonaws.com</code>

Azione «Distribuisci su cluster Kubernetes» YAML

Di seguito è riportata la definizione YAML dell'azione del cluster Deploy to Kubernetes. Per informazioni su come utilizzare questa azione, consulta [Implementazione su Amazon EKS con un flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
DeployToKubernetesCluster\_nn:
  Identifier: aws/kubernetes-deploy@v1
  DependsOn:
    - build-action
  Compute:
    - Type: EC2 | Lambda
    - Fleet: fleet-name
  Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
    Role: DeployToEKS
  Inputs:
    # Specify a source or an artifact, but not both.
  Sources:
    - source-name-1
```

Artifacts:

- *manifest-artifact*

Configuration:

Namespace: *namespace*

Region: *us-east-1*

Cluster: *eks-cluster*

Manifests: *manifest-path*

DeployToKubernetesCluster

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: `DeployToKubernetesCluster_nn`.

Interfaccia utente corrispondente: scheda di configurazione/nome visualizzato dell'azione

Identifier

(*DeployToKubernetesCluster*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: `aws/kubernetes-deploy@v1`.

Interfaccia utente corrispondente: Workflow diagram/ `DeployToKubernetesCluster _nn/ aws/kubernetes-deploy @v1` label

DependsOn

(*DeployToKubernetesCluster*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente affinché questa azione possa essere eseguita.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Ingressi/Dipende da - opzionale

Compute

(*DeployToKubernetesCluster*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*DeployToKubernetesCluster*/Compute/Type)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: configurazione tab/Advanced : opzionale/tipo di calcolo

Fleet

(*DeployToKubernetesCluster*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse

necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand: `Linux.x86-64.Large` `Linux.x86-64.XLarge` Per ulteriori informazioni sulle flotte on-demand, vedere [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omissa, l'impostazione predefinita è `Linux.x86-64.Large`

Interfaccia utente corrispondente: Configurazione tab/Advanced : parco opzionale/di calcolo

Timeout

(*DeployToKubernetesCluster*/Timeout)

(Facoltativo)

Specificate il periodo di tempo, in minuti (editor YAML) o in ore e minuti (editor visivo), che l'azione può essere eseguita prima che termini l'azione. CodeCatalyst Il minimo è 5 minuti e il massimo è descritto in [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Environment

(*DeployToKubernetesCluster*/Environment)

(Obbligatorio)

Specificare l'ambiente CodeCatalyst da utilizzare con l'azione. L'azione si connette all'Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs](#) e [Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*DeployToKubernetesCluster*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*DeployToKubernetesCluster*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo

- (Versioni precedenti) Scheda di configurazione/' / connessione all'account Environment/account/role AWS

Name

(*DeployToKubernetesCluster*/Environment/Connections/Name)

(Facoltativo)

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' / connessione all'account Environment/account/role AWS

Role

(*DeployToKubernetesCluster*/Environment/Connections/Role)

(Richiesto se è incluso [Connections](#))

Specificate il nome del ruolo IAM a cui l'azione cluster Deploy to Kubernetes utilizza per accedere. AWS Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

- La seguente politica di autorizzazioni:

 Warning

Limita le autorizzazioni a quelle mostrate nella seguente politica. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

La prima volta che il ruolo viene utilizzato, utilizza il seguente carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- La seguente politica di fiducia personalizzata:

Assicurati che questo ruolo sia aggiunto a:

- Connessione al tuo account. Per ulteriori informazioni sull'aggiunta di un ruolo IAM alla connessione di un account, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).
- Il tuo Kubernetes ConfigMap. Per ulteriori informazioni sull'aggiunta di un ruolo IAM a ConfigMap, consulta [Gestire gli utenti e i ruoli IAM nella documentazione](#). eksctl

Tip

Vedi anche [Tutorial: distribuzione di un'applicazione su Amazon EKS](#) per istruzioni su come aggiungere un ruolo IAM alla connessione di un account e ConfigMap.

Note

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#).

Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / Ruolo Environment/account/role

Inputs

(*DeployToKubernetesCluster*/Inputs)

(Obbligatorio se incluso [Connections](#))

La Inputs sezione definisce i dati DeployToKubernetesCluster necessari durante l'esecuzione di un flusso di lavoro.

Note

È consentito un solo input (fonte o artefatto) per azione Deploy to Amazon EKS.

Interfaccia utente corrispondente: scheda Input

Sources

(*DeployToKubernetesCluster*/Inputs/Sources)

(Richiesto se il file manifest è archiviato in un repository di origine)

Se il file o i file manifest di Kubernetes sono archiviati in un repository di origine, specifica l'etichetta di tale repository di origine. Attualmente, l'unica etichetta supportata è `WorkflowSource`

Se i file manifest non sono contenuti in un repository di origine, devono risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*DeployToKubernetesCluster*/Inputs/Artifacts)

(Richiesto se il file manifest è memorizzato in un elemento [di output di un'azione precedente](#))

Se il file o i file manifest di Kubernetes sono contenuti in un artefatto generato da un'azione precedente, specifica quell'artefatto qui. Se i tuoi file manifest non sono contenuti in un artefatto, devono risiedere nel tuo repository di origine.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda di configurazione/artefatti - opzionale

Configuration

(*DeployToKubernetesCluster*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

Namespace

(*DeployToKubernetesCluster*/Configuration/Namespace)

(Facoltativo)

Specificate lo spazio dei nomi Kubernetes in cui verrà distribuita l'applicazione Kubernetes. Utilizzalo default se non utilizzi namespace con il tuo cluster. Per ulteriori informazioni sugli spazi dei nomi, consulta [Suddivisione del cluster utilizzando gli spazi dei nomi Kubernetes nella documentazione di Kubernetes](#).

Se ometti lo spazio dei default nomi, viene utilizzato il valore di.

Interfaccia utente corrispondente: scheda di configurazione/ Namespace

Region

(*DeployToKubernetesCluster*/Configuration/Region)

(Obbligatorio)

Specificare la AWS regione in cui risiedono il cluster e il servizio Amazon EKS. Per un elenco dei codici regionali, consulta [Endpoint regionali](#) nel. Riferimenti generali di AWS

Interfaccia utente corrispondente: scheda/regione di configurazione

Cluster

(*DeployToKubernetesCluster*/Configuration/Cluster)

(Obbligatorio)

Specificare il nome di un cluster Amazon EKS esistente. L'azione Deploy to Kubernetes cluster distribuirà l'applicazione containerizzata in questo cluster. Per ulteriori informazioni sui cluster Amazon EKS, consulta [Clusters](#) nella Amazon EKS User Guide.

Interfaccia utente corrispondente: scheda di configurazione/ Cluster

Manifests

(*DeployToKubernetesCluster*/Configuration/Manifests)

(Obbligatorio)

Specificate il percorso dei file manifest di Kubernetes in formato YAML, denominati file di configurazione, file di configurazione o semplicemente configurazioni nella documentazione di Kubernetes.

Se utilizzi più file manifest, inseriscili in un'unica cartella e fai riferimento a quella cartella. I file manifest vengono elaborati in ordine alfanumerico da Kubernetes, quindi assicurati di anteporre ai nomi dei file numeri o lettere crescenti per controllare l'ordine di elaborazione. Esempio:

```
00-namespace.yaml
```

```
01-deployment.yaml
```

Se i file manifest risiedono nel repository di origine, il percorso è relativo alla cartella principale del repository di origine. Se i file si trovano in un elemento di un'azione precedente del workflow, il percorso è relativo alla cartella principale dell'artefatto.

Esempi:

```
Manifests/
```

```
deployment.yaml
```

```
my-deployment.yaml
```

Non utilizzate caratteri jolly (). *

Note

I [grafici Helm](#) e i file [di personalizzazione](#) non sono supportati.

Per ulteriori informazioni sui file manifest, consulta [Organizzazione delle configurazioni delle risorse](#) nella documentazione di Kubernetes.

Interfaccia utente corrispondente: scheda di configurazione/ Manifesti

Implementazione di uno stack CloudFormation

Questa sezione descrive come distribuire uno AWS CloudFormation stack utilizzando un flusso di lavoro. CodeCatalyst A tale scopo, è necessario aggiungere l'azione Deploy CloudFormation stack al flusso di lavoro. L'azione distribuisce una CloudFormation pila di risorse in AWS base a un modello fornito dall'utente. Il modello può essere un:

- CloudFormation modello: per ulteriori informazioni, consulta [Lavorare con CloudFormation i modelli](#).

- AWS SAM modello — Per ulteriori informazioni, vedere la [specifica AWS Serverless Application Model \(AWS SAM\)](#).

Note

Per utilizzare un AWS SAM modello, dovete prima impacchettare l' AWS SAM applicazione utilizzando l'[sam package](#) operazione. Per un tutorial che mostra come eseguire automaticamente questa confezione come parte di un CodeCatalyst flusso di lavoro Amazon, consulta [Tutorial: Implementazione di un'applicazione serverless](#).

Se lo stack esiste già, l'azione esegue l' CloudFormation [CreateChangeSet](#) operazione e quindi l'[ExecuteChangeSet](#) operazione. L'azione attende quindi che le modifiche vengano distribuite e si contrassegna come riuscita o non riuscita, a seconda dei risultati.

Utilizza l'azione Deploy CloudFormation stack se disponi già di un AWS SAM modello CloudFormation o che contiene risorse che desideri distribuire, oppure prevedi di generarne uno automaticamente come parte di un'azione di [creazione](#) del flusso di lavoro utilizzando strumenti come e. AWS SAM [AWS Cloud Development Kit \(AWS CDK\)](#)

Non ci sono restrizioni sul modello che puoi utilizzare, qualunque cosa tu possa creare CloudFormation o utilizzare con l'azione AWS SAM Deploy stack. CloudFormation

Tip

Per un tutorial che mostra come distribuire un'applicazione serverless utilizzando l'azione Deploy stack, consulta. CloudFormation [Tutorial: Implementazione di un'applicazione serverless](#)

Argomenti

- [Immagine di runtime utilizzata dall'azione 'Deploy stack' CloudFormation](#)
- [Tutorial: Implementazione di un'applicazione serverless](#)
- [Aggiungere l'azione «Deploy stack» CloudFormation](#)
- [Configurazione dei rollback](#)
- [Variabili 'Deploy CloudFormation stack'](#)
- [Azione 'Implementa CloudFormation stack' YAML](#)

Immagine di runtime utilizzata dall'azione 'Deploy stack' CloudFormation

[L'azione Deploy CloudFormation stack viene eseguita su un'immagine di novembre 2022.](#) Per ulteriori informazioni, consulta [Immagini attive](#).

Tutorial: Implementazione di un'applicazione serverless

In questo tutorial, imparerai come creare, testare e distribuire un'applicazione serverless come stack utilizzando un flusso di lavoro. CloudFormation

L'applicazione in questo tutorial è una semplice applicazione web che emette un messaggio «Hello World». È costituito da una AWS Lambda funzione e da un Amazon API Gateway e lo si crea utilizzando [AWS Serverless Application Model \(AWS SAM\)](#), che è un'estensione di [CloudFormation](#).

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un archivio di sorgenti](#)
- [Fase 2: Creare ruoli AWS](#)
- [Fase 3: Aggiungere ruoli a AWS CodeCatalyst](#)
- [Fase 4: creare un bucket Amazon S3](#)
- [Passaggio 5: Aggiungere i file sorgente](#)
- [Passaggio 6: creare ed eseguire un flusso di lavoro](#)
- [Fase 7: Apportare una modifica](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare:

- È necessario uno CodeCatalyst spazio con un AWS account connesso. Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Nel tuo spazio, hai bisogno di un progetto vuoto chiamato:

```
codecatalyst-cfn-project
```

Usa l'opzione Inizia da zero per creare questo progetto.

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

- Nel tuo progetto, hai bisogno di un CodeCatalyst ambiente chiamato:

```
codecatalyst-cfn-environment
```

Configura questo ambiente come segue:

- Scegli qualsiasi tipo, ad esempio Non di produzione.
- Connect il tuo AWS account.
- Per il ruolo IAM predefinito, scegli un ruolo qualsiasi. Specificherai un ruolo diverso in seguito.

Per ulteriori informazioni, consulta [Implementazione in e Account AWS VPCs](#).

Fase 1: Creare un archivio di sorgenti

In questo passaggio, creerai un repository di origine in CodeCatalyst. Questo repository viene utilizzato per archiviare i file sorgente del tutorial, come il file della funzione Lambda.

Per ulteriori informazioni sui repository dei sorgenti, consulta [Creazione di un repository di sorgenti](#)

Per creare un archivio di sorgenti

1. Nel CodeCatalyst riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
2. Scegli Aggiungi repository, quindi scegli Crea repository.
3. Nel nome del repository, inserisci:

```
codecatalyst-cfn-source-repository
```

4. Scegli Create (Crea).

Ora hai creato un repository chiamato `codecatalyst-cfn-source-repository`

Fase 2: Creare ruoli AWS

In questo passaggio, crei i seguenti ruoli AWS IAM:

- Ruolo Deploy: concede all'azione CodeCatalyst Deploy CloudFormation stack l'autorizzazione ad accedere al tuo AWS account e al CloudFormation servizio su cui distribuirai la tua applicazione serverless. L'azione Deploy CloudFormation stack fa parte del tuo flusso di lavoro.

- Ruolo di compilazione: concede all'azione di CodeCatalyst compilazione l'autorizzazione ad accedere al tuo AWS account e scrivere su Amazon S3 dove verrà archiviato il pacchetto di applicazioni serverless. L'azione di compilazione fa parte del tuo flusso di lavoro.
- Ruolo Stack: concede CloudFormation l'autorizzazione a leggere e modificare le risorse specificate nel AWS SAM modello che fornirai in seguito. Concede inoltre l'autorizzazione a CloudWatch

Per ulteriori informazioni sui ruoli IAM, consulta i [ruoli IAM](#) nella Guida per l'AWS Identity and Access Management utente.

Note

Per risparmiare tempo, puoi creare un singolo ruolo, chiamato `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo, anziché i tre ruoli elencati in precedenza. Per ulteriori informazioni, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo ha autorizzazioni molto ampie che possono rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante. Questo tutorial presuppone che stiate creando i tre ruoli elencati in precedenza.

Note

È richiesto anche un [ruolo di esecuzione Lambda](#), ma non è necessario crearlo ora perché il `sam-template.yml` file lo crea automaticamente quando si esegue il flusso di lavoro nel passaggio 5.

Per creare un ruolo di distribuzione

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - c. Nel riquadro di navigazione, scegli Policy.

- d. Scegli Create Policy (Crea policy).
- e. Scegliere la scheda JSON.
- f. Eliminare il codice esistente.
- g. Incolla il codice seguente:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-deploy-policy
```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di distribuzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione, cerca `codecatalyst-deploy-policy` e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per il nome del ruolo, inserisci:

```
codecatalyst-deploy-role
```

- i. Per la descrizione del ruolo, inserisci:

CodeCatalyst deploy role

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di distribuzione con una politica di fiducia e una politica di autorizzazioni.

3. Ottenere l'ARN del ruolo di distribuzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (`()codecatalyst-deploy-role`).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di distribuzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Per creare un ruolo di costruzione

1. Crea una politica per il ruolo, come segue:

- a. Accedi a AWS.
- b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
- c. Nel riquadro di navigazione, scegli Policy.
- d. Scegli Create Policy (Crea policy).
- e. Scegliere la scheda JSON.
- f. Eliminare il codice esistente.
- g. Incolla il codice seguente:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next: Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-build-policy
```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di costruzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione, cerca `codecatalyst-build-policy` e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per il nome del ruolo, inserisci:

```
codecatalyst-build-role
```

- i. Per la descrizione del ruolo, inserisci:

```
CodeCatalyst build role
```

j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di fiducia e una politica di autorizzazioni.

3. Ottieni l'ARN del ruolo di costruzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (`codecatalyst-build-role`).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di costruzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Per creare un ruolo stack

1. Accedi AWS utilizzando l'account in cui desideri distribuire lo stack.
2. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
3. Crea il ruolo dello stack come segue:
 - a. Nel pannello di navigazione, seleziona Roles (Ruoli).
 - b. Selezionare Create role (Crea ruolo).
 - c. Scegli un servizio in Servizio AWS .
 - d. Nella sezione Caso d'uso, scegli CloudFormation dall'elenco a discesa.
 - e. Seleziona il pulsante CloudFormationradio.
 - f. In basso, scegli Avanti.
 - g. Utilizzando la casella di ricerca, trova le seguenti politiche di autorizzazione, quindi seleziona le rispettive caselle di controllo.

 Note

Se cerchi una politica e questa non viene visualizzata, assicurati di scegliere Cancella filtri e riprova.

- CloudWatchFullAccess
- AWS CloudFormationFullAccess
- IAMFullAccess
- AWS Lambda_FullAccess
- APIGatewayAmministratore Amazon
- Amazon S3 FullAccess
- AmazonEC2ContainerRegistryFullAccess

La prima policy consente l'accesso per CloudWatch abilitare i rollback dello stack quando si verifica un allarme.

Le politiche rimanenti consentono di accedere AWS SAM ai servizi e alle risorse dello stack che verrà distribuito in questo tutorial. Per ulteriori informazioni, consulta [Autorizzazioni nella Guida](#) per gli AWS Serverless Application Model sviluppatori.

- Scegli Next (Successivo).
- Per il nome del ruolo, inserisci:

codecatalyst-stack-role

- Scegli Crea ruolo.
- Ottieni l'ARN del ruolo stack, come segue:
 - Nel riquadro di navigazione, seleziona Ruoli.
 - Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato ()codecatalyst-stack-role.
 - Scegli il ruolo dall'elenco.
 - Nella sezione Riepilogo, copia il valore ARN. perché sarà necessaria in seguito.

Ora hai creato il ruolo stack con le autorizzazioni appropriate e ne hai ottenuto l'ARN.

Fase 3: Aggiungere ruoli a AWS CodeCatalyst

In questo passaggio, aggiungi build role (`codecatalyst-build-role`) e deploy role (`codecatalyst-deploy-role`) alla connessione dell' CodeCatalyst account nel tuo spazio.

Note

Non è necessario aggiungere lo stack role (`codecatalyst-stack-role`) alla connessione. Questo perché il ruolo stack viene utilizzato da CloudFormation(non CodeCatalyst), dopo che è già stata stabilita una connessione tra CodeCatalyst e AWS utilizzando il ruolo deploy. Poiché il ruolo stack non viene utilizzato per accedere CodeCatalyst a AWS, non è necessario che sia associato a una connessione all'account.

Per aggiungere ruoli, build e deploy alla connessione del tuo account

1. Nel CodeCatalyst, accedi al tuo spazio.
2. Seleziona AWS accounts (Account). Viene visualizzato un elenco di connessioni all'account.
3. Scegli la connessione all'account che rappresenta l' AWS account in cui hai creato i ruoli di compilazione e distribuzione.
4. Scegli Gestisci ruoli dalla console AWS di gestione.

Viene visualizzata la pagina Aggiungi ruolo IAM CodeCatalyst allo spazio Amazon. Potrebbe essere necessario effettuare il login per accedere alla pagina.

5. Seleziona Aggiungi un ruolo esistente che hai creato in IAM.

Viene visualizzato un elenco a discesa. L'elenco mostra tutti i ruoli IAM con una policy di fiducia che include i principali `codecatalyst-runner.amazonaws.com` di `codecatalyst.amazonaws.com` servizio.

6. Nell'elenco a discesa **`codecatalyst-build-role`**, scegli e scegli Aggiungi ruolo.
7. Scegli Aggiungi ruolo IAM, scegli Aggiungi un ruolo esistente che hai creato in IAM e, nell'elenco a discesa, scegli. `codecatalyst-deploy-role` Scegliere Add role (Aggiungi ruolo).

Ora hai aggiunto i ruoli di creazione e distribuzione al tuo spazio.

8. Copia il valore del nome CodeCatalyst visualizzato di Amazon. Questo valore ti servirà in seguito, durante la creazione del tuo flusso di lavoro.

Fase 4: creare un bucket Amazon S3

In questa fase, crei un bucket Amazon S3 in cui archiviare il file.zip del pacchetto di distribuzione dell'applicazione serverless.

Come creare un bucket Amazon S3.

1. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
2. Nel riquadro principale, scegli Crea bucket.
3. Per il nome del bucket, inserisci:

codecatalyst-cfn-s3-bucket

4. In AWS Region (Regione), scegliere una Regione. Questo tutorial presuppone che tu abbia scelto US West (Oregon) us-west-2. Per informazioni sulle regioni supportate da Amazon S3, consulta gli [endpoint e le quote di Amazon Simple Storage Service](#) nel. Riferimenti generali di AWS
5. Nella parte inferiore della pagina, scegli Crea bucket.

Ora hai creato un bucket chiamato **codecatalyst-cfn-s3-bucket** nella regione us-west-2 degli Stati Uniti occidentali (Oregon).

Passaggio 5: Aggiungere i file sorgente

In questo passaggio, aggiungi diversi file sorgente dell'applicazione al tuo repository CodeCatalyst di origine. La `hello-world` cartella contiene i file dell'applicazione che distribuirete. La `tests` cartella contiene i test unitari. La struttura delle cartelle è la seguente:

```
.
|- hello-world
|  |- tests
|     |- unit
|        |- test-handler.js
|  |- app.js
|- .npmignore
|- package.json
|- sam-template.yml
|- setup-sam.sh
```


file.npmignore

Il `.npmignore` file indica quali file e cartelle npm deve escludere dal pacchetto dell'applicazione. In questo tutorial, npm esclude la `tests` cartella perché non fa parte dell'applicazione.

Per aggiungere il file.npmignore

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/).
2. Scegli il tuo progetto, `codecatalyst-cfn-project`
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Dall'elenco dei repository di origine, scegli il tuo repository, `codecatalyst-cfn-source-repository`
5. In File, scegli Crea file.
6. Per Nome file, inserisci:

```
.npmignore
```

7. Nella casella di testo, inserisci il seguente codice:

```
tests/*
```

8. Scegliete Commit, quindi scegliete nuovamente Commit.

Ora hai creato un file chiamato `.npmignore` nella radice del tuo repository.

file package.json

Il `package.json` file contiene importanti metadati sul progetto Node, come il nome del progetto, il numero di versione, la descrizione, le dipendenze e altri dettagli che descrivono come interagire ed eseguire l'applicazione.

`package.json` In questo tutorial sono inclusi un elenco di dipendenze e uno script. `test` Lo script di `test` esegue le seguenti operazioni:

- Utilizzando [mocha](#), lo script di test esegue i test unitari specificati in `hello-world/tests/unit/` e scrive i risultati in un `junit.xml` file utilizzando il reporter [xunit](#).
- [Utilizzando Istanbul \(nyc\), lo script di test genera un rapporto sulla copertura del codice \(clover.xml\) utilizzando il clover reporter.](#) Per ulteriori informazioni, consulta [Usare reporter alternativi](#) nella documentazione di Istanbul.

Per aggiungere il file `package.json`

1. Nel tuo repository, in File, scegli Crea file.
2. Per Nome file, inserisci:

`package.json`

3. Nella casella di testo, inserisci il seguente codice:

```
{
  "name": "hello_world",
  "version": "1.0.0",
  "description": "hello world sample for NodeJS",
  "main": "app.js",
  "repository": "https://github.com/awslabs/aws-sam-cli/tree/develop/samcli/local/
init/templates/cookiecutter-aws-sam-hello-nodejs",
  "author": "SAM CLI",
  "license": "MIT",
  "dependencies": {
    "axios": "^0.21.1",
    "nyc": "^15.1.0"
  },
  "scripts": {
    "test": "nyc --reporter=clover mocha hello-world/tests/unit/ --reporter xunit
--reporter-option output=junit.xml"
  },
  "devDependencies": {
    "aws-sdk": "^2.815.0",
    "chai": "^4.2.0",
    "mocha": "^8.2.1"
  }
}
```

4. Scegliete Commit, quindi scegliete nuovamente Commit.

Ora hai aggiunto un file chiamato `package.json` alla radice del repository.

file `sam-template.yml`

Il `sam-template.yml` file contiene le istruzioni per implementare la funzione Lambda e API Gateway e configurarli insieme. Segue la specifica del [AWS Serverless Application Model modello, che estende la specifica](#) del CloudFormation modello.

In questo tutorial utilizzi un AWS SAM modello invece di un CloudFormation modello normale perché AWS SAM offre un utile tipo di risorsa [AWS::Serverless::Function](#). Questo tipo esegue molte behind-the-scenes configurazioni che normalmente è necessario scrivere per utilizzare la sintassi di base. CloudFormation Ad esempio, `AWS::Serverless::Function` crea una funzione Lambda, un ruolo di esecuzione Lambda e mappature delle sorgenti degli eventi che avviano la funzione. Devi codificare tutto questo se vuoi scriverlo usando basic. CloudFormation

Sebbene questo tutorial utilizzi un modello già scritto, puoi generarne uno come parte del tuo flusso di lavoro utilizzando un'azione di compilazione. Per ulteriori informazioni, consulta [Implementazione di uno stack CloudFormation](#).

Per aggiungere il file `sam-template.yml`

1. Nel tuo repository, in File, scegli Crea file.
2. Per Nome file, inserisci:

```
sam-template.yml
```

3. Nella casella di testo, inserisci il seguente codice:

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: >
  serverless-api

  Sample SAM Template for serverless-api

# More info about Globals: https://github.com/awslabs/serverless-application-model/
blob/master/docs/globals.rst
Globals:
  Function:
    Timeout: 3

Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function # For details on this resource type,
    see https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#awsserverlessfunction
    Properties:
      CodeUri: hello-world/
      Handler: app.lambdaHandler
      Runtime: nodejs12.x
```

```

Events:
  HelloWorld:
    Type: Api # For details on this event source type, see
    https://github.com/aws-labs/serverless-application-model/blob/master/
versions/2016-10-31.md#api
    Properties:
      Path: /hello
      Method: get

Outputs:
  # ServerlessRestApi is an implicit API created out of the events key under
  Serverless::Function
  # Find out about other implicit resources you can reference within AWS SAM at
  # https://github.com/aws-labs/serverless-application-model/blob/master/docs/
internals/generated_resources.rst#api
  HelloWorldApi:
    Description: "API Gateway endpoint URL for the Hello World function"
    Value: !Sub "https://${ServerlessRestApi}.execute-api.
${AWS::Region}.amazonaws.com/Prod/hello/"
  HelloWorldFunction:
    Description: "Hello World Lambda function ARN"
    Value: !GetAtt HelloWorldFunction.Arn
  HelloWorldFunctionIamRole:
    Description: "Implicit Lambda execution role created for the Hello World
function"
    Value: !GetAtt HelloWorldFunctionRole.Arn

```

4. Scegliete Commit, quindi scegliete nuovamente Commit.

Ora hai aggiunto un file chiamato `sam-template.yml` nella cartella principale del tuo repository.

file `setup-sam.sh`

Il `setup-sam.sh` file contiene le istruzioni per scaricare e installare l'utilità AWS SAM CLI. Il flusso di lavoro utilizza questa utilità per impacchettare il `hello-world` codice sorgente.

Per aggiungere il file `setup-sam.sh`

1. Nel tuo repository, in File, scegli Crea file.
2. Per Nome file, inserisci:

```
setup-sam.sh
```

3. Nella casella di testo, inserisci il seguente codice:

```
#!/usr/bin/env bash
echo "Setting up sam"

yum install unzip -y

curl -LO https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-linux-x86_64.zip
unzip -qq aws-sam-cli-linux-x86_64.zip -d sam-installation-directory

./sam-installation-directory/install; export AWS_DEFAULT_REGION=us-west-2
```

Nel codice precedente, sostituiscilo *us-west-2* con la tua AWS regione.

4. Scegli Commit, quindi scegli nuovamente Commit.

Ora hai aggiunto un file chiamato `setup-sam.sh` alla radice del repository.

file `app.js`

`app.js` Contiene il codice della funzione Lambda. In questo tutorial, il codice restituisce il testo `hello world`.

Per aggiungere il file `app.js`

1. Nel tuo repository, in File, scegli Crea file.
2. Per Nome file, inserisci:

```
hello-world/app.js
```

3. Nella casella di testo, inserisci il seguente codice:

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;

/**
 *
```

```

* Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
* @param {Object} event - API Gateway Lambda Proxy Input Format
*
* Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
* @param {Object} context
*
* Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
* @returns {Object} object - API Gateway Lambda Proxy Output Format
*
*/
exports.lambdaHandler = async (event, context) => {
  try {
    // const ret = await axios(url);
    response = {
      'statusCode': 200,
      'body': JSON.stringify({
        message: 'hello world',
        // location: ret.data.trim()
      })
    }
  } catch (err) {
    console.log(err);
    return err;
  }

  return response
};

```

4. Scegliete Commit, quindi scegliete nuovamente Commit.

Ora hai creato una cartella chiamata `hello-world` e un file chiamato `app.js`.

file `test-handler.js`

Il `test-handler.js` file contiene test unitari per la funzione Lambda.

Per aggiungere il file `test-handler.js`

1. Nel tuo repository, in File, scegli Crea file.
2. Per Nome file, inserisci:

```
hello-world/tests/unit/test-handler.js
```

3. Nella casella di testo, inserisci il seguente codice:

```
'use strict';

const app = require('../..../app.js');
const chai = require('chai');
const expect = chai.expect;
var event, context;

describe('Tests index', function () {
  it('verifies successful response', async () => {
    const result = await app.lambdaHandler(event, context)

    expect(result).to.be.an('object');
    expect(result.statusCode).to.equal(200);
    expect(result.body).to.be.an('string');

    let response = JSON.parse(result.body);

    expect(response).to.be.an('object');
    expect(response.message).to.be.equal("hello world");
    // expect(response.location).to.be.an("string");
  });
});
```

4. Scegliete Commit, quindi scegliete nuovamente Commit.

Ora hai aggiunto un file chiamato `test-handler.js` sotto la `hello-world/tests/unit` cartella.

Ora hai aggiunto tutti i tuoi file sorgente.

Prenditi un momento per ricontrollare il tuo lavoro e assicurati di aver inserito tutti i file nelle cartelle corrette. La struttura delle cartelle è la seguente:

```
.
|- hello-world
|   |- tests
|       |- unit
```

```
|      |— test-handler.js
|  |— app.js
|— .npmignore
|— README.md
|— package.json
|— sam-template.yml
|— setup-sam.sh
```

Passaggio 6: creare ed eseguire un flusso di lavoro

In questo passaggio, crei un flusso di lavoro che impacchetta il codice sorgente Lambda e lo distribuisce. Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione di test (Test): al momento del trigger, questa azione installa il [gestore di pacchetti Node \(npm\)](#), quindi esegue il comando `npm run test`. Questo comando dice a npm di eseguire lo test script definito nel file `package.json`. Lo test script, a sua volta, esegue gli unit test e genera due report: un test report (`junit.xml`) e un rapporto sulla copertura del codice (`clover.xml`). Per ulteriori informazioni, consulta [file package.json](#).

Successivamente, l'azione di test trasforma i report XML in CodeCatalyst report e li visualizza nella CodeCatalyst console, nella scheda Report dell'azione di test.

Per ulteriori informazioni sull'azione di test, vedere [Test con flussi di lavoro](#).

- Un'azione di compilazione (BuildBackend): al termine dell'azione di test, l'azione di compilazione scarica e installa la AWS SAM CLI, impacchetta `hello-world` il codice sorgente e copia il pacchetto nel bucket Amazon S3, dove il servizio Lambda prevede che si trovi. L'azione genera anche un nuovo file AWS SAM modello chiamato `sam-template-packaged.yml` e lo inserisce in un elemento di output chiamato `buildArtifact`.

Per ulteriori informazioni sull'azione di compilazione, consulta [Creazione con flussi di lavoro](#).

- A deploy action (DeployCloudFormationStack): al termine dell'azione di compilazione, l'azione deploy cerca l'artefatto di output generato dall'azione di compilazione (`buildArtifact`), trova il AWS SAM modello al suo interno e quindi esegue il modello. Il AWS SAM modello crea uno stack che distribuisce l'applicazione serverless.

Come creare un flusso di lavoro

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli Crea flusso di lavoro.
3. Per Source repository, scegli `codecatalyst-cfn-source-repository`.
4. Per Branch, scegli `main`.
5. Scegli Create (Crea).
6. Eliminate il codice di esempio YAML.
7. Aggiungi il seguente codice YAML:

Note

Nel codice YAML che segue, puoi omettere le sezioni se lo `Connections:` desideri. Se ometti queste sezioni, devi assicurarti che il ruolo specificato nel campo Default IAM role nel tuo ambiente includa le autorizzazioni e le policy di fiducia di entrambi i ruoli descritte in. [Fase 2: Creare ruoli AWS](#) Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta. [Creazione di un ambiente](#)

```
Name: codecatalyst-cfn-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  Test:
    Identifier: aws/managed-test@v1
    Inputs:
      Sources:
        - WorkflowSource
    Outputs:
      Reports:
        CoverageReport:
          Format: CLOVERXML
          IncludePaths:
            - "coverage/*"
        TestReport:
```

```

        Format: JUNITXML
        IncludePaths:
          - junit.xml
      Configuration:
        Steps:
          - Run: npm install
          - Run: npm run test
    BuildBackend:
      Identifier: aws/build@v1
      DependsOn:
        - Test
      Environment:
        Name: codecatalyst-cfn-environment
      Connections:
        - Name: codecatalyst-account-connection
          Role: codecatalyst-build-role
      Inputs:
        Sources:
          - WorkflowSource
      Configuration:
        Steps:
          - Run: . ./setup-sam.sh
          - Run: sam package --template-file sam-template.yml --s3-
bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml
--region us-west-2
      Outputs:
        Artifacts:
          - Name: buildArtifact
          Files:
            - "**/*"
    DeployCloudFormationStack:
      Identifier: aws/cfn-deploy@v1
      DependsOn:
        - BuildBackend
      Environment:
        Name: codecatalyst-cfn-environment
      Connections:
        - Name: codecatalyst-account-connection
          Role: codecatalyst-deploy-role
      Inputs:
        Artifacts:
          - buildArtifact
        Sources: []
      Configuration:

```

```

name: codecatalyst-cfn-stack
region: us-west-2
role-arn: arn:aws:iam::111122223333:role/StackRole
template: ./sam-template-packaged.yml
capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND

```

Nel codice precedente, sostituisci:

- Entrambe le istanze *codecatalyst-cfn-environment* con il nome dell'ambiente.
- Entrambe le istanze *codecatalyst-account-connection* con il nome visualizzato della connessione all'account. Il nome visualizzato potrebbe essere un numero. Per ulteriori informazioni, consulta [Fase 3: Aggiungere ruoli a AWS CodeCatalyst](#).
- *codecatalyst-build-role* con il nome del ruolo di build in cui hai creato [Fase 2: Creare ruoli AWS](#).
- *codecatalyst-cfn-s3-bucket* con il nome del bucket Amazon S3 in cui hai creato. [Fase 4: creare un bucket Amazon S3](#)
- Entrambe le istanze sono nella regione in cui risiede il bucket Amazon S3 (prima istanza) e in cui verrà distribuito lo stack (seconda istanza). *us-west-2* Queste regioni possono essere diverse. Questo tutorial presuppone che entrambe le regioni siano impostate su. *us-west-2* Per informazioni dettagliate sulle regioni supportate da Amazon S3 e CloudFormation, consulta [Endpoint e quote del servizio](#) nel. Riferimenti generali di AWS
- *codecatalyst-deploy-role* con il nome del ruolo di distribuzione in cui hai creato. [Fase 2: Creare ruoli AWS](#)
- *codecatalyst-cfn-environment* con il nome dell'ambiente in [Prerequisiti](#) cui hai creato.
- *arn:aws:iam::111122223333:role/StackRole* con l'Amazon Resource Name (ARN) del ruolo stack in cui hai creato. [Fase 2: Creare ruoli AWS](#)

Note

Se avete deciso di non creare ruoli di compilazione, distribuzione e impilamento *codecatalyst-build-role* *codecatalyst-deploy-role*, sostituiteli e *arn:aws:iam::111122223333:role/StackRole* inserite il nome o l'ARN del ruolo. CodeCatalystWorkflowDevelopmentRole-*spaceName* Per ulteriori informazioni su questo ruolo, consulta [Fase 2: Creare ruoli AWS](#).

Per informazioni sulle proprietà del codice mostrato in precedenza, consulta la. [Azione 'Implementa CloudFormation stack' YAML](#)

8. (Facoltativo) Scegliete Convalida per assicurarvi che il codice YAML sia valido prima di eseguire il commit.
9. Scegli Applica.
10. Nella finestra di dialogo Conferma flusso di lavoro, inserisci quanto segue:
 - a. Per il nome del file Workflow, mantenete il valore predefinito, `codecatalyst-cfn-workflow`.
 - b. Per il messaggio di conferma, inserisci:

add initial workflow file

- c. Per Repository, scegli `codecatalyst-cfn-source-repository`.
- d. Per il nome del ramo, scegli principale.
- e. Scegli Applica.

Ora hai creato un flusso di lavoro. L'esecuzione di un flusso di lavoro viene avviata automaticamente a causa del trigger definito nella parte superiore del flusso di lavoro. In particolare, quando hai eseguito il commit (e inviato) del `codecatalyst-cfn-workflow.yaml` file nell'archivio di origine, il trigger ha avviato l'esecuzione del flusso di lavoro.

Per visualizzare il flusso di lavoro in corso

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il flusso di lavoro che hai appena creato: `codecatalyst-cfn-workflow`
3. Scegli la scheda Esecuzioni.
4. Nella colonna Run ID, scegli l'ID di esecuzione.
5. Scegli Test per vedere l'avanzamento dei test.
6. Scegli BuildBackenddi vedere lo stato di avanzamento della costruzione.
7. Scegli DeployCloudFormationStackdi vedere lo stato di avanzamento della distribuzione.

Per ulteriori informazioni sulla visualizzazione dei dettagli dell'esecuzione, consulta [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

8. Al termine dell'`DeployCloudFormationStackazione`, procedi come segue:

- Se l'esecuzione del flusso di lavoro è riuscita, passate alla procedura successiva.
- Se l'esecuzione del flusso di lavoro non è riuscita durante il test o `BuildBackend` l'azione, scegli Registri per risolvere il problema.
- Se l'esecuzione del flusso di lavoro non è riuscita nell'`DeployCloudFormationStackazione`, scegli l'azione di distribuzione, quindi scegli la scheda Riepilogo. Scorri fino alla sezione CloudFormation degli eventi per visualizzare il messaggio di errore dettagliato. Se si è verificato un rollback, elimina lo `codecatalyst-cfn-stack` stack tramite la CloudFormation console AWS prima di eseguire nuovamente il flusso di lavoro.

Per verificare la distribuzione

1. Dopo una distribuzione riuscita, scegli Variabili (7) dalla barra dei menu orizzontale in alto. (Non scegliete Variabili nel riquadro a destra).
2. Successivamente `HelloWorldApi`, incolla l'`https://URL` in un browser.

Viene visualizzato un messaggio JSON hello world dalla funzione Lambda, che indica che il flusso di lavoro ha distribuito e configurato correttamente la funzione Lambda e l'API Gateway.

 Tip

È possibile CodeCatalyst visualizzare questo URL nel diagramma del flusso di lavoro con alcune piccole configurazioni. Per ulteriori informazioni, consulta [Visualizzazione dell'URL dell'app nel diagramma del flusso di lavoro](#).

Per verificare i risultati dei test unitari e la copertura del codice

1. Nel diagramma del flusso di lavoro, scegli Test, quindi scegli Report.
2. Scegliete di `TestReport` visualizzare i risultati del test unitario o scegliete di `CoverageReport` visualizzare i dettagli della copertura del codice dei file da testare, in questo caso, `app.js` e `test-handler.js`.

Per verificare le risorse distribuite

1. Accedi Console di gestione AWS e apri la console API Gateway all'indirizzo <https://console.aws.amazon.com/apigateway/>.
2. Osserva l'codecatalyst-cfn-stackAPI creata dal AWS SAM modello. Il nome dell'API deriva dal Configuration/name valore nel file di definizione del flusso di lavoro (codecatalyst-cfn-workflow.yaml).
3. Apri la AWS Lambda console all'indirizzo <https://console.aws.amazon.com/lambda/>.
4. Nel riquadro di navigazione, seleziona Funzioni.
5. Scegli la tua funzione Lambda, . codecatalyst-cfn-stack-HelloWorldFunction-*string*
6. Puoi vedere come l'API Gateway funge da trigger per la funzione. Questa integrazione è stata configurata automaticamente in base al tipo di AWS SAM `AWS::Serverless::Function` risorsa.

Fase 7: Apportare una modifica

In questo passaggio, apporti una modifica al codice sorgente Lambda e la esegui il commit. Questo commit avvia l'esecuzione di un nuovo flusso di lavoro. Questa esecuzione implementa la nuova funzione Lambda in uno schema blu-verde che utilizza la configurazione di spostamento del traffico predefinita specificata nella console Lambda.

Per apportare una modifica alla sorgente Lambda

1. In CodeCatalyst, accedi al tuo progetto.
2. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Scegli il tuo repository codecatalyst-cfn-source-repository di origine.
4. Cambia il file dell'applicazione:
 - a. Scegliere la cartella hello-world.
 - b. Scegli il app.js file.
 - c. Scegli Modifica.
 - d. Sulla riga 23, cambia hello world in **Tutorial complete!**.
 - e. Scegli Commetti, quindi scegli di nuovo Commetti.

Il commit avvia l'esecuzione di un flusso di lavoro. Questa esecuzione avrà esito negativo perché non hai aggiornato gli unit test per riflettere la modifica del nome.

5. Aggiorna gli unit test:

- a. Scegli `hello-world\tests\unit\test-handler.js`.
- b. Scegli Modifica.
- c. Sulla riga 19, passare `hello world` a **Tutorial complete!**.
- d. Scegli Commetti, quindi scegli di nuovo Commetti.

Il commit provoca l'avvio di un altro workflow. L'esecuzione avrà esito positivo.

6. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
7. Scegli **codecatalyst-cfn-workflow**, quindi scegli Esecuzioni.
8. Scegli l'ID di esecuzione dell'ultima esecuzione. Dovrebbe essere ancora in corso.
9. Scegli Test BuildBackend, e DeployCloudFormationStack per vedere l'avanzamento del flusso di lavoro viene eseguito.
10. Al termine del flusso di lavoro, scegli Variabili (7) nella parte superiore.
11. Successivamente HelloWorldApi, incolla l'`https://URL` in un browser.

Nel browser viene visualizzato un **Tutorial complete!** messaggio che indica che la nuova applicazione è stata distribuita correttamente.

Eliminazione

Pulisci i file e i servizi utilizzati in questo tutorial per evitare che ti vengano addebitati dei costi.

Per pulire la CodeCatalyst console

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elimina `codecatalyst-cfn-workflow`.
3. Elimina `codecatalyst-cfn-environment`.
4. Elimina `codecatalyst-cfn-source-repository`.
5. Elimina `codecatalyst-cfn-project`.

Per pulire in Console di gestione AWS

1. Effettuare CloudFormation la pulizia nel modo seguente:
 - a. Apri la CloudFormation console in <https://console.aws.amazon.com/cloudformation>.
 - b. Eliminare il `codecatalyst-cfn-stack`.

L'eliminazione dello stack rimuove tutte le risorse del tutorial dai servizi API Gateway e Lambda.
2. Esegui la pulizia in Amazon S3, come segue:
 - a. Apri la console Amazon S3 all'indirizzo. <https://console.aws.amazon.com/s3/>
 - b. Seleziona `codecatalyst-cfn-s3-bucket`.
 - c. Elimina il contenuto del bucket.
 - d. Elimina il bucket.
3. Esegui la pulizia in IAM, come segue:
 - a. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - b. Eliminare il `codecatalyst-deploy-policy`.
 - c. Eliminare il `codecatalyst-build-policy`.
 - d. Eliminare il `codecatalyst-stack-policy`.
 - e. Eliminare il `codecatalyst-deploy-role`.
 - f. Eliminare il `codecatalyst-build-role`.
 - g. Eliminare il `codecatalyst-stack-role`.

In questo tutorial, hai imparato a distribuire un'applicazione serverless come CloudFormation stack utilizzando un CodeCatalyst flusso di lavoro e un'azione CloudFormation Deploy stack.

Aggiungere l'azione «Deploy stack» CloudFormation

Utilizza le seguenti istruzioni per aggiungere l'azione Deploy stack al tuo flusso di lavoro CloudFormation .

Visual

Per aggiungere l'azione 'Deploy CloudFormation stack' utilizzando l'editor visuale

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/) .
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione Deploy CloudFormation stack ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Deploy stack. CloudFormation Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprirne il riquadro di configurazione.
10. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, vedi. [Azione 'Implementa CloudFormation stack' YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere l'azione 'Deploy CloudFormation stack' utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). [CodeCatalyst](#)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione Deploy CloudFormation stack ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Deploy stack. CloudFormation Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprirne il riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [Azione 'Implementa CloudFormation stack' YAML](#)
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Configurazione dei rollback

Per impostazione predefinita, se l'azione Deploy CloudFormation stack fallisce, lo stack CloudFormation verrà ripristinato all'ultimo stato stabile conosciuto. Puoi modificare il comportamento

in modo che i rollback si verifichino non solo quando l'azione fallisce, ma anche quando si verifica uno specifico CloudWatch allarme Amazon. Per ulteriori informazioni sugli CloudWatch allarmi, consulta [Using Amazon CloudWatch alarms](#) nella Amazon CloudWatch User Guide.

Puoi anche modificare il comportamento predefinito in modo da CloudFormation non ripristinare lo stack quando l'azione fallisce.

Utilizza le seguenti istruzioni per configurare i rollback.

Note

Non è possibile avviare un rollback manualmente.

Visual

Prima di iniziare

1. Assicurati di disporre di un [flusso di lavoro](#) che includa un'azione Deploy CloudFormation stack funzionante. Per ulteriori informazioni, consulta [Implementazione di uno stack CloudFormation](#).
2. Nel ruolo specificato nel campo Stack role - opzionale dell'azione Deploy CloudFormation stack, assicurati di includere l'autorizzazione CloudWatchFullAccess. Per informazioni sulla creazione di questo ruolo con le autorizzazioni appropriate, consulta [Fase 2: Creare ruoli AWS](#).

Per configurare gli allarmi di rollback per l'azione «Deploy stack» CloudFormation

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). [CodeCatalyst](#)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Scegli la tua azione Deploy CloudFormation stack.

8. Nel riquadro dei dettagli, scegli Configurazione.
9. In basso, espandi Avanzate.
10. In Monitora allarme ARNs, scegli Aggiungi allarme.
11. Inserisci le informazioni nei seguenti campi.

- Allarme ARN

Specificare l'Amazon Resource Name (ARN) di un CloudWatch allarme Amazon da utilizzare come trigger di rollback. Ad esempio, `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Puoi avere un massimo di cinque trigger di rollback.

 Note

Se specifichi un ARN di CloudWatch allarme, dovrai anche configurare autorizzazioni aggiuntive per abilitare l'azione di accesso. CloudWatch Per ulteriori informazioni, consulta [Configurazione dei rollback](#).

- Tempo di monitoraggio

Specificare un periodo di tempo, da 0 a 180 minuti, durante il quale CloudFormation monitorare gli allarmi specificati. Il monitoraggio inizia dopo che tutte le risorse dello stack sono state distribuite. Se l'allarme si verifica entro il periodo di monitoraggio specificato, l'implementazione fallisce e CloudFormation ripristina l'intera operazione dello stack.

Impostazione predefinita: 0. CloudFormation monitora gli allarmi solo mentre le risorse dello stack vengono distribuite, non dopo.

YAML

Per configurare i trigger di rollback per l'azione «Deploy stack» CloudFormation

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome di un flusso di lavoro che include l'azione CloudFormation Deploy stack. È possibile filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Aggiungi le `monitor-timeout-in-minutes` proprietà `monitor-alarm-arns` and nel codice YAML per aggiungere i trigger di rollback. Per una spiegazione di ciascuna proprietà, vedere. [Azione 'Implementa CloudFormation stack' YAML](#)
8. Nel ruolo specificato nella `role-arn` proprietà dell'azione Deploy CloudFormation stack, assicurati di includere l'CloudWatchFullAccess autorizzazione. Per informazioni sulla creazione di questo ruolo con le autorizzazioni appropriate, consulta. [Fase 2: Creare ruoli AWS](#)

Visual

Per disattivare i rollback per l'azione 'Deploy stack' CloudFormation

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome di un flusso di lavoro che include l'azione CloudFormation Deploy stack. È possibile filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Scegli la tua azione Deploy CloudFormation stack.
8. Nel riquadro dei dettagli, scegli Configurazione.
9. In basso, espandi Avanzate.
10. Attiva Disabilita il rollback.

YAML

Per disattivare i rollback per l'azione «Deploy stack» CloudFormation

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst

2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome di un flusso di lavoro che include l'azione CloudFormation Deploy stack. È possibile filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Aggiungi la `disable-rollback`: 1 proprietà nel codice YAML per interrompere i rollback. Per una spiegazione di questa proprietà, vedere. [Azione 'Implementa CloudFormation stack' YAML](#)

Variabili 'Deploy CloudFormation stack'

L'azione Deploy CloudFormation stack produce e imposta le seguenti variabili in fase di esecuzione. Queste sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

Chiave	Valore
piattaforma di distribuzione	Il nome della piattaforma di distribuzione. Codificato in. <code>AWS::CloudFormation</code>
region	Il codice regionale su Regione AWS cui è stato distribuito durante l'esecuzione del flusso di lavoro. Ad esempio: <code>us-west-2</code>
stack-id	L'Amazon Resource Name (ARN) dello stack distribuito. Ad esempio: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cfn-sta</code>

Chiave	Valore
	ck/6aad4380-100a-11ec-a10a-03b8a84d40df

Azione 'Implementa CloudFormation stack' YAML

Di seguito è riportata la definizione YAML dell'azione CloudFormation Deploy stack. Per informazioni su come utilizzare questa azione, consulta [Implementazione di uno stack CloudFormation](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.  
# See Proprietà di primo livello for details.
```

```
Name: MyWorkflow  
SchemaVersion: 1.0  
Actions:
```

```
# The action definition starts here.
```

```
DeployCloudFormationStack:
```

```
  Identifier: aws/cfn-deploy@v1
```

```
  DependsOn:
```

```
    - build-action
```

```
  Compute:
```

```
    Type: EC2 | Lambda
```

```
    Fleet: fleet-name
```

```
    Timeout: timeout-minutes
```

```
  Environment:
```

```
    Name: environment-name
```

```
  Connections:
```

```

- Name: account-connection-name
  Role: DeployRole
Inputs:
  Sources:
    - source-name-1
  Artifacts:
    - CloudFormation-artifact
Configuration:
  name: stack-name
  region: us-west-2
  template: template-path
  role-arn: arn:aws:iam::123456789012:role/StackRole
  capabilities: CAPABILITY_IAM,CAPABILITY_NAMED_IAM,CAPABILITY_AUTO_EXPAND
  parameter-overrides: KeyOne=ValueOne,KeyTwo=ValueTwo | path-to-JSON-file
  no-execute-changeset: 1|0
  fail-on-empty-changeset: 1|0
  disable-rollback: 1|0
  termination-protection: 1|0
  timeout-in-minutes: minutes
  notification-arns: arn:aws:sns:us-east-1:123456789012:MyTopic,arn:aws:sns:us-east-1:123456789012:MyOtherTopic
  monitor-alarm-arns: arn:aws:cloudwatch::123456789012:alarm/MyAlarm,arn:aws:cloudwatch::123456789012:alarm/MyOtherAlarm
  monitor-timeout-in-minutes: minutes
  tags: '[{"Key": "MyKey1", "Value": "MyValue1"}, {"Key": "MyKey2", "Value": "MyValue2"}]'

```

DeployCloudFormationStack

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: `DeployCloudFormationStack_nn`.

Interfaccia utente corrispondente: scheda di configurazione/nome visualizzato dell'azione

Identifier

(*DeployCloudFormationStack*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: aws/cfn-deploy@v1.

Interfaccia utente corrispondente: Workflow diagram/ DeployCloudFormationStack _nn/ aws/cfn-deploy @v1 label

DependsOn

(*DeployCloudFormationStack*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente affinché questa azione venga eseguita.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Input/Dipende da - opzionale

Compute

(*DeployCloudFormationStack*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*DeployCloudFormationStack*/Compute/Type)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: configurazione tab/Advanced : opzionale/tipo di calcolo

Fleet

(*DeployCloudFormationStack*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand:,. Linux.x86-64.Large Linux.x86-64.XLarge Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere. [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omesso, l'impostazione predefinita è. Linux.x86-64.Large

Interfaccia utente corrispondente: Configurazione tab/Advanced : parco opzionale/di calcolo

Timeout

(*DeployCloudFormationStack*/Timeout)

(Facoltativo)

Specificate il periodo di tempo, in minuti (editor YAML) o in ore e minuti (editor visivo), che l'azione può essere eseguita prima che termini l'azione. CodeCatalyst Il minimo è 5 minuti e il massimo è descritto in. [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout in minuti - opzionale

Environment

(*DeployCloudFormationStack*/Environment)

(Obbligatorio)

Specificare l'ambiente CodeCatalyst da utilizzare con l'azione. L'azione si connette all'Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs](#) e [Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*DeployCloudFormationStack*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*DeployCloudFormationStack*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' '/ connessione all'account Environment/account/role AWS

Name

(*DeployCloudFormationStack*/Environment/Connections/Name)

(Richiesto se incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' '/ connessione all'account Environment/account/role AWS

Role

(*DeployCloudFormationStack*/Environment/Connections/Role)

(Richiesto se incluso [Connections](#))

Specificate il nome del ruolo IAM utilizzato dall'azione Deploy CloudFormation stack per accedere AWS e al CloudFormation servizio. Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

- La seguente politica di autorizzazione:

 Warning

Limita le autorizzazioni a quelle mostrate nella seguente politica. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

 Note

La prima volta che il ruolo viene utilizzato, utilizza il seguente carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*" 
```

- La seguente politica di fiducia personalizzata:

 Note

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' / Ruolo Environment/account/role

Inputs

(*DeployCloudFormationStack*/Inputs)

(Facoltativo)

La Inputs sezione definisce i dati DeployCloudFormationStack necessari durante l'esecuzione di un flusso di lavoro.

Note

Sono consentiti un massimo di quattro input (una fonte e tre artefatti) per ogni azione dello stack CloudFormation Deploy.

Se è necessario fare riferimento a file che risiedono in input diversi (ad esempio una sorgente e un artefatto), l'input di origine è l'input principale e l'artefatto è l'input secondario. I riferimenti ai file negli input secondari richiedono un prefisso speciale per distinguerli dal primario. Per informazioni dettagliate, vedi [Esempio: riferimento a file in più artefatti](#).

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*DeployCloudFormationStack*/Inputs/Sources)

(Richiesto se il AWS SAM modello CloudFormation o è archiviato in un repository di origine)

Se il AWS SAM modello CloudFormation o è archiviato in un repository di origine, specifica l'etichetta di tale repository di origine. Attualmente, l'unica etichetta supportata è. WorkflowSource

Se il AWS SAM modello CloudFormation o non è contenuto in un repository di origine, deve risiedere in un artefatto generato da un'altra azione o in un bucket Amazon S3.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*DeployCloudFormationStack*/Inputs/Artifacts)

(Richiesto se il AWS SAM modello CloudFormation o è memorizzato in un elemento [di output di un'azione precedente](#))

Se il AWS SAM modello CloudFormation o che desideri distribuire è contenuto in un artefatto generato da un'azione precedente, specifica quell'artefatto qui. Se il CloudFormation modello non è contenuto in un artefatto, deve risiedere nel repository di origine o in un bucket Amazon S3.

Per ulteriori informazioni sugli artefatti, inclusi esempi, consulta. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda di configurazione/artefatti - opzionale

Configuration

(*DeployCloudFormationStack*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

name

(*DeployCloudFormationStack*/Configuration/name)

(Obbligatorio)

Specificate un nome per lo CloudFormation stack creato o aggiornato dall'azione Deploy CloudFormation stack.

Interfaccia utente corrispondente: scheda di configurazione/nome dello stack

region

(*DeployCloudFormationStack*/Configuration/region)

(Obbligatorio)

Specificare Regione AWS in che modo verrà distribuito lo stack. Per un elenco dei codici regionali, consulta Endpoint [regionali](#).

Interfaccia utente corrispondente: scheda di configurazione/regione dello stack

template

(*DeployCloudFormationStack*/Configuration/template)

(Obbligatorio)

Specificate il nome e il percorso del file CloudFormation o AWS SAM del modello. Il modello può essere in formato JSON o YAML e può risiedere in un repository di origine, in un elemento di un'azione precedente o in un bucket Amazon S3. Se il file modello si trova in un repository o in un artefatto di origine, il percorso è relativo all'origine o alla radice dell'artefatto. Se il modello si trova in un bucket Amazon S3, il percorso è il valore dell'URL dell'oggetto del modello.

Esempi:

./MyFolder/MyTemplate.json

MyFolder/MyTemplate.yml

https://MyBucket.s3.us-west-2.amazonaws.com/MyTemplate.yml

Note

Potrebbe essere necessario aggiungere un prefisso al percorso del file del modello per indicare in quale artefatto o fonte trovarlo. Per ulteriori informazioni, consultare [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

Interfaccia utente corrispondente: scheda di configurazione/modello

role-arn

(*DeployCloudFormationStack*/Configuration/role-arn)

(Obbligatorio)

Specificare l'Amazon Resource Name (ARN) del ruolo dello stack. CloudFormation utilizza questo ruolo per accedere e modificare le risorse del tuo stack. Ad esempio: `arn:aws:iam::123456789012:role/StackRole`.

Assicurati che il ruolo dello stack includa:

- Una o più politiche di autorizzazione. Le politiche dipendono dalle risorse presenti nello stack. Ad esempio, se lo stack include una AWS Lambda funzione, devi aggiungere le autorizzazioni che concedono l'accesso a Lambda. Se hai seguito il tutorial descritto in [Tutorial: Implementazione di un'applicazione serverless](#), include una procedura intitolata, [Per creare un ruolo stack](#) che elenca le autorizzazioni necessarie allo stack role se stai distribuendo un tipico stack di applicazioni serverless.

 Warning

Limita le autorizzazioni a quelle richieste dal CloudFormation servizio per accedere alle risorse del tuo stack. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

- La seguente politica di fiducia:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Facoltativamente, associa questo ruolo alla connessione del tuo account. Per ulteriori informazioni sull'associazione di un ruolo IAM a una connessione all'account, consulta [Aggiungere ruoli IAM alle connessioni degli account](#). Se non associ il ruolo dello stack alla connessione dell'account, il ruolo

dello stack non verrà visualizzato nell'elenco a discesa del ruolo Stack nell'editor visivo; tuttavia, il ruolo ARN può ancora essere specificato nel campo utilizzando l'editor YAML. `role-arn`

Note

Puoi usare il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: scheda di configurazione/ruolo Stack - opzionale

capabilities

(*DeployCloudFormationStack*/Configuration/capabilities)

(Obbligatorio)

Specificate un elenco di funzionalità IAM necessarie per consentire la creazione CloudFormation di determinati stack. Nella maggior parte dei casi, puoi lasciare capabilities il valore predefinito di `CAPABILITY_IAM,CAPABILITY_NAMED_IAM,CAPABILITY_AUTO_EXPAND`.

Se vedi `##[error] requires capabilities: [capability-name]` nei log delle azioni CloudFormation dello stack Deploy, consulta [Come posso correggere gli errori relativi alle funzionalità IAM?](#) per informazioni su come risolvere il problema.

Per ulteriori informazioni sulle funzionalità IAM, consulta [Riconoscere le risorse IAM nei CloudFormation modelli nella Guida](#) per l'utente IAM.

Interfaccia utente corrispondente: Configuration Tab/Advanced/ Capabilities

parameter-overrides

(*DeployCloudFormationStack*/Configuration/parameter-overrides)

(Facoltativo)

Specificate i parametri nel vostro AWS SAM modello CloudFormation o che non hanno valori predefiniti o per i quali desiderate specificare valori non predefiniti. Per ulteriori informazioni sui parametri, consultate [Parametri](#) nella Guida per l' AWS CloudFormation utente.

La `parameter-overrides` proprietà accetta:

- Un file JSON contenente i parametri e i valori.
- Un elenco di parametri e valori separati da virgole.

Per specificare un file JSON

1. Assicuratevi che il file JSON utilizzi una delle seguenti sintassi:

```
{
  "Parameters": {
    "Param1": "Value1",
    "Param2": "Value2",
    ...
  }
}
```

Oppure...

```
[
  {
    "ParameterKey": "Param1",
    "ParameterValue": "Value1"
  },
  ...
]
```

(Esistono altre sintassi, ma non sono supportate da CodeCatalyst al momento della scrittura.)

Per ulteriori informazioni sulla specificazione CloudFormation dei parametri in un file JSON, vedete [Sintassi JSON supportata nel Command Reference](#).AWS CLI

2. Specificate il percorso del file JSON utilizzando uno dei seguenti formati:

- Se il tuo file JSON si trova in un elemento di output di un'azione precedente, usa:

`file:///artifacts/current-action-name/output-artifact-name/path-to-json-file`

Vedi l'Esempio 1 per i dettagli.

- Se il tuo file JSON si trova nel tuo repository di origine, usa:

`file:///sources/WorkflowSource/path-to-json-file`

Vedi l'Esempio 2 per i dettagli.

Esempio 1: il file JSON risiede in un artefatto di output

```
##My workflow YAML
...
Actions:
  MyBuildAction:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ParamArtifact
          Files:
            - params.json
    Configuration:
      ...
  MyDeployCFNStackAction:
    Identifier: aws/cfn-deploy@v1
    Configuration:
      parameter-overrides: file:///artifacts/MyDeployCFNStackAction/
ParamArtifact/params.json
```

Esempio 2: il file JSON si trova nell'archivio di origine, in una cartella denominata my/folder

```
##My workflow YAML
...
Actions:
  MyDeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      parameter-overrides: file:///sources/WorkflowSource/my/folder/params.json
```

Per utilizzare un elenco di parametri separati da virgole

- Aggiungere coppie nome-valore dei parametri nella `parameter-overrides` proprietà utilizzando il seguente formato:

`param-1=value-1,param-2=value-2`

Ad esempio, supponendo il seguente modello: CloudFormation

```
##My CloudFormation template

Description: My CloudFormation template

Parameters:
  InstanceType:
    Description: Defines the Amazon EC2 compute for the production server.
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t2.small
      - t3.medium

Resources:
  ...
```

... potresti impostare la `parameter-overrides` proprietà come segue:

```
##My workflow YAML
...
Actions:
  ...
  DeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Configuration:
      parameter-overrides: InstanceType=t3.medium,UseVPC=true
```

Note

È possibile specificare il nome di un parametro senza un valore corrispondente `undefined` da utilizzare come valore. Esempio:

```
parameter-overrides: MyParameter=undefined
```

L'effetto è che durante un aggiornamento dello stack, CloudFormation utilizza il valore del parametro esistente per il nome del parametro specificato.

Interfaccia utente corrispondente:

- Scheda di configurazione/Avanzata/ Sostituzioni dei parametri
- Sostituzioni di configurazione/ Specificare le sostituzioni utilizzando un file tab/Advanced/Parameter
- Sostituzioni di configurazione/ tab/Advanced/Parameter Specificare le sostituzioni utilizzando un set di valori

no-execute-changeset

(*DeployCloudFormationStack*/Configuration/no-execute-changeset)

(Facoltativo)

Specificate se desiderate creare il set CodeCatalyst di CloudFormation modifiche e quindi interromperlo prima di eseguirlo. In questo modo è possibile rivedere il set di modifiche nella CloudFormation console. Se ritieni che il set di modifiche abbia un bell'aspetto, disabilita questa opzione e quindi riesegui il flusso di lavoro in modo da CodeCatalyst poter creare ed eseguire il set di modifiche senza interruzioni. L'impostazione predefinita prevede la creazione e l'esecuzione del set di modifiche senza interruzioni. Per ulteriori informazioni, vedete il parametro CloudFormation [deploy](#) nel AWS CLI Command Reference. Per ulteriori informazioni sulla visualizzazione dei set di modifiche, vedere [Visualizzazione di un set di modifiche](#) nella Guida per l'AWS CloudFormation utente.

Interfaccia utente corrispondente: scheda Configurazione/Avanzata/ Nessun set di modifiche da eseguire

fail-on-empty-changeset

(*DeployCloudFormationStack*/Configuration/fail-on-empty-changeset)

(Facoltativo)

Specificate se desiderate fallire CodeCatalyst l'azione Deploy CloudFormation stack se il set di modifiche è vuoto. CloudFormation (Se un set di modifiche è vuoto, significa che non sono state

apportate modifiche allo stack durante l'ultima distribuzione.) L'impostazione predefinita prevede il proseguimento dell'azione se il set di modifiche è vuoto e la restituzione di un UPDATE_COMPLETE messaggio anche se lo stack non è stato aggiornato.

Per ulteriori informazioni su questa impostazione, vedete il parametro CloudFormation [deploy](#) nel AWS CLI Command Reference. Per ulteriori informazioni sui set di modifiche, vedere [Aggiornamento degli stack utilizzando i set di modifiche nella Guida](#) per l'AWS CloudFormation utente.

Interfaccia utente corrispondente: Configuration tab/Advanced/ Fail su changeset vuoto

disable-rollback

(*DeployCloudFormationStack*/Configuration/disable-rollback)

(Facoltativo)

Specificate se desiderate CodeCatalyst ripristinare la distribuzione dello stack se fallisce. Il rollback riporta lo stack all'ultimo stato stabile conosciuto. L'impostazione predefinita è abilitare i rollback. Per ulteriori informazioni su questa impostazione, vedete il parametro CloudFormation [deploy](#) nel AWS CLI Command Reference.

Per ulteriori informazioni su come l'azione Deploy CloudFormation stack gestisce i rollback, consulta [Configurazione dei rollback](#)

Per ulteriori informazioni sul rollback di uno stack, consulta [Stack failure options](#) nella Guida per l'utente.AWS CloudFormation

Interfaccia utente corrispondente: scheda Configurazione/Avanzata/ Disabilita il rollback

termination-protection

(*DeployCloudFormationStack*/Configuration/termination-protection)

(Facoltativo)

Specificate se desiderate che lo stack Deploy aggiunga la protezione dalla terminazione allo CloudFormation stack che sta distribuendo. Se un utente tenta di eliminare uno stack con protezione da cessazione abilitata, l'eliminazione ha esito negativo e lo stack, incluso lo stato, rimane invariato. L'impostazione predefinita è disabilitare la protezione dalla terminazione. Per ulteriori informazioni, consulta [Proteggere uno stack dall'eliminazione nella Guida](#) per l'AWS CloudFormation utente.

Interfaccia utente corrispondente: scheda di configurazione/Avanzata/Protezione da terminazione

timeout-in-minutes

(*DeployCloudFormationStack*/Configuration/timeout-in-minutes)

(Facoltativo)

Specificate la quantità di tempo, in minuti, da dedicare prima di scadere le CloudFormation operazioni di creazione dello stack e impostare lo stato dello stack su. `CREATE_FAILED` Se CloudFormation non è in grado di creare l'intero stack nel tempo assegnato, produce un errore di timeout ed esegue il rollback dello stack.

Per impostazione predefinita, non è previsto alcun timeout per la creazione dello stack. Tuttavia, per la natura del servizio che implementano, alcune singole risorse possono avere un proprio timeout. Ad esempio, se una singola risorsa nello stack produce un timeout, anche la creazione dello stack viene arrestata, anche se il timeout specificato per la creazione dello stack non è ancora stato raggiunto.

Interfaccia utente corrispondente: scheda Configurazione/Avanzata/timeout CloudFormation

notification-arns

(*DeployCloudFormationStack*/Configuration/notification-arns)

(Facoltativo)

Specificate l'ARN di un argomento Amazon SNS a cui CodeCatalyst desiderate inviare messaggi di notifica. Ad esempio, `arn:aws:sns:us-east-1:111222333:MyTopic`. Quando viene eseguita l'azione Deploy CloudFormation stack, si CodeCatalyst coordina con CloudFormation per inviare una notifica per CloudFormation evento che si verifica durante il processo di creazione o aggiornamento dello stack. (Gli eventi sono visibili nella scheda Eventi della CloudFormation console relativa allo stack.) È possibile specificare fino a cinque argomenti. Per ulteriori informazioni, consulta [Cos'è Amazon SNS?](#)

Interfaccia utente corrispondente: scheda Configurazione/Avanzata/Notifica ARNs

monitor-alarm-arns

(*DeployCloudFormationStack*/Configuration/monitor-alarm-arns)

(Facoltativo)

Specificare l'Amazon Resource Name (ARN) di un CloudWatch allarme Amazon da utilizzare come trigger di rollback. Ad esempio, `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Puoi avere un massimo di cinque trigger di rollback.

Note

Se specifichi un ARN di CloudWatch allarme, dovrai anche configurare autorizzazioni aggiuntive per abilitare l'azione di accesso. CloudWatch Per ulteriori informazioni, consulta [Configurazione dei rollback](#).

Interfaccia utente corrispondente: scheda di configurazione/Avanzata/Monitor alarm ARNs

monitor-timeout-in-minutes

(*DeployCloudFormationStack*/Configuration/monitor-timeout-in-minutes)

(Facoltativo)

Specificare un periodo di tempo, da 0 a 180 minuti, durante il quale CloudFormation monitorare gli allarmi specificati. Il monitoraggio inizia dopo che tutte le risorse dello stack sono state distribuite. Se l'allarme si verifica entro il periodo di monitoraggio specificato, l'implementazione fallisce e CloudFormation ripristina l'intera operazione dello stack.

Impostazione predefinita: 0. CloudFormation monitora gli allarmi solo mentre le risorse dello stack vengono distribuite, non dopo.

Interfaccia utente corrispondente: scheda Configurazione/Avanzata/Tempo di monitoraggio

tags

(*DeployCloudFormationStack*/Configuration/tags)

(Facoltativo)

Specificate i tag da allegare allo stack. CloudFormation I tag sono coppie chiave-valore arbitrarie che è possibile utilizzare per identificare lo stack per scopi quali l'allocazione dei costi. Per ulteriori informazioni su cosa sono i tag e su come possono essere utilizzati, consulta [Tagging your resources](#) nella Amazon EC2 User Guide. Per ulteriori informazioni sull'aggiunta di tag CloudFormation, consulta [Impostazione delle opzioni CloudFormation dello stack](#) nella Guida per l'AWS CloudFormation utente.

Una chiave può contenere caratteri alfanumerici o spazi e può contenere fino a 127 caratteri. Un valore può contenere caratteri alfanumerici o spazi e può contenere fino a 255 caratteri.

È possibile aggiungere fino a 50 tag univoci per ogni stack.

Interfaccia utente corrispondente: Configuration Tab/Advanced/ Tags

Implementazione di un' AWS CDK app con un flusso di lavoro

Questa sezione descrive come distribuire un' AWS Cloud Development Kit (AWS CDK) app nel tuo AWS account utilizzando un flusso di lavoro. A tale scopo, è necessario aggiungere l'azione di AWS CDK distribuzione al flusso di lavoro. L'azione di AWS CDK distribuzione sintetizza e distribuisce l'app in. AWS Cloud Development Kit (AWS CDK) AWS Se l'app esiste già in AWS, l'azione la aggiorna se necessario.

Per informazioni generali sulla scrittura di app utilizzando il AWS CDK, vedi [Cos'è il AWS CDK?](#) nella Guida per gli AWS Cloud Development Kit (AWS CDK) sviluppatori.

Argomenti

- [Quando utilizzare l'azione «AWS CDK deploy»](#)
- [Come funziona l'azione 'AWS CDK deploy'](#)
- [Versioni CDK CLI utilizzate dall'azione 'deploy'AWS CDK](#)
- [Immagine di runtime utilizzata dall'azione 'deploy'AWS CDK](#)
- [Quanti stack può dispiegare l'azione?](#)
- [Esempio: distribuzione di un'app AWS CDK](#)
- [Aggiungere l'azione 'AWS CDK deploy'](#)
- [variabili 'AWS CDK deploy'](#)
- [AWS CDK azione 'distribuisci' YAML](#)

Quando utilizzare l'azione «AWS CDK deploy»

Utilizza questa azione se hai sviluppato un'app utilizzando e ora desideri distribuirla AWS CDK automaticamente come parte del flusso di lavoro automatizzato di integrazione e distribuzione continue (CI/CD). Ad esempio, potresti voler distribuire la tua AWS CDK app automaticamente ogni volta che qualcuno unisce una pull request relativa alla fonte dell'app. AWS CDK

Come funziona l'azione 'AWS CDK deploy'

La AWS CDK distribuzione funziona nel modo seguente:

1. [In fase di esecuzione, se hai specificato la versione 1.0.12 o precedente dell'azione, l'azione scarica l'ultima CLI CDK \(chiamata anche Toolkit\) AWS CDK nell'immagine dell'ambiente di runtime. CodeCatalyst](#)

Se hai specificato la versione 1.0.13 o successiva, l'azione viene fornita in bundle con una [versione specifica](#) della CLI CDK, quindi non viene effettuato alcun download.

2. L'azione utilizza la CLI CDK per eseguire il comando `cdk deploy`. Questo comando sintetizza e distribuisce l'app in AWS CDK AWS. Per ulteriori informazioni su questo comando, consultate l'argomento [AWS CDK Toolkit \(comando cdk\)](#) nella Guida per gli sviluppatori AWS Cloud Development Kit (AWS CDK)

Versioni CDK CLI utilizzate dall'azione 'deploy' AWS CDK

La tabella seguente mostra quale versione della CLI CDK viene utilizzata per impostazione predefinita dalle diverse versioni AWS CDK dell'azione di distribuzione.

Note

Potresti essere in grado di sovrascrivere l'impostazione predefinita. Per ulteriori informazioni, consulta [CdkCliVersion](#) nella [AWS CDK azione 'distribuisci' YAML](#).

versione dell'AWS CDK azione 'deploy'	AWS CDK Versione CLI
1.0.0 — 1.0.12	più recente
1.0.13 o successivo	2.99.1

Immagine di runtime utilizzata dall'azione 'deploy' AWS CDK

La tabella seguente mostra le immagini dell'ambiente di runtime CodeCatalyst utilizzate per eseguire diverse versioni dell'azione di AWS CDK distribuzione. Le immagini includono diversi set di strumenti preinstallati. Per ulteriori informazioni, consulta [Immagini attive](#).

Note

Ti consigliamo di aggiornare l'azione di AWS CDK distribuzione alla versione 2.x per sfruttare gli strumenti più recenti disponibili nell'immagine di marzo 2024. Per aggiornare l'azione, impostane la `Identifier` proprietà su nel file di definizione del flusso di lavoro `aws/cdk-deploy@v2`. Per ulteriori informazioni, consulta [AWS CDK azione 'distribuisci' YAML](#).

versione dell'AWS CDK azione 'deploy'	Immagini dell'ambiente di runtime
1.x	Immagini di novembre 2022
2.x	Immagini di marzo 2024

Quanti stack può dispiegare l'azione?

La AWS CDK distribuzione può implementare solo un singolo stack. Se la tua AWS CDK app è composta da più stack, devi creare uno stack principale con stack annidati e distribuire lo stack principale utilizzando questa azione.

Esempio: distribuzione di un'app AWS CDK

Il seguente flusso di lavoro di esempio include l'azione AWS CDK deploy, insieme all'AWS CDK azione bootstrap. Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Questo repository contiene la tua AWS CDK app. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione di AWS CDK bootstrap (CDKBootstrap): all'attivazione, l'azione distribuisce lo stack di CDKToolkit bootstrap in AWS. Se lo CDKToolkit stack esiste già nell'ambiente, verrà aggiornato se necessario; in caso contrario, non succede nulla e l'azione viene contrassegnata come riuscita.
- An AWS CDK deploy action (AWS CDK Deploy): al termine dell'azione AWS CDK bootstrap, l'azione AWS CDK deploy sintetizza il codice dell' AWS CDK app in un CloudFormation modello e distribuisce lo stack definito nel modello. AWS

Note

Il seguente esempio di flusso di lavoro è a scopo illustrativo e non funzionerà senza una configurazione aggiuntiva.

Note

Nel codice YAML che segue, puoi omettere le `Connections`: sezioni se lo desideri. Se ometti queste sezioni, devi assicurarti che il ruolo specificato nel campo `Default IAM role` nel tuo ambiente includa le autorizzazioni e le policy di fiducia richieste dalle azioni `bootstrap` e `deploy`. AWS CDK AWS CDK Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta [Creazione di un ambiente](#). Per ulteriori informazioni sulle autorizzazioni e sulle politiche di attendibilità richieste dalle azioni AWS CDK `bootstrap` e AWS CDK `deploy`, consulta la descrizione della `Role` proprietà nel e. [azione 'AWS CDK bootstrap' YAML](#) [AWS CDK azione 'distribuisci' YAML](#)

```
Name: codecatalyst-cdk-deploy-workflow
SchemaVersion: 1.0
```

Triggers:

- Type: PUSH
- Branches:
 - main

Actions:**CDKBootstrap:**

```
Identifier: aws/cdk-bootstrap@v2
```

Inputs:**Sources:**

- WorkflowSource

Environment:

```
Name: codecatalyst-cdk-deploy-environment
```

Connections:

- Name: codecatalyst-account-connection
- Role: codecatalyst-cdk-bootstrap-role

Configuration:

```
Region: us-west-2
```

CDKDeploy:

```
Identifier: aws/cdk-deploy@v2
DependsOn:
  - CDKBootstrap
Environment:
  Name: codecatalyst-cdk-deploy-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-cdk-deploy-role
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  StackName: my-app-stack
  Region: us-west-2
```

Aggiungere l'azione 'AWS CDK deploy'

Utilizza le seguenti istruzioni per aggiungere l'azione di AWS CDK distribuzione al tuo flusso di lavoro.

Prima di iniziare

Prima di aggiungere l'azione di AWS CDK distribuzione al flusso di lavoro, completa le seguenti attività:

1. Tieni pronta un' AWS CDK app. Puoi scrivere la tua AWS CDK app usando AWS CDK v1 o v2, in qualsiasi linguaggio di programmazione supportato da AWS CDK. Assicurati che i file della tua AWS CDK app siano disponibili in:
 - Un [archivio CodeCatalyst di sorgenti](#), oppure
 - Un [artefatto CodeCatalyst di output](#) generato da un'altra azione del flusso di lavoro
2. Avvia il tuo ambiente AWS. Per eseguire il bootstrap, puoi:
 - Utilizza uno dei metodi descritti in [Come eseguire il bootstrap nella Guida](#) per gli AWS Cloud Development Kit (AWS CDK) sviluppatori.
 - Usa l'azione AWS CDK bootstrap. Puoi aggiungere questa azione nello stesso flusso di lavoro della AWS CDK distribuzione o in uno diverso. Assicurati solo che l'azione bootstrap venga eseguita almeno una volta prima di eseguire l'azione di AWS CDK distribuzione in modo che siano disponibili le risorse necessarie. Per ulteriori informazioni sull'azione AWS CDK bootstrap, consulta [Avvio di un' AWS CDK app con un flusso di lavoro](#)

Per ulteriori informazioni sul bootstrap, consulta [Bootstrap](#) nella Developer Guide.AWS Cloud Development Kit (AWS CDK)

Visual

Per aggiungere l'azione 'AWS CDK deploy' utilizzando l'editor visuale

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di AWS CDK distribuzione ed esegui una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli AWS CDK deploy. Viene visualizzata la finestra di dialogo dei dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprirne il riquadro di configurazione.
10. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, vedi. [AWS CDK azione 'distribuisci' YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.

12. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

 Note

Se l'azione di AWS CDK distribuzione fallisce con un `npm install` errore, consulta [Come posso correggere gli errori di «npm install»? per informazioni su come correggere l'errore.](#)

YAML

Per aggiungere l'azione 'AWS CDK deploy' utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di AWS CDK distribuzione ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli AWS CDK deploy. Viene visualizzata la finestra di dialogo dei dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Scarica per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprirne il riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [AWS CDK azione 'distribuisci' YAML](#)

11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

 Note

Se l'azione di AWS CDK distribuzione fallisce con un `npm install` errore, consulta [Come posso correggere gli errori di «npm install»? per informazioni su come correggere l'errore.](#)

variabili 'AWS CDK deploy'

L'azione AWS CDK deploy produce e imposta le seguenti variabili in fase di esecuzione. Queste sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

Chiave	Valore
stack-id	L'Amazon Resource Name (ARN) dello stack di AWS CDK applicazioni su cui è stato distribuito durante l'esecuzione del flusso di lavoro. Ad esempio: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-app-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code>
piattaforma di distribuzione	Il nome della piattaforma di distribuzione. Codificato in. <code>AWS:CloudFormation</code>
region	Il codice regionale su Regione AWS cui è stato distribuito durante l'esecuzione del flusso di lavoro.

Chiave	Valore
	Ad esempio: <code>us-west-2</code>
SALTA LA DISTRIBUZIONE	Il valore di <code>true</code> indica che la distribuzione dello stack di AWS CDK applicazioni è stata ignorata durante l'esecuzione del flusso di lavoro. Una distribuzione dello stack verrà ignorata se non vi sono modifiche nello stack dall'ultima distribuzione. Questa variabile viene prodotta solo se il suo valore è <code>true</code> Codificato in <code>true</code>
CloudFormation variabili	Oltre a generare le variabili elencate in precedenza, l'azione di AWS CDK distribuzione espone anche le variabili di CloudFormation output come variabili del flusso di lavoro da utilizzare nelle azioni successive del flusso di lavoro. Per impostazione predefinita, l'azione espone solo le prime quattro (o meno) CloudFormation variabili trovate. Per determinare quali sono esposte, esegui l'azione di AWS CDK distribuzione una volta, quindi consulta la scheda Variabili della pagina dei dettagli di esecuzione. Se le variabili elencate nella scheda Variabili non sono quelle desiderate, puoi configurarne di diverse utilizzando la proprietà <code>CfnOutputVariables</code> in YAML. Per ulteriori informazioni, consultate la descrizione della CfnOutputVariables proprietà in AWS CDK azione 'distribuisci' YAML

AWS CDK azione 'distribuisci' YAML

Di seguito è riportata la definizione YAML dell'azione di AWS CDK distribuzione. Per informazioni su come utilizzare questa azione, consulta [Implementazione di un' AWS CDK app con un flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.  
# See Proprietà di primo livello for details.
```

```
Name: MyWorkflow  
SchemaVersion: 1.0  
Actions:
```

```
# The action definition starts here.
```

```
CDKDeploy\_nn:
```

```
  Identifier: aws/cdk-deploy@v2
```

```
  DependsOn:
```

```
    - CDKBootstrap
```

```
  Compute:
```

```
    Type: EC2 | Lambda
```

```
    Fleet: fleet-name
```

```
  Timeout: timeout-minutes
```

```
  Inputs:
```

```
    # Specify a source or an artifact, but not both.
```

```
    Sources:
```

```
      - source-name-1
```

```
    Artifacts:
```

```
      - artifact-name
```

```
  Outputs:
```

```
    Artifacts:
```

```
      - Name: cdk_artifact
```

```

Files:
- "cdk.out/**/*"
Environment:
Name: environment-name
Connections:
- Name: account-connection-name
  Role: iam-role-name
Configuration:
StackName: my-cdk-stack
Region: us-west-2
Tags: '{"key1": "value1", "key2": "value2"}'
Context: '{"key1": "value1", "key2": "value2"}'
CdkCliVersion: version
CdkRootPath: directory-containing-cdk.json-file
CfnOutputVariables: '["CfnOutputKey1", "CfnOutputKey2", "CfnOutputKey3"]'
CloudAssemblyRootPath: path-to-cdk.out

```

CDKDeploy

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: `CDKDeploy_nn`.

Interfaccia utente corrispondente: scheda di configurazione/nome dell'azione

Identifier

(*CDKDeploy*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Note

La specificazione `aws/cdk-deploy@v2` causa l'esecuzione dell'azione sull'[immagine di marzo 2024](#) che include strumenti più recenti come Node.js 18. La specifica `aws/cdk-`

`deploy@v1` fa sì che l'azione venga eseguita sull'[immagine di novembre 2022](#) che include strumenti meno recenti come Node.js 16.

Default: `aws/cdk-deploy@v2`.

Interfaccia utente corrispondente: Workflow diagram/ _nn/ CDKDeploy `aws/cdk-deploy @v2` label

DependsOn

(*CDKDeploy*/DependsOn)

(Facoltativo)

Specificare un'azione o un gruppo di azioni che deve essere eseguito correttamente per l'esecuzione dell'azione di distribuzione.AWS CDK Ti consigliamo di specificare l'azione AWS CDK bootstrap nella DependsOn proprietà, in questo modo:

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v2
  DependsOn:
    - CDKBootstrap
```

Note

Il [bootstrap](#) è un prerequisito obbligatorio per la distribuzione di un'app. AWS CDK Se non includi l'azione AWS CDK Bootstrap nel tuo flusso di lavoro, devi trovare un altro modo per distribuire lo stack AWS CDK bootstrap prima di eseguire l'azione di distribuzione.AWS CDK Per ulteriori informazioni, consulta [Aggiungere l'azione 'AWS CDK deploy' in Implementazione di un' AWS CDK app con un flusso di lavoro](#).

Per ulteriori informazioni sulla funzionalità 'depends on', consulta. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Ingressi/Dipende da - opzionale

Compute

(*CDKDeploy*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*CDKDeploy*/Compute/Type)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: configurazione tab/Advanced : opzionale/tipo di calcolo

Fleet

(*CDKDeploy*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand:,. Linux.x86-64.Large Linux.x86-64.XLarge Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere. [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omissa, l'impostazione predefinita è. `Linux.x86-64.Large`

Interfaccia utente corrispondente: Configurazione tab/Advanced : parco opzionale/di calcolo

Timeout

(*CDKDeploy*/Timeout)

(Obbligatorio)

Specificate il periodo di tempo, in minuti (editor YAML) o in ore e minuti (editor visivo), che l'azione può essere eseguita prima che termini l'azione. CodeCatalyst Il minimo è 5 minuti e il massimo è descritto in. [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Inputs

(*CDKDeploy*/Inputs)

(Facoltativo)

La Inputs sezione definisce i dati CDKDeploy necessari durante l'esecuzione di un flusso di lavoro.

Note

È consentito un solo input (una fonte o un artefatto) per ogni azione di AWS CDK distribuzione.

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*CDKDeploy*/Inputs/Sources)

(Richiesto se l' AWS CDK app che desideri distribuire è archiviata in un repository di origine)

Se AWS CDK l'app è archiviata in un repository di origine, specifica l'etichetta di tale repository di origine. L'azione AWS CDK deploy sintetizza l'app in questo repository prima di iniziare il processo di distribuzione. Attualmente, l'unica etichetta supportata è `WorkflowSource`

Se AWS CDK l'app non è contenuta in un repository di origine, deve risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*CDKDeploy*/Inputs/Artifacts)

(Richiesto se l' AWS CDK app che desideri distribuire è archiviata in un elemento di [output di un'azione precedente](#))

Se AWS CDK l'app è contenuta in un artefatto generato da un'azione precedente, specificalo qui. L'azione AWS CDK deploy sintetizza l'app nell'artefatto specificato in un modello prima di iniziare il processo di distribuzione. CloudFormation Se AWS CDK l'app non è contenuta in un artefatto, deve risiedere nel repository di origine.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Input/Artifacts - opzionale

Outputs

(*CDKDeploy*/Outputs)

(Facoltativo)

Definisce i dati emessi dall'azione durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Output

Artifacts - output

(*CDKDeploy*/Outputs/Artifacts)

(Facoltativo)

Specificate gli artefatti generati dall'azione. È possibile fare riferimento a questi artefatti come input in altre azioni.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Outputs/Artifacts

Name

(*CDKDeploy*/Outputs/Artifacts/Name)

(Richiesto se incluso) [Artifacts - output](#)

Specificate il nome dell'artefatto che conterrà il CloudFormation modello sintetizzato dall'azione di AWS CDK distribuzione in fase di esecuzione. Il valore predefinito è `cdk_artifact`. Se non specificate un artefatto, l'azione sintetizza il modello ma non lo salva in un artefatto. Prendi in considerazione la possibilità di salvare il modello sintetizzato in un artefatto per conservarne una registrazione a scopo di test o risoluzione dei problemi.

Interfaccia utente corrispondente: restituisce il nome dell'artefatto/crea il nome dell'artefatto tab/Artifacts/Add

Files

(*CDKDeploy*/Outputs/Artifacts/Files)

(Richiesto se incluso) [Artifacts - output](#)

Specificate i file da includere nell'artefatto. È necessario specificare `"cdk.out/**/*"` di includere il modello sintetizzato CloudFormation dell' AWS CDK app.

Note

`cdk.out` è la directory predefinita in cui vengono salvati i file sintetizzati. Se avete specificato una directory di output diversa `cdk.out` da quella `cdk.json` del file, specificate quella directory qui invece di `cdk.out`

Interfaccia utente corrispondente: restituisce tab/Artifacts/Add artefatti/file prodotti da build

Environment

(*CDKDeploy*/Environment)

(Obbligatorio)

Specificare l'ambiente CodeCatalyst da utilizzare con l'azione. L'azione si connette all'Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs](#) e [Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*CDKDeploy*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*CDKDeploy*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / connessione all'account Environment/account/ role AWS

Name

(*CDKDeploy*/Environment/Connections/Name)

(Richiesto se è incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / connessione all'account Environment/account/ role AWS

Role

(*CDKDeploy*/Environment/Connections/Role)

(Richiesto se è incluso [Connections](#))

Specificare il nome della connessione all'account.

Specificate il nome del ruolo IAM AWS CDK utilizzato dall'azione di distribuzione per accedere AWS e distribuire lo stack di AWS CDK applicazioni. Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

- La seguente politica di autorizzazioni:

 Warning

Limita le autorizzazioni a quelle mostrate nella seguente politica. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackResources"
      ],
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::111122223333:role/cdk-*"
    }
  ]
}
```

```
    ]
}
```

- La seguente politica di fiducia personalizzata:

Note

Puoi usare il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#).

Tieni presente che il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) `tab/Environment/What La configurazione è attiva? my-environment /three dot menu/ Cambia ruolo`
- (Versioni precedenti) `Scheda di configurazione/' '/ Ruolo Environment/account/role`

Configuration

(*CDKDeploy*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

StackName

(*CDKDeploy*/Configuration/StackName)

(Obbligatorio)

Il nome dello stack di AWS CDK app, così come appare nel file `entrypoint` nella directory AWS CDK dell'app. `bin` L'esempio seguente mostra il contenuto di un file TypeScript `entrypoint`, con il nome

dello stack evidenziato in. *red italics* Se il file entrypoint è in una lingua diversa, avrà un aspetto simile.

```
import * as cdk from 'aws-cdk-lib';
import { CdkWorkshopTypescriptStack } from '../lib/cdk_workshop_typescript-stack';

const app = new cdk.App();
new CdkWorkshopTypescriptStack(app, 'CdkWorkshopTypescriptStack');
```

Puoi specificare solo uno stack.

Tip

Se disponi di più stack, puoi creare uno stack principale con pile annidate. È quindi possibile specificare lo stack principale in questa azione per distribuire tutti gli stack.

Interfaccia utente corrispondente: scheda di configurazione/nome dello stack

Region

(*CDKDeploy*/Configuration/Region)

(Facoltativo)

Specificare Regione AWS in che modo verrà distribuito lo stack AWS CDK dell'applicazione. Per un elenco dei codici regionali, consulta Endpoint [regionali](#).

Se non specifichi una regione, l'azione di AWS CDK distribuzione viene distribuita nella regione specificata nel codice. AWS CDK Per ulteriori informazioni, consulta [Environments](#) nella AWS Cloud Development Kit (AWS CDK) Developer Guide.

Interfaccia utente corrispondente: scheda/regione di configurazione

Tags

(*CDKDeploy*/Configuration/Tags)

(Facoltativo)

Specificate i tag che desiderate applicare alle AWS risorse nello stack di AWS CDK applicazioni. I tag vengono applicati allo stack stesso e alle singole risorse dello stack. Per ulteriori informazioni sui tag, consulta [Tagging](#) nella Developer Guide.AWS Cloud Development Kit (AWS CDK)

Interfaccia utente corrispondente: Configuration tab/Advanced - opzionale/ Tags

Context

(*CDKDeploy*/Configuration/Context)

(Facoltativo)

Specificate i contesti, sotto forma di coppie chiave-valore, da associare allo stack dell'applicazione. AWS CDK Per ulteriori informazioni sui contesti, consulta [Runtime contexts](#) nella Developer Guide.AWS Cloud Development Kit (AWS CDK)

Interfaccia utente corrispondente: Configurazione tab/Advanced - opzionale/ Context

CdkCliVersion

(*CDKDeploy*/Configuration/CdkCliVersion)

(Facoltativo)

Questa proprietà è disponibile con la versione 1.0.13 o successiva dell'azione AWS CDK deploy e la versione 1.0.8 o successiva dell'azione bootstrap.AWS CDK

Specifica una delle seguenti proprietà:

- La versione completa dell'interfaccia a riga di AWS Cloud Development Kit (AWS CDK) comando (CLI) (chiamata anche AWS CDK Toolkit) che si desidera utilizzare con questa azione. Esempio: 2.102.1. Valuta la possibilità di specificare una versione completa per garantire coerenza e stabilità durante la creazione e la distribuzione dell'applicazione.

Or

- latest. Valuta la possibilità latest di specificare se sfruttare le funzionalità e le correzioni più recenti della CLI CDK.

L'azione scaricherà la versione specificata (o la versione più recente) della AWS CDK CLI nell'[immagine di CodeCatalyst build](#), quindi utilizzerà questa versione per eseguire i comandi necessari per distribuire l'applicazione CDK o avviare il proprio ambiente. AWS

[Per un elenco delle versioni CDK CLI supportate che puoi utilizzare,AWS CDK consulta Versioni.](#)

Se si omette questa proprietà, l'azione utilizza una versione AWS CDK CLI predefinita descritta in uno dei seguenti argomenti:

- [Versioni CDK CLI utilizzate dall'azione 'deploy'AWS CDK](#)
- [Versioni CDK CLI utilizzate dall'azione AWS CDK "bootstrap»](#)

Interfaccia utente corrispondente: scheda di configurazione/versione AWS CDK CLI

CdkRootPath

(*CDKDeploy*/Configuration/CdkRootPath)

(Facoltativo)

Il percorso della directory che contiene il file del AWS CDK `cdk.json` progetto. L'azione AWS CDK di distribuzione viene eseguita da questa cartella e tutti gli output creati dall'azione verranno aggiunti a questa directory. Se non è specificata, l'azione di AWS CDK distribuzione presuppone che il `cdk.json` file si trovi nella radice del progetto. AWS CDK

Interfaccia utente corrispondente: scheda di configurazione/ Directory in cui risiede `cdk.json`

CfnOutputVariables

(*CDKDeploy*/Configuration/CfnOutputVariables)

(Facoltativo)

Specificate quali CfnOutput costrutti del codice AWS CDK dell'applicazione desiderate esporre come variabili di output del flusso di lavoro. È quindi possibile fare riferimento alle variabili di output del flusso di lavoro nelle azioni successive del flusso di lavoro. Per ulteriori informazioni sulle variabili in CodeCatalyst, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Ad esempio, se il codice AWS CDK dell'applicazione è simile al seguente:

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);
    const bucket = new s3.Bucket(this, 'amzn-s3-demo-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
```



```
new CfnOutput(this, 'bucketName', {
  value: bucket.bucketName,
  description: 'The name of the s3 bucket',
  exportName: 'amzn-s3-demo-bucket',
});
const table = new dynamodb.Table(this, 'todos-table', {
  partitionKey: {name: 'todoId', type: dynamodb.AttributeType.NUMBER},
  billingMode: dynamodb.BillingMode.PAY_PER_REQUEST,
  removalPolicy: RemovalPolicy.DESTROY,
})
new CfnOutput(this, 'tableName', {
  value: table.tableName,
  description: 'The name of the dynamodb table',
  exportName: 'myDynamoDbTable',
});
...
}
```

... e la tua `CfnOutputVariables` proprietà ha questo aspetto:

Configuration:

```
...
CfnOutputVariables: ['bucketName','tableName']
```

... quindi l'azione genera le seguenti variabili di output del flusso di lavoro:

Chiave	Valore
bucketName	bucket.bucketName
tableName	table.tableName

È quindi possibile fare riferimento alle `tableName` variabili `bucketName` and nelle azioni successive. Per informazioni su come fare riferimento alle variabili di output del flusso di lavoro nelle azioni successive, consulta [Riferimento a una variabile predefinita](#).

Se non specificate alcun `CfnOutput` costruito nella `CfnOutputVariables` proprietà, l'azione espone le prime quattro (o meno) variabili di output trovate come variabili di CloudFormation output del flusso di lavoro. Per ulteriori informazioni, consulta [variabili 'AWS CDK deploy'](#).

Tip

Per ottenere un elenco di tutte le variabili di CloudFormation output prodotte dall'azione, esegui una volta il flusso di lavoro contenente l'azione di AWS CDK distribuzione, quindi consulta la scheda Registri dell'azione. I log contengono un elenco di tutte le variabili di CloudFormation output associate all'app. AWS CDK Una volta che sapete quali sono tutte le CloudFormation variabili, potete specificare quelle che desiderate convertire in variabili di output del flusso di lavoro utilizzando la `CfnOutputVariables` proprietà.

Per ulteriori informazioni sulle variabili di CloudFormation output, consultate la documentazione relativa al `CfnOutput` costruito, disponibile all'indirizzo [class CfnOutput \(construct\)](#) nell'AWS Cloud Development Kit (AWS CDK) API Reference.

Interfaccia utente corrispondente: scheda di configurazione/variabili di output CloudFormation

CloudAssemblyRootPath

(*CDKDeploy*/Configuration/CloudAssemblyRootPath)

(Facoltativo)

Se hai già sintetizzato lo stack dell' AWS CDK app in un assembly cloud (utilizzando l'`cdk synth` operazione), specifica il percorso principale della directory dell'assembly cloud (`cdk.out`). Il CloudFormation modello che si trova nella directory di assemblaggio cloud specificata verrà distribuito dall'azione AWS CDK deploy nell'utente utilizzando il comando. Account AWS `cdk deploy --app` Quando l'`--app` opzione è presente, l'`cdk synth` operazione non viene eseguita.

Se non si specifica una directory di assemblaggio cloud, l'azione AWS CDK deploy eseguirà il `cdk deploy` comando senza l'`--app` opzione. Senza l'`--app` opzione, l'`cdk deploy` operazione sintetizzerà (`cdk synth`) e distribuirà l'app nel tuo AWS CDK . Account AWS

Perché dovrei specificare un assembly cloud esistente e sintetizzato quando l'azione "AWS CDK deploy» può eseguire la sintesi in fase di esecuzione?

Potresti voler specificare un assembly cloud esistente e sintetizzato per:

- Assicurati che venga distribuito lo stesso identico set di risorse ogni volta che viene eseguita l'azione "AWS CDK deploy»

Se non specifichi un assembly cloud, è possibile che l'azione di distribuzione sintetizzi e AWS CDK distribuisca file diversi a seconda di quando viene eseguita. Ad esempio, l'azione di AWS CDK implementazione potrebbe sintetizzare un assembly cloud con un set di dipendenze durante una fase di test e un altro set di dipendenze durante una fase di produzione (se tali dipendenze sono cambiate tra le fasi). Per garantire l'esatta parità tra ciò che viene testato e ciò che viene distribuito, consigliamo di eseguire la sintesi una volta e poi utilizzare il campo della directory Path to cloud assembly (editor visivo) o la `CloudAssemblyRootPath` proprietà (editor YAML) per specificare l'assembly cloud già sintetizzato.

- Usa AWS CDK gestori di pacchetti e strumenti non standard con l'app

Durante un'`synth`operazione, l'azione AWS CDK deploy tenta di eseguire l'app utilizzando strumenti standard come npm o pip. Se l'azione non riesce a eseguire correttamente l'app utilizzando tali strumenti, la sintesi non verrà eseguita e l'azione avrà esito negativo. Per risolvere questo problema, puoi specificare i comandi esatti necessari per eseguire correttamente l'app nel `cdk.json` file dell' AWS CDK app e quindi sintetizzare l'app utilizzando un metodo che non prevede l'azione di AWS CDK distribuzione. Dopo che l'assembly cloud è stato generato, è possibile specificarlo nel campo Path to cloud assembly directory (editor visivo) o nella **`CloudAssemblyRootPath`** proprietà (editor YAML) dell'azione di distribuzione.AWS CDK

Per informazioni sulla configurazione del `cdk.json` file per includere i comandi per l'installazione e l'esecuzione dell' AWS CDK app, consulta [Specificazione](#) del comando dell'app.

[Per informazioni sui `cdk synth` comandi `cdk deploy` and e sull'`--appopzione`, consulta \[Deploying stacks, Synthesizing stacks e Skipping synthesis nella Developer Guide.AWS Cloud Development Kit \\(AWS CDK\\)\]\(#\)](#)

[Per informazioni sugli assembly cloud, consulta \[Cloud Assembly nell'API Reference.AWS Cloud Development Kit \\(AWS CDK\\)\]\(#\)](#)

Interfaccia utente corrispondente: scheda di configurazione/percorso della directory di assemblaggio cloud

Avvio di un' AWS CDK app con un flusso di lavoro

Questa sezione descrive come avviare un'applicazione utilizzando un AWS CDK flusso di lavoro. CodeCatalyst A tale scopo, è necessario aggiungere l'azione AWS CDK bootstrap al flusso di lavoro. [L'azione AWS CDK bootstrap fornisce uno stack di bootstrap nell' AWS ambiente utilizzando](#)

[il modello moderno](#). Se esiste già uno stack di bootstrap, l'azione lo aggiorna se necessario. La presenza di uno stack di bootstrap AWS è un prerequisito per la distribuzione di un'app. AWS CDK

[Per ulteriori informazioni sul bootstrap, consulta Bootstrapping nella Developer Guide.AWS Cloud Development Kit \(AWS CDK\)](#)

Argomenti

- [Quando usare l'azione 'bootstrap'AWS CDK](#)
- [Come funziona l'azione 'AWS CDK bootstrap'](#)
- [Versioni CDK CLI utilizzate dall'azione AWS CDK "bootstrap»](#)
- [Immagine di runtime utilizzata dall'azione 'bootstrap'AWS CDK](#)
- [Esempio: avvio di un'app AWS CDK](#)
- [Aggiungere l'azione 'AWS CDK bootstrap'](#)
- [variabili 'AWS CDK bootstrap'](#)
- [azione 'AWS CDK bootstrap' YAML](#)

Quando usare l'azione 'bootstrap'AWS CDK

Utilizza questa azione se disponi di un flusso di lavoro che distribuisce un' AWS CDK app e desideri distribuire (e aggiornare, se necessario) lo stack di bootstrap contemporaneamente. In questo caso, aggiungeresti l'azione AWS CDK bootstrap allo stesso flusso di lavoro di quello che distribuisce l'app. AWS CDK

Non utilizzare questa azione se si verifica una delle seguenti condizioni:

- Hai già distribuito uno stack di bootstrap utilizzando un altro meccanismo e desideri mantenerlo intatto (nessun aggiornamento).
- Vuoi usare un modello di [bootstrap personalizzato](#), che non è supportato con l'azione bootstrap.AWS CDK

Come funziona l'azione 'AWS CDK bootstrap'

Il AWS CDK bootstrap funziona come segue:

1. [In fase di esecuzione, se hai specificato la versione 1.0.7 o precedente dell'azione, l'azione scarica l'ultima CLI CDK \(chiamata anche AWS CDK Toolkit\) nell'immagine di build. CodeCatalyst](#)

Se hai specificato la versione 1.0.8 o successiva, l'azione viene fornita in bundle con una [versione specifica](#) della CLI CDK, quindi non viene effettuato alcun download.

2. L'azione utilizza la CLI CDK per eseguire il comando `cdk bootstrap`. Questo comando esegue le attività di bootstrap descritte nell'argomento [Bootstrap](#) della Guida per gli sviluppatori AWS Cloud Development Kit (AWS CDK)

Versioni CDK CLI utilizzate dall'azione AWS CDK "bootstrap»

La tabella seguente mostra quale versione della CLI CDK viene utilizzata per impostazione predefinita da diverse versioni AWS CDK dell'azione bootstrap.

Note

Potresti essere in grado di sovrascrivere l'impostazione predefinita. Per ulteriori informazioni, consulta [CdkCliVersion](#) nella [azione 'AWS CDK bootstrap' YAML](#).

versione d'azione 'AWS CDK bootstrap'	AWS CDK Versione CLI
1.0.0 — 1.0.7	più recente
1.0.8 o versione successiva	2.99.1

Immagine di runtime utilizzata dall'azione 'bootstrap'AWS CDK

La tabella seguente mostra le immagini dell'ambiente di runtime CodeCatalyst utilizzate per eseguire diverse versioni dell'azione AWS CDK bootstrap. Le immagini includono diversi set di strumenti preinstallati. Per ulteriori informazioni, consulta [Immagini attive](#).

Note

Ti consigliamo di aggiornare l'azione AWS CDK bootstrap alla versione 2.x per sfruttare gli strumenti più recenti disponibili nell'immagine di marzo 2024. Per aggiornare l'azione, impostane la `Identifier` proprietà su nel file di definizione del flusso di lavoro `aws/cdk-bootstrap@v2`. Per ulteriori informazioni, consulta [AWS CDK azione 'distribuisci' YAML](#).

versione d'azione 'AWS CDK bootstrap'	Immagini dell'ambiente di runtime
1.x	Immagini di novembre 2022
2.x	Immagini di marzo 2024

Esempio: avvio di un'app AWS CDK

Fate riferimento a [Esempio: distribuzione di un'app AWS CDK](#) in [Implementazione di un' AWS CDK app con un flusso di lavoro](#) per un flusso di lavoro che include l'AWS CDK azione bootstrap.

Aggiungere l'azione 'AWS CDK bootstrap'

Utilizza le seguenti istruzioni per aggiungere l'azione bootstrap al tuo flusso di lavoro AWS CDK .

Prima di iniziare

Prima di poter utilizzare l'azione AWS CDK bootstrap, assicurati di avere un' AWS CDK app pronta. L'azione bootstrap sintetizzerà l'app prima del AWS CDK bootstrap. Puoi scrivere la tua app in qualsiasi linguaggio di programmazione supportato da AWS CDK

Assicurati che i file AWS CDK dell'app siano disponibili in:

- Un [archivio CodeCatalyst di sorgenti](#), oppure
- Un [artefatto CodeCatalyst di output](#) generato da un'altra azione del flusso di lavoro

Visual

Per aggiungere l'azione 'AWS CDK bootstrap' utilizzando l'editor visuale

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.

6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione AWS CDK bootstrap ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.
- Or
- Scegli AWS CDK bootstrap. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Nelle schede Ingressi, Configurazione e Uscite, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, vedi. [azione 'AWS CDK bootstrap' YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

Note

Se l'operazione di AWS CDK bootstrap fallisce con un `npm install` errore, consulta [Come posso correggere gli errori di «npm install»? per informazioni su come correggere l'errore.](#)

YAML

Per aggiungere l'azione 'AWS CDK bootstrap' usando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione AWS CDK bootstrap e scegli + per aggiungerla al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in [azione 'AWS CDK bootstrap' YAML](#)
11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

 Note

Se l'operazione di AWS CDK bootstrap fallisce con un `npm install` errore, consulta [Come posso correggere gli errori di «npm install»? per informazioni su come correggere l'errore.](#)

variabili 'AWS CDK bootstrap'

L'azione AWS CDK bootstrap produce e imposta le seguenti variabili in fase di esecuzione. Queste sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

Chiave	Valore
piattaforma di distribuzione	Il nome della piattaforma di distribuzione. Codificato in. <code>AWS::CloudFormation</code>

Chiave	Valore
region	Il codice regionale in cui è stato distribuito Regione AWS lo stack di AWS CDK bootstrap durante l'esecuzione del flusso di lavoro. Ad esempio: us-west-2
stack-id	L'Amazon Resource Name (ARN) dello stack di bootstrap distribuito AWS CDK . Ad esempio: arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-bootstrap-stack/6aad4380-100a-11ec-a10a-03b8a84d40df
SALTA LA DISTRIBUZIONE	Il valore di <code>true</code> indica che la distribuzione dello stack di AWS CDK bootstrap è stata ignorata durante l'esecuzione del flusso di lavoro. Una distribuzione dello stack verrà ignorata se non vi sono modifiche nello stack dall'ultima distribuzione. Questa variabile viene prodotta solo se il suo valore è. <code>true</code> Codificato in. <code>true</code>

azione 'AWS CDK bootstrap' YAML

Di seguito è riportata la definizione YAML dell'azione bootstrap.AWS CDK Per informazioni su come utilizzare questa azione, consulta. [Avvio di un' AWS CDK app con un flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKBootstrapAction\_nn:
  Identifier: aws/cdk-bootstrap@v2
  DependsOn:
    - action-name
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
    Artifacts:
      - artifact-name
  Outputs:
    Artifacts:
      - Name: cdk_bootstrap_artifacts
    Files:
      - "cdk.out/**/*"
  Environment:
    Name: environment-name
    Connections:
      - Name: account-connection-name
      - Role: iam-role-name
  Configuration:
    Region: us-west-2
    CdkCliVersion: version
```

CDKBootstrapAction

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: `CDKBootstrapAction_nn`.

Interfaccia utente corrispondente: scheda di configurazione/nome visualizzato dell'azione

Identifier

(*CDKBootstrapAction*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Note

La specificazione `aws/cdk-bootstrap@v2` causa l'esecuzione dell'azione sull'[immagine di marzo 2024](#) che include strumenti più recenti come Node.js 18. La specifica `aws/cdk-bootstrap@v1` fa sì che l'azione venga eseguita sull'[immagine di novembre 2022](#) che include strumenti meno recenti come Node.js 16.

Default: `aws/cdk-bootstrap@v2`.

Interfaccia utente corrispondente: Workflow diagram/ _nn/ CDKBootstrapAction aws/cdk-bootstrap@v2 label

DependsOn

(*CDKBootstrapAction*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente per l'esecuzione di questa azione.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Input/Dipende da - opzionale

Compute

(*CDKBootstrapAction*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*CDKBootstrapAction*/Compute/Type)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: configurazione tab/Advanced : opzionale/tipo di calcolo

Fleet

(*CDKBootstrapAction*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand: `Linux.x86-64.Large` `Linux.x86-64.XLarge` Per ulteriori informazioni sulle flotte on-demand, vedere [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere [Proprietà del parco veicoli assegnate](#)

Se `Fleet` viene omissso, l'impostazione predefinita è `Linux.x86-64.Large`

Interfaccia utente corrispondente: Configurazione tab/Advanced : parco opzionale/di calcolo

Timeout

(*CDKBootstrapAction*/Timeout)

(Obbligatorio)

Specificate il periodo di tempo, in minuti (editor YAML) o in ore e minuti (editor visivo), che l'azione può essere eseguita prima che termini l'azione. CodeCatalyst Il minimo è 5 minuti e il massimo è descritto in [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Inputs

(*CDKBootstrapAction*/Inputs)

(Facoltativo)

La `Inputs` sezione definisce i dati necessari all'azione AWS CDK bootstrap durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Input

Note

È consentito un solo input (fonte o artefatto) per ogni AWS CDK azione di bootstrap.

Sources

(*CDKBootstrapAction*/Inputs/Sources)

(Richiesto se AWS CDK l'app è archiviata in un archivio di origine)

Se AWS CDK l'app è archiviata in un repository di origine, specifica l'etichetta di tale repository di origine. L'azione AWS CDK bootstrap sintetizza l'app in questo repository prima di avviare il processo di bootstrap. Attualmente, l'unica etichetta di repository supportata è `WorkflowSource`

Se AWS CDK l'app non è contenuta in un repository di origine, deve risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*CDKBootstrapAction*/Inputs/Artifacts)

(Richiesto se AWS CDK l'app è memorizzata in un elemento [di output di un'azione precedente](#))

Se la tua AWS CDK app è contenuta in un artefatto generato da un'azione precedente, specifica quell'artefatto qui. L'azione AWS CDK bootstrap sintetizza l'app nell'artefatto specificato in un modello prima di iniziare il processo di avvio. CloudFormation Se AWS CDK l'app non è contenuta in un artefatto, deve risiedere nel repository di origine.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Input/Artifacts - opzionale

Outputs

(*CDKBootstrapAction*/Outputs)

(Facoltativo)

Definisce i dati emessi dall'azione durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Output

Artifacts - output

(*CDKBootstrapAction*/Outputs/Artifacts)

(Facoltativo)

Specificate gli artefatti generati dall'azione. È possibile fare riferimento a questi artefatti come input in altre azioni.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Outputs/Artifacts

Name

(*CDKBootstrapAction*/Outputs/Artifacts/Name)

(Richiesto se incluso) [Artifacts - output](#)

Specificate il nome dell'elemento che conterrà il CloudFormation modello sintetizzato dall'azione AWS CDK bootstrap in fase di esecuzione. Il valore predefinito è `cdk_bootstrap_artifacts`. Se non specificate un artefatto, l'azione sintetizza il modello, ma non lo salva in un artefatto. Valuta la possibilità di salvare il modello sintetizzato in un artefatto per conservarne una registrazione a scopo di test o risoluzione dei problemi.

Interfaccia utente corrispondente: restituisce il nome dell'artefatto/crea l'artefatto tab/Artifacts/Add

Files

(*CDKBootstrapAction*/Outputs/Artifacts/Files)

(Obbligatorio se incluso) [Artifacts - output](#)

Specificate i file da includere nell'artefatto. È necessario specificare `"cdk.out/**/*"` di includere il modello sintetizzato CloudFormation dell' AWS CDK app.

Note

`cdk.out` è la directory predefinita in cui vengono salvati i file sintetizzati. Se avete specificato una directory di output diversa `cdk.out` da quella `cdk.json` del file, specificate quella directory qui invece di `cdk.out`

Interfaccia utente corrispondente: restituisce tab/Artifacts/Add artefatti/file prodotti da build

Environment

(*CDKBootstrapAction*/Environment)

(Obbligatorio)

Specificare l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs e Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*CDKBootstrapAction*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*CDKBootstrapAction*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).

- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' '/ connessione all'account Environment/account/role AWS

Name

(*CDKBootstrapAction*/Environment/Connections/Name)

(Richiesto se è incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' '/ connessione all'account Environment/account/role AWS

Role

(*CDKBootstrapAction*/Environment/Connections/Role)

(Richiesto se è incluso [Connections](#))

Specificate il nome del ruolo IAM utilizzato dall'azione AWS CDK bootstrap per accedere AWS e aggiungere lo stack di bootstrap. Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le politiche appropriate.

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#).

Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza.

Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / Ruolo Environment/account/role

Configuration

(*CDKBootstrapAction*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

Region

(*CDKBootstrapAction*/Configuration/Region)

(Obbligatorio)

Specificare Regione AWS in che modo verrà distribuito lo stack di bootstrap. Questa regione deve corrispondere a quella in cui viene distribuita AWS CDK l'app. Per un elenco dei codici regionali, consulta [Endpoint regionali](#).

Interfaccia utente corrispondente: scheda di configurazione/regione

CdkCliVersion

(*CDKBootstrapAction*/Configuration/CdkCliVersion)

(Facoltativo)

Questa proprietà è disponibile con la versione 1.0.13 o successiva dell'azione AWS CDK deploy e la versione 1.0.8 o successiva dell'azione bootstrap.AWS CDK

Specifica una delle seguenti proprietà:

- La versione completa dell'interfaccia a riga di AWS Cloud Development Kit (AWS CDK) comando (CLI) (chiamata anche AWS CDK Toolkit) che si desidera utilizzare con questa azione. Esempio: 2.102.1. Valuta la possibilità di specificare una versione completa per garantire coerenza e stabilità durante la creazione e la distribuzione dell'applicazione.

Or

- latest. Valuta la possibilità latest di specificare se sfruttare le funzionalità e le correzioni più recenti della CLI CDK.

L'azione scaricherà la versione specificata (o la versione più recente) della AWS CDK CLI nell'[immagine di CodeCatalyst build](#), quindi utilizzerà questa versione per eseguire i comandi necessari per distribuire l'applicazione CDK o avviare il proprio ambiente. AWS

[Per un elenco delle versioni CDK CLI supportate che puoi utilizzare,AWS CDK consulta Versioni.](#)

Se si omette questa proprietà, l'azione utilizza una versione AWS CDK CLI predefinita descritta in uno dei seguenti argomenti:

- [Versioni CDK CLI utilizzate dall'azione 'deploy'AWS CDK](#)
- [Versioni CDK CLI utilizzate dall'azione AWS CDK "bootstrap»](#)

Interfaccia utente corrispondente: scheda di configurazione/versione AWS CDK CLI

Pubblicazione di file su Amazon S3 con un flusso di lavoro

Questa sezione descrive come pubblicare file su Amazon S3 utilizzando un CodeCatalyst flusso di lavoro. A tale scopo, devi aggiungere l'azione di pubblicazione di Amazon S3 al tuo flusso di lavoro. L'azione di pubblicazione di Amazon S3 copia i file da una directory di origine a un bucket Amazon S3. La directory dei sorgenti può risiedere in:

- Un [repository di origine](#), oppure
- Un [artefatto di output](#) generato da un'altra azione del flusso di lavoro

Argomenti

- [Quando usare l'azione «Pubblica su Amazon S3»](#)
- [Immagine di runtime utilizzata dall'azione 'Amazon S3 publish'](#)
- [Esempio: pubblicazione di file su Amazon S3](#)
- [Aggiungere l'azione «Amazon S3 publish»](#)
- [Azione «pubblicazione Amazon S3» YAML](#)

Quando usare l'azione «Pubblica su Amazon S3»

Usa questa azione se:

- Hai un flusso di lavoro che genera file che desideri archiviare in Amazon S3.

Ad esempio, potresti avere un flusso di lavoro che crea un sito Web statico da ospitare in Amazon S3. In questo caso, il flusso di lavoro includerebbe un'[azione di compilazione](#) per creare il codice HTML e i file di supporto del sito e un'azione di pubblicazione di Amazon S3 per copiare i file su Amazon S3.

- Hai un repository di origine che contiene i file che desideri archiviare in Amazon S3.

Ad esempio, potresti avere un repository di sorgenti con i file sorgente delle applicazioni che desideri archiviare ogni notte su Amazon S3.

Immagine di runtime utilizzata dall'azione 'Amazon S3 publish'

L'azione di pubblicazione di Amazon S3 viene eseguita su un'immagine di [novembre 2022](#). Per ulteriori informazioni, consulta [Immagini attive](#).

Esempio: pubblicazione di file su Amazon S3

Il seguente flusso di lavoro di esempio include l'azione di pubblicazione di Amazon S3, insieme a un'azione di compilazione. Il flusso di lavoro crea un sito Web di documentazione statico e quindi lo pubblica su Amazon S3, dove è ospitato. Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione di compilazione (BuildDocs): all'attivazione, l'azione crea un sito Web di documentazione statico (mkdocs build) e aggiunge i file HTML associati e i metadati di supporto a un elemento chiamato MyDocsSite. Per ulteriori informazioni sull'azione di compilazione, consulta [Creazione con flussi di lavoro](#).
- Un'azione di pubblicazione di Amazon S3 (PublishToS3): al termine dell'azione di creazione, questa azione copia il sito nell'MyDocsSite artefatto su Amazon S3 per l'hosting.

Note

Il seguente esempio di flusso di lavoro è a scopo illustrativo e non funzionerà senza una configurazione aggiuntiva.

Note

Nel codice YAML che segue, puoi omettere la `Connections:` sezione se lo desideri. Se ometti questa sezione, devi assicurarti che il ruolo specificato nel campo `Ruolo IAM` predefinito nel tuo ambiente includa le autorizzazioni e le policy di fiducia richieste dall'azione di pubblicazione di Amazon S3. Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta [Creazione di un ambiente](#). Per ulteriori informazioni sulle autorizzazioni e sulle politiche di attendibilità richieste dall'azione di pubblicazione di Amazon S3, consulta la descrizione della proprietà [Role](#) nel [Azione «pubblicazione Amazon S3» YAML](#).

Name: codecatalyst-s3-publish-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH
- Branches:
 - main

Actions:

BuildDocs:

```
Identifier: aws/build@v1
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  Steps:
    - Run: echo BuildDocs started on `date`
    - Run: pip install --upgrade pip
    - Run: pip install mkdocs
    - Run: mkdocs build
    - Run: echo BuildDocs completed on `date`
Outputs:
  Artifacts:
    - Name: MyDocsSite
    Files:
      - "site/**/*"
```

```
PublishToS3:
Identifier: aws/s3-publish@v1
Environment:
  Name: codecatalyst-s3-publish-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-s3-publish-build-role
Inputs:
  Sources:
    - WorkflowSource
  Artifacts:
    - MyDocsSite
Configuration:
  DestinationBucketName: amzn-s3-demo-bucket
  SourcePath: /artifacts/PublishToS3/MyDocSite/site
  TargetPath: my/docs/site
```

Aggiungere l'azione «Amazon S3 publish»

Utilizza le seguenti istruzioni per aggiungere l'azione di pubblicazione di Amazon S3 al tuo flusso di lavoro.

Visual

Per aggiungere l'azione «Amazon S3 publish» utilizzando l'editor visuale

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di pubblicazione di Amazon S3 ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Amazon S3 Publish. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Nelle schede Ingressi, Configurazione e Uscite, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, consulta la. [Azione «pubblicazione Amazon S3» YAML](#). Questo riferimento fornisce informazioni dettagliate su ogni campo (e sul valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

YAML

Per aggiungere l'azione «Amazon S3 publish» utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). [CodeCatalyst](#)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di pubblicazione di Amazon S3 ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli Amazon S3 Publish. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [Azione «pubblicazione Amazon S3» YAML](#)
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

Azione «pubblicazione Amazon S3» YAML

Di seguito è riportata la definizione YAML dell'azione di pubblicazione di Amazon S3. Per informazioni su come utilizzare questa azione, consulta [Pubblicazione di file su Amazon S3 con un flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
S3Publish\_nn:
  Identifier: aws/s3-publish@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
  Timeout: timeout-minutes
  Inputs:
    Sources:
      - source-name-1
    Artifacts:
      - artifact-name
    Variables:
      - Name: variable-name-1
        Value: variable-value-1
```

```
- Name: variable-name-2
  Value: variable-value-2
Environment:
  Name: environment-name
Connections:
  - Name: account-connection-name
    Role: iam-role-name
Configuration:
  SourcePath: my/source
  DestinationBucketName: amzn-s3-demo-bucket
  TargetPath: my/target
```

S3Publish

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: S3Publish_nn.

Interfaccia utente corrispondente: scheda di configurazione/nome dell'azione

Identifier

(*S3Publish*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: aws/s3-publish@v1.

Interfaccia utente corrispondente: Workflow diagram/ S3Publish _nn/ aws/s3-publish @v1 label

DependsOn

(*S3Publish*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente affinché questa azione venga eseguita.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Input/Dipende da - opzionale

Compute

(*S3Publish*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*S3Publish*/Compute/Type)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: scheda di configurazione/tipo di calcolo

Fleet

(*S3Publish*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand: `Linux.x86-64.Large` `Linux.x86-64.XLarge` Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere. [Proprietà del parco veicoli assegnate](#)

Se `Fleet` viene omesso, l'impostazione predefinita è `Linux.x86-64.Large`

Interfaccia utente corrispondente: scheda di configurazione/parco di calcolo

Timeout

(*S3Publish*/Timeout)

(Obbligatorio)

Specificate il periodo di tempo, in minuti (editor YAML) o ore e minuti (editor visivo), che l'azione può essere eseguita prima CodeCatalyst che termini l'azione. Il minimo è 5 minuti e il massimo è descritto in. [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Inputs

(*S3Publish*/Inputs)

(Facoltativo)

La `Inputs` sezione definisce i dati `S3Publish` necessari durante l'esecuzione di un flusso di lavoro.

Note

Sono consentiti un massimo di quattro input (una fonte e tre artefatti) per ogni AWS CDK azione di distribuzione. Le variabili non vengono conteggiate per questo totale.

Se è necessario fare riferimento a file che risiedono in input diversi (ad esempio una sorgente e un artefatto), l'input di origine è l'input principale e l'artefatto è l'input secondario. I riferimenti ai file negli input secondari richiedono un prefisso speciale per distinguerli dal primario. Per informazioni dettagliate, vedi [Esempio: riferimento a file in più artefatti](#).

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*S3Publish*/Inputs/Sources)

(Obbligatorio se i file che desideri pubblicare su Amazon S3 sono archiviati in un repository di origine)

Se i file che desideri pubblicare su Amazon S3 sono archiviati in un repository di origine, specifica l'etichetta di tale repository di origine. Attualmente, l'unica etichetta supportata è `WorkflowSource`.

Se i file che desideri pubblicare su Amazon S3 non sono contenuti in un repository di origine, devono risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*S3Publish*/Inputs/Artifacts)

(Obbligatorio se i file che desideri pubblicare su Amazon S3 sono archiviati in un elemento [di output di un'azione precedente](#))

Se i file che desideri pubblicare su Amazon S3 sono contenuti in un artefatto generato da un'azione precedente, specifica tale elemento qui. Se i tuoi file non sono contenuti in un artefatto, devono risiedere nel tuo repository di origine.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda di configurazione/artefatti - opzionale

Variables - input

(*S3Publish*/Inputs/Variables)

(Facoltativo)

Specificate una sequenza di name/value coppie che definisca le variabili di input che desiderate rendere disponibili all'azione. I nomi delle variabili sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle variabili.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Variabili - opzionale

Environment

(*S3Publish*/Environment)

(Obbligatorio)

Specificare l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs](#) e [Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*S3Publish*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*S3Publish*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' / connessione all'account Environment/account/role AWS

Name

(*S3Publish*/Environment/Connections/Name)

(Richiesto se è incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / connessione all'account Environment/account/role AWS

Role

(*S3Publish*/Environment/Connections/Role)

(Richiesto se è incluso [Connections](#))

Specificare il nome del ruolo IAM utilizzato dall'azione di pubblicazione di Amazon S3 per accedere AWS e copiare i file su Amazon S3. Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

- La seguente politica di autorizzazioni:

Warning

Limita le autorizzazioni a quelle mostrate nella seguente politica. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject"
      ]
    }
  ],
```



```

        "Resource": [
            "arn:aws:s3:::bucket-name",
            "arn:aws:s3:::bucket-name/*"
        ]
    }
}

```

- La seguente politica di fiducia personalizzata:

Note

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' / Ruolo Environment/account/role

Configuration

(*S3Publish*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

SourcePath

(*S3Publish*/Configuration/SourcePath)

(Obbligatorio)

Specificare il nome e il percorso di una directory o di un file che si desidera pubblicare su Amazon S3. La directory o il file possono risiedere in un repository di origine o in un elemento di un'azione precedente ed è relativo al repository di origine o alla radice dell'artefatto.

Esempi:

Specificando `./myFolder/` copia il contenuto di su Amazon S3 e `/myFolder` preserva la struttura di directory sottostante.

Specificare `./myFolder/myfile.txt` copie solo `myfile.txt` su Amazon S3. (La struttura delle cartelle viene rimossa).

Non è possibile utilizzare i caratteri jolly.

Note

Potrebbe essere necessario aggiungere un prefisso alla directory o al percorso del file per indicare in quale elemento o fonte trovarlo. Per ulteriori informazioni, consultare [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

Interfaccia utente corrispondente: scheda di configurazione/percorso di origine

DestinationBucketName

(*S3Publish*/Configuration/DestinationBucketName)

(Obbligatorio)

Specificate il nome del bucket Amazon S3 in cui desiderate pubblicare i file.

Interfaccia utente corrispondente: scheda di configurazione/bucket di destinazione - opzionale

TargetPath

(*S3Publish*/Configuration/TargetPath)

(Facoltativo)

Specificate il nome e il percorso della directory in Amazon S3 in cui desiderate pubblicare i file. Se la directory non esiste, verrà creata. Il percorso della directory non deve includere il nome del bucket.

Esempi:

```
myS3Folder
```

```
./myS3Folder/myS3Subfolder
```

Interfaccia utente corrispondente: scheda di configurazione/directory di destinazione - opzionale

Implementazione in e Account AWS VPCs

Utilizzando [CodeCatalyst i flussi](#) di lavoro, puoi distribuire applicazioni e altre risorse per targetizzare Account AWS s e Amazon VPCs nel AWS cloud. Per abilitare queste implementazioni, devi configurare ambienti. CodeCatalyst

Un CodeCatalyst ambiente, da non confondere con un [ambiente di sviluppo](#), definisce l'Amazon VPC di destinazione Account AWS e opzionale a cui si connette un CodeCatalyst [flusso di lavoro](#). Un ambiente definisce anche il [ruolo IAM](#) necessario a un flusso di lavoro per accedere ai AWS servizi e alle risorse all'interno dell'account di destinazione.

È possibile configurare più ambienti e assegnare loro nomi come sviluppo, test, staging e produzione. Quando si esegue la distribuzione in questi ambienti, le informazioni sulle distribuzioni vengono visualizzate nelle schede CodeCatalyst Deployment activity e Deployment targets dell'ambiente.

Come posso iniziare a usare gli ambienti?

I passaggi di alto livello per aggiungere e utilizzare un CodeCatalyst ambiente sono i seguenti:

1. Nel tuo CodeCatalyst spazio, collega uno o più AWS account. Durante questo processo, aggiungi i ruoli IAM necessari al tuo flusso di lavoro per accedere alle risorse del tuo Account AWS. Per ulteriori informazioni, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS](#).
2. Nel tuo CodeCatalyst progetto, crea un ambiente che includa uno dei ruoli Account AWS s e IAM della fase 1. Per ulteriori informazioni, consulta [Creazione di un ambiente](#).
3. Nel tuo CodeCatalyst progetto, in un flusso di lavoro, aggiungi un'[azione](#) che rimanda all'ambiente che hai creato nel passaggio 2. Per ulteriori informazioni, consulta [Aggiungere un'azione a un flusso di lavoro](#).

Ora hai configurato un ambiente. L'azione può ora distribuire risorse nell'ambiente Account AWS specificato.

 Note

Puoi anche aggiungere un Amazon VPC all'ambiente. Per ulteriori informazioni, vedere [Aggiungere connessioni VPC per uno spazio](#) nella Guida all'CodeCatalyst amministrazione e [Associazione di un VPC a un ambiente](#)

Possono esistere più ambienti all'interno di un unico flusso di lavoro?

Sì. Se un flusso di lavoro include più azioni, a ciascuna di esse può essere assegnato un ambiente. Ad esempio, è possibile disporre di un flusso di lavoro che include due azioni di distribuzione, in cui a una viene assegnato un my-staging-environment ambiente e all'altra viene assegnato un my-production-environment ambiente.

Quali azioni del flusso di lavoro supportano gli ambienti?

Qualsiasi azione del flusso di lavoro che distribuisce risorse nel AWS cloud o comunica con i AWS servizi per altri motivi (come il monitoraggio e il reporting), supporta gli ambienti.

Quali azioni supportano la visualizzazione delle informazioni sulla distribuzione in? CodeCatalyst

Tra le azioni del flusso di lavoro che supportano gli ambienti, solo alcune supportano la visualizzazione delle informazioni sulla distribuzione nelle pagine Attività di distribuzione e Target di distribuzione della CodeCatalyst console.

Le seguenti azioni del flusso di lavoro supportano la visualizzazione delle relative informazioni di distribuzione:

- Deploy CloudFormation stack: per ulteriori informazioni, vedere [Implementazione di uno stack CloudFormation](#)
- Implementazione su Amazon ECS: per ulteriori informazioni, consulta [Distribuzione su Amazon ECS con un flusso di lavoro](#)
- Esegui la distribuzione su un cluster Kubernetes: per ulteriori informazioni, consulta [Implementazione su Amazon EKS con un flusso di lavoro](#)
- AWS CDK deploy: per ulteriori informazioni, consulta [Implementazione di un' AWS CDK app con un flusso di lavoro](#)

Regioni supportate

La pagina Ambienti può visualizzare risorse in qualsiasi AWS regione.

L'ambiente è obbligatorio?

Un ambiente è obbligatorio se l'azione del flusso di lavoro a cui è assegnato distribuisce risorse nel AWS cloud o comunica con i AWS servizi per altri motivi (come il monitoraggio e il reporting).

Ad esempio, se hai un'azione di compilazione che crea un'applicazione ma non ha bisogno di comunicare con il tuo Account AWS o con Amazon VPC, non è necessario assegnare un ambiente all'azione. Se, tuttavia, l'azione di compilazione invia i log al CloudWatch servizio Amazon del tuo Account AWS, all'azione deve essere assegnato un ambiente.

Argomenti

- [Creazione di un ambiente](#)
- [Associare un ambiente a un'azione](#)
- [Associazione di un VPC a un ambiente](#)
- [Associare un a Account AWS un ambiente](#)
- [Modifica del ruolo IAM di un'azione](#)

Creazione di un ambiente

Utilizzate le seguenti istruzioni per creare un ambiente da associare successivamente a un'azione del flusso di lavoro.

Prima di iniziare

È necessario quanto segue:

- Uno CodeCatalyst spazio. Per ulteriori informazioni, consulta [Configura e accedi a CodeCatalyst](#).
- Un CodeCatalyst progetto. Per ulteriori informazioni, consulta [Creare un progetto con un blueprint](#).
- Una connessione all' AWS account che include i ruoli IAM a cui dovrà accedere l'azione del workflow AWS. Per informazioni sulla creazione di una connessione all'account, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS](#). È possibile utilizzare al massimo una connessione a un account per ambiente.

 Note

È possibile creare un ambiente senza una connessione all'account; tuttavia, sarà necessario tornare indietro e aggiungere la connessione in un secondo momento.

- Uno dei seguenti CodeCatalyst ruoli:
 - Amministratore dello spazio
 - Amministratore del progetto
 - Collaboratore

 Note

Se hai il ruolo di Collaboratore, sarai in grado di creare un ambiente ma non potrai associarlo a una Account AWS connessione. Dovrai chiedere a qualcuno con il ruolo di amministratore dello spazio o di amministratore del progetto di associare l'ambiente a una Account AWS connessione.

Per ulteriori informazioni su autorizzazioni e ruoli, consulta [Concessione delle autorizzazioni di progetto agli utenti](#).

Come creare un ambiente

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
4. In Nome ambiente, inserisci un nome, ad esempio **Production** o **Staging**.
5. In Tipo di ambiente, selezionate una delle seguenti opzioni:
 - Non di produzione: un ambiente in cui è possibile testare l'applicazione per assicurarsi che funzioni come previsto prima di passare alla produzione.
 - Produzione: un ambiente «live» disponibile al pubblico che ospita l'applicazione definitiva.

Se scegli Produzione, nell'interfaccia utente viene visualizzato un badge di produzione accanto a tutte le azioni a cui è associato l'ambiente. Il badge ti aiuta a vedere rapidamente quali azioni

vengono implementate nella produzione. Oltre all'aspetto del badge, non ci sono differenze tra ambienti di produzione e non di produzione.

6. (Facoltativo) In Descrizione, inserisci una descrizione come **Production environment for the hello-world app**
7. In Account AWS Connessione: facoltativo, scegli la connessione AWS dell'account che desideri associare a questo ambiente. Le azioni del flusso di lavoro assegnate a questo ambiente potranno connettersi a quello associato Account AWS. Per ulteriori informazioni sulla creazione di Account AWS connessioni in CodeCatalyst, vedere [Consentire l'accesso alle AWS risorse con connessione Account AWS](#).

Se la Account AWS connessione che desideri utilizzare non è elencata, è possibile che non sia consentita nel progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

8. Nel ruolo IAM predefinito, scegli il ruolo IAM che desideri associare a questo ambiente. Le azioni del flusso di lavoro assegnate a questo ambiente ereditano questo ruolo IAM e potranno utilizzarlo per connettersi ai servizi e alle risorse del tuo Account AWS.

Se devi assegnare l'ambiente a più azioni e tali azioni richiedono ruoli IAM diversi da quello predefinito qui specificato, puoi specificare i diversi ruoli nella scheda Configurazione di ciascuna azione, utilizzando l'opzione Switch role. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Se il ruolo IAM che desideri utilizzare come predefinito non è elencato, potrebbe essere perché non lo hai ancora aggiunto alla tua Account AWS connessione. Per aggiungere un ruolo IAM alla connessione di un account, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

9. (Facoltativo) Nella connessione VPC, scegli una connessione VPC che desideri associare a questo ambiente. Per ulteriori informazioni sulla creazione di connessioni VPC, consulta Managing [Amazon Virtual Private Clouds](#) nella Amazon CodeCatalyst Administrator Guide.

Se la connessione VPC che desideri utilizzare non è elencata, potrebbe essere perché include una Account AWS connessione non consentita nel tuo progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

10. Scegli Crea ambiente. CodeCatalyst crea un ambiente vuoto.

Fasi successive

- Ora che hai creato un ambiente, sei pronto per associarlo a un'azione del flusso di lavoro. Per ulteriori informazioni, consulta [Associare un ambiente a un'azione](#).

Associare un ambiente a un'azione

Quando associ un ambiente a un'[azione di workflow supportata](#), all'azione vengono assegnati l'ambiente, il ruolo IAM predefinito e Amazon VPC opzionale. Account AWS L'azione può quindi connettersi e implementare al ruolo Account AWS Using the IAM e connettersi anche all'Amazon VPC opzionale.

Utilizza le seguenti istruzioni per associare un ambiente a un'azione.

Fase 1: Associare l'ambiente a un'azione del flusso di lavoro

Utilizzare la procedura seguente per associare un ambiente a un'azione del flusso di lavoro.

Visual

Per associare un ambiente a un'azione del flusso di lavoro utilizzando l'editor visivo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli un'azione supportata dagli ambienti. Per ulteriori informazioni, consulta [Quali azioni supportano la visualizzazione delle informazioni sulla distribuzione in? CodeCatalyst](#).
8. Scegliete la scheda Configurazione e specificate le informazioni nel campo Ambiente, come segue.

Ambiente

Specificate l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

 Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs e Creazione di un ambiente](#).

9. (Facoltativo) Modifica il ruolo IAM associato all'azione. Potresti voler cambiare il ruolo se contiene il set di autorizzazioni errato per l'azione.

Per cambiare il ruolo:

1. Nella sezione Cosa c'è dentro **my-environment**? casella e scegliete l'icona con i puntini di sospensione verticali ()
2. Seleziona una delle seguenti opzioni:
 - Cambia ruolo. Scegli questa opzione per modificare il ruolo IAM utilizzato da questa azione e solo da questa azione. Altre azioni continuano a utilizzare il ruolo IAM predefinito specificato nell'ambiente associato. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).
 - Modifica ambiente. Scegli questa opzione per modificare il ruolo IAM predefinito elencato nel tuo ambiente. Quando scegli questa opzione, la tua azione, e qualsiasi altra azione associata allo stesso ambiente, inizia a utilizzare il nuovo ruolo IAM predefinito.

 Important

Fai attenzione quando aggiorni il ruolo IAM predefinito. La modifica del ruolo potrebbe causare errori nelle azioni se le autorizzazioni relative al ruolo non sono sufficienti per tutte le azioni che condividono l'ambiente.

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per associare un ambiente a un'azione del flusso di lavoro utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell'azione del flusso di lavoro che desideri associare a un ambiente, aggiungi un codice simile al seguente:

```
action-name:
  Environment:
    Name: environment-name
```

Per ulteriori informazioni, consulta l'[Tipi di operazione](#) argomento. Questo argomento contiene collegamenti alla documentazione per ogni azione, incluso il relativo riferimento YAML.

8. (Facoltativo) Se desideri che l'azione utilizzi un ruolo diverso dal ruolo IAM predefinito elencato nell'ambiente, aggiungi una `Connections:` sezione che includa il ruolo che desideri utilizzare. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).
9. (Facoltativo) Scegli Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Fase 2: Compila la pagina delle attività di distribuzione

Dopo aver associato un ambiente a un'azione del flusso di lavoro, è possibile compilare le pagine Attività di distribuzione e Target di distribuzione nella sezione Ambienti della CodeCatalyst console con informazioni sulla distribuzione. Utilizza le seguenti istruzioni per compilare queste pagine.

Note

Solo alcune azioni supportano la visualizzazione delle informazioni sulla distribuzione nella CodeCatalyst console. Per ulteriori informazioni, consulta [Quali azioni supportano la visualizzazione delle informazioni sulla distribuzione in? CodeCatalyst](#).

Per aggiungere informazioni sulla distribuzione a CodeCatalyst

1. Se l'esecuzione di un flusso di lavoro non si avvia automaticamente al momento del commit delle modifiche [Fase 1: Associare l'ambiente a un'azione del flusso di lavoro](#), avvia manualmente un'esecuzione come segue:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - b. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
 - c. Scegli Esegui.

L'esecuzione del flusso di lavoro avvia una nuova distribuzione, che comporta CodeCatalyst l'aggiunta di informazioni sulla distribuzione. CodeCatalyst

2. Verifica che l'attività di distribuzione sia stata aggiunta alla CodeCatalyst console:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
 - b. Scegli il tuo ambiente (ad esempio, `Production`).
 - c. Scegli la scheda Attività di distribuzione e verifica che venga visualizzata una distribuzione con lo stato RIUSCITA. Ciò indica che un flusso di lavoro eseguito ha distribuito correttamente le risorse dell'applicazione.
 - d. Scegli la scheda Obiettivi di distribuzione e verifica che vengano visualizzate le risorse dell'applicazione.

Associazione di un VPC a un ambiente

Quando un'azione è configurata con un ambiente che dispone di una connessione VPC, l'azione verrà eseguita connessa al VPC, rispettando le regole di rete e le risorse di accesso specificate dal VPC associato. La stessa connessione VPC può essere utilizzata da uno o più ambienti.

Utilizza le seguenti istruzioni per associare una connessione VPC a un ambiente.

Per associare una connessione VPC a un ambiente

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
4. Scegliete il vostro ambiente (ad esempio, `Production`).
5. Scegli la scheda delle proprietà dell'ambiente.
6. Scegli Gestisci connessione VPC, scegli la connessione VPC desiderata e scegli Conferma. Ciò associa la connessione VPC selezionata a questo ambiente.

Note

Se la connessione VPC che desideri utilizzare non è elencata, potrebbe essere perché include una Account AWS connessione non consentita nel tuo progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

Per ulteriori informazioni, consulta [Managing Amazon Virtual Private Clouds](#) nella CodeCatalyst Administrator Guide.

Associare un Account AWS a un ambiente

Usa le seguenti istruzioni per Account AWS associare un uomo a un ambiente. Quando si associa un Account AWS ambiente, le azioni del flusso di lavoro assegnate all'ambiente potranno connettersi a Account AWS.

Per ulteriori informazioni sulle connessioni degli account, vedere [Consentire l'accesso alle AWS risorse con connessione Account AWS](#).

Prima di iniziare

È necessario quanto segue:

- Una connessione AWS all'account che includa i ruoli IAM a cui dovrà accedere l'azione del flusso di lavoro AWS. Per informazioni sulla creazione di una connessione all'account, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS](#). È possibile utilizzare al massimo una connessione a un account per ambiente.
- Uno dei seguenti CodeCatalyst ruoli: amministratore dello spazio o amministratore del progetto. Per ulteriori informazioni, consulta [Concessione delle autorizzazioni di progetto agli utenti](#).

Associare un uomo Account AWS a un ambiente

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
4. Scegliete il vostro ambiente (ad esempio, Production).
5. Scegli Modifica ambiente.
6. In Proprietà dell'ambiente, nell'elenco a discesa Account AWS Connessione - opzionale, scegli quello che desideri. Account AWS

Se la Account AWS connessione che desideri utilizzare non è elencata, è possibile che non sia consentita nel progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#) nella Amazon CodeCatalyst Administrator Guide.

7. Nel ruolo IAM predefinito, scegli il ruolo IAM che desideri associare a questo ambiente. Le azioni del flusso di lavoro assegnate a questo ambiente ereditano questo ruolo IAM e potranno utilizzarlo per connettersi ai servizi e alle risorse del tuo Account AWS.

Se il ruolo IAM che desideri utilizzare come predefinito non è elencato, potrebbe essere perché non lo hai ancora aggiunto alla tua Account AWS connessione. Per aggiungere un ruolo IAM alla connessione di un account, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

Modifica del ruolo IAM di un'azione

Per impostazione predefinita, quando si associa un [ambiente](#) a un'[azione](#) del flusso di lavoro, l'azione eredita il ruolo IAM predefinito specificato nell'ambiente. È possibile modificare questo

comportamento in modo che l'azione utilizzi un ruolo diverso. Potresti volere che un'azione utilizzi un ruolo diverso se al ruolo IAM predefinito mancano le autorizzazioni necessarie all'azione per operare nel AWS cloud.

Per assegnare un ruolo IAM diverso a un'azione, puoi utilizzare l'opzione Switch role nell'editor visuale o la `Connections:` proprietà nell'editor YAML. Il nuovo ruolo sostituisce il ruolo IAM predefinito specificato nell'ambiente, consentendoti di mantenere il ruolo IAM predefinito così com'è. Potresti voler mantenere il ruolo IAM predefinito così com'è se ci sono altre azioni che lo utilizzano.

Utilizza le seguenti istruzioni per configurare un'azione per utilizzare un ruolo IAM diverso da quello specificato nel relativo ambiente.

Visual

Per assegnare un ruolo IAM diverso a un'azione (editor visivo)

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli la casella che rappresenta l'azione di cui desideri aggiornare il ruolo IAM.
7. Scegli la scheda Configurazione.
8. Nella sezione Cosa c'è dentro **my-environment**? casella, scegliete l'icona con i puntini di sospensione verticali
().
9. Scegli Cambia ruolo.
10. Nella finestra di dialogo Cambia ruolo, nell'elenco a discesa del ruolo IAM, scegli il ruolo IAM che desideri venga utilizzato dall'azione. Questo ruolo sostituirà il ruolo IAM predefinito nell'ambiente. Se il ruolo che desideri utilizzare non è nell'elenco, assicurati di averlo aggiunto al tuo spazio. Per ulteriori informazioni, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

Il ruolo scelto viene ora visualizzato nella sezione Cosa c'è in **my-environment**? riquadro con il badge Definito nel flusso di lavoro. Il ruolo appare anche nel file di definizione del flusso di lavoro, nella `Connections`: sezione.

11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per assegnare un ruolo IAM diverso a un'azione (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell'azione del flusso di lavoro in cui desideri utilizzare un ruolo IAM diverso, aggiungi una `Connections`: sezione simile alla seguente:

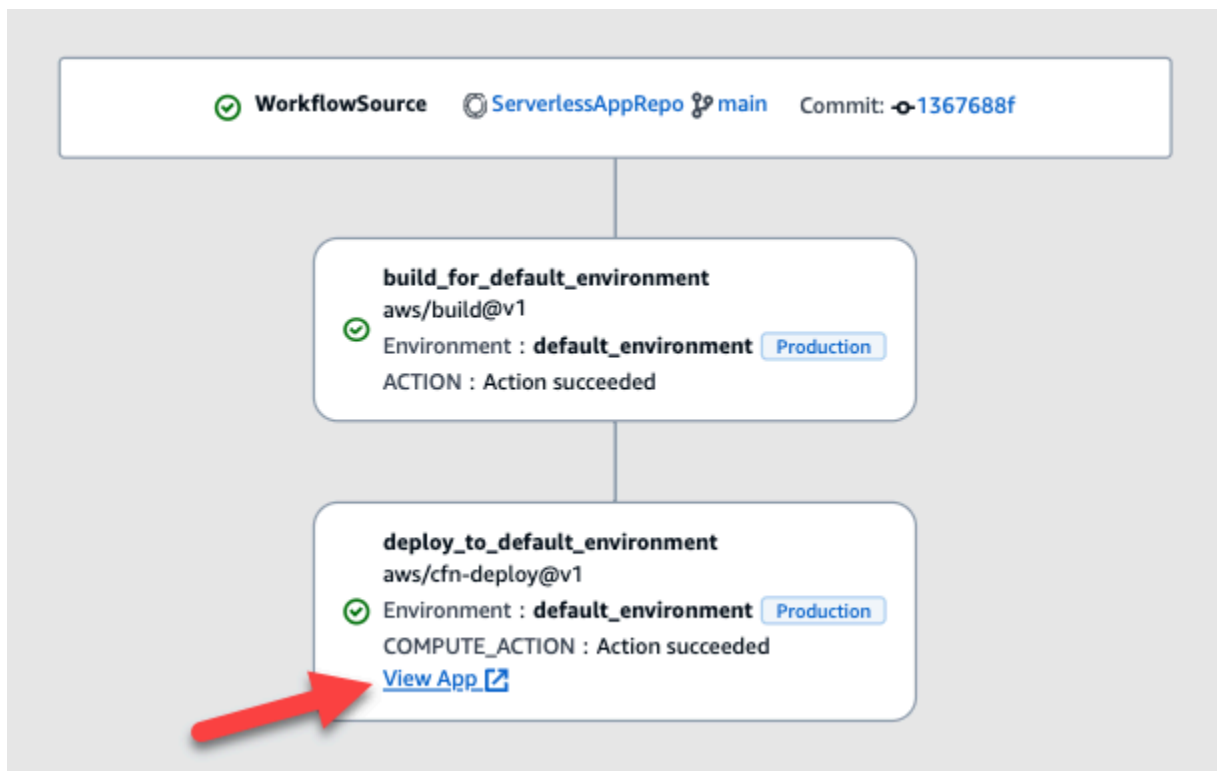
```
action-name:
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
```

Nel codice precedente, sostituiscilo **account-connection-name** con il nome della [connessione all'account](#) che contiene il ruolo IAM e **iam-role-name** sostituiscilo con il nome del ruolo IAM che desideri che l'azione utilizzi. Questo ruolo sostituirà il ruolo IAM predefinito nell'ambiente. Assicurati di aver aggiunto il ruolo al tuo spazio. Per ulteriori informazioni, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

Per ulteriori informazioni, consulta l'[Tipi di operazione](#) argomento. Questo argomento contiene collegamenti alla documentazione per ogni azione, incluso il relativo riferimento YAML.

Visualizzazione dell'URL dell'app nel diagramma del flusso di lavoro

Se il tuo flusso di lavoro implementa un'applicazione, puoi configurare Amazon CodeCatalyst per visualizzare l'URL dell'applicazione come link cliccabile. Questo link appare nella CodeCatalyst console, all'interno dell'azione che lo ha distribuito. Il seguente diagramma del flusso di lavoro mostra l'URL dell'app View che appare nella parte inferiore di un'azione.



Rendendo cliccabile questo URL nella CodeCatalyst console, puoi verificare rapidamente la distribuzione dell'applicazione.

Note

L'URL dell'app non è supportato con l'azione Deploy to Amazon ECS.

Per abilitare questa funzionalità, aggiungi una variabile di output all'azione con un nome che contenga `appurl` o `endpointurl`. Puoi usare un nome con o senza un trattino (-), un

trattino basso (`_`) o uno spazio (). La stringa non fa distinzione tra maiuscole e minuscole. Imposta il valore della variabile `https` sull'URL `http` o dell'applicazione distribuita.

Note

Se stai aggiornando una variabile di output esistente per includere la endpoint `url` stringa `oapp_url`, aggiorna tutti i riferimenti a questa variabile per utilizzare il nuovo nome della variabile.

Per i passaggi dettagliati, consulta una delle seguenti procedure:

- [Per visualizzare l'URL dell'app nell'azione "AWS CDK deploy»](#)
- [Per visualizzare l'URL dell'app nell'azione «Deploy CloudFormation stack»](#)
- [Per visualizzare l'URL dell'app in tutte le altre azioni](#)

Al termine della configurazione dell'URL, verifica che appaia come previsto seguendo queste istruzioni:

- [Per verificare che l'URL dell'applicazione sia stato aggiunto](#)

Per visualizzare l'URL dell'app nell'azione "AWS CDK deploy»

1. Se utilizzi l'azione AWS CDK deploy, aggiungi un `CfnOutput` costruito (che è una coppia chiave-valore) nel codice dell'applicazione: AWS CDK
 - Il nome della chiave deve contenere `appurl`, `oendpointurl`, con o senza un trattino (`_`), un trattino di sottolineatura (`-`) o uno spazio (). La stringa non fa distinzione tra maiuscole e minuscole.
 - Il valore deve essere l'URL `https` o l'URL dell'applicazione distribuita.

Ad esempio, il AWS CDK codice potrebbe avere il seguente aspetto:

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
```

```
export class HelloCdkStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);
    const bucket = new s3.Bucket(this, 'amzn-s3-demo-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'APP-URL', {
      value: https://mycompany.myapp.com,
      description: 'The URL of the deployed application',
      exportName: 'myApp',
    });
    ...
  }
}
```

Per ulteriori informazioni sul CfnOutput costruito, consulta l'[interfaccia CfnOutputProps](#) nell'AWS Cloud Development Kit (AWS CDK) API Reference.

2. Salva e conferma il codice.
3. Passa a [Per verificare che l'URL dell'applicazione sia stato aggiunto](#).

Per visualizzare l'URL dell'app nell'azione «Deploy CloudFormation stack»

1. Se utilizzi l'azione Deploy CloudFormation stack, aggiungi un output alla Outputs sezione del tuo CloudFormation modello o AWS SAM modello con queste caratteristiche:
 - La chiave (chiamata anche ID logico) deve contenere `appurl`, con o senza, un trattino di unione (`endpointurl`), un trattino di sottolineatura (`-`) o uno `_` spazio (). La stringa non fa distinzione tra maiuscole e minuscole.
 - Il valore deve essere l'HTTPS URL http o l'URL dell'applicazione distribuita.

Ad esempio, il CloudFormation modello potrebbe avere il seguente aspetto:

```
"Outputs" : {
  "APP-URL" : {
    "Description" : "The URL of the deployed app",
    "Value" : "https://mycompany.myapp.com",
    "Export" : {
      "Name" : "My App"
    }
  }
}
```

```
}  
}
```

Per ulteriori informazioni sugli CloudFormation output, consulta [Output nella Guida](#) per l'AWS CloudFormation utente.

2. Salva e conferma il codice.
3. Passa a [Per verificare che l'URL dell'applicazione sia stato aggiunto](#).

Per visualizzare l'URL dell'app in tutte le altre azioni

Se stai utilizzando un'altra azione per distribuire l'applicazione, ad esempio l'azione di compilazione o GitHub le azioni, procedi come segue per visualizzare l'URL dell'app.

1. Definite una variabile di ambiente nella Steps sezione Inputs o nell'azione nel file di definizione del flusso di lavoro. La variabile deve avere le seguenti caratteristiche:
 - nameDeve contenereappurl, oendpointurl, con o senza un trattino (-), un carattere di sottolineatura (_) o uno spazio (). La stringa non fa distinzione tra maiuscole e minuscole.
 - Il valore deve essere l'httpsURL http o l'URL dell'applicazione distribuita.

Ad esempio, un'azione di compilazione potrebbe avere il seguente aspetto:

```
Build-action:  
  Identifier: aws/build@v1  
  Inputs:  
    Variables:  
      - Name: APP-URL  
        Value: https://mycompany.myapp.com
```

... o questo:

```
Actions:  
  Build:  
    Identifier: aws/build@v1  
    Configuration:  
      Steps:  
        - Run: APP-URL=https://mycompany.myapp.com
```

Per ulteriori informazioni sulla definizione delle variabili di ambiente, vedere [Definizione di una variabile](#).

2. Esporta la variabile.

Ad esempio, l'azione di compilazione potrebbe essere simile alla seguente:

```
Build-action:
...
Outputs:
  Variables:
    - APP-URL
```

Per informazioni sull'esportazione delle variabili, consulta [Esportazione di una variabile in modo che altre azioni possano utilizzarla](#).

3. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
4. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.
5. Passa a [Per verificare che l'URL dell'applicazione sia stato aggiunto](#).

Per verificare che l'URL dell'applicazione sia stato aggiunto

- Avvia l'esecuzione di un flusso di lavoro, se non è stato avviato automaticamente. La nuova esecuzione dovrebbe avere l'URL dell'app visualizzato come collegamento cliccabile nel diagramma del flusso di lavoro. Per ulteriori informazioni sull'avvio delle esecuzioni, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#)

Rimozione di un obiettivo di distribuzione

Puoi rimuovere un obiettivo di distribuzione, ad esempio un cluster o uno CloudFormation stack Amazon ECS, dalla pagina Destinazioni di distribuzione nella CodeCatalyst console.

Important

Quando rimuovi un obiettivo di distribuzione, questo viene rimosso dalla CodeCatalyst console, ma rimane disponibile nel AWS servizio che lo ospita (se esiste ancora).

Prendi in considerazione la possibilità di rimuovere un obiettivo di distribuzione se il target è diventato obsoleto in CodeCatalyst. Gli obiettivi potrebbero diventare obsoleti se:

- Hai eliminato il flusso di lavoro distribuito sulla destinazione.
- Hai cambiato lo stack o il cluster in cui esegui la distribuzione.
- Hai eliminato lo stack o il cluster dal CloudFormation servizio Amazon ECS nella AWS console.

Per rimuovere un obiettivo di distribuzione

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
4. Scegli il nome dell'ambiente che contiene l'obiettivo di distribuzione che desideri rimuovere. Per informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs](#).
5. Scegli la scheda Obiettivi di distribuzione.
6. Scegli il pulsante di opzione accanto al target di distribuzione che desideri rimuovere.
7. Scegli Rimuovi.

L'obiettivo viene rimosso dalla pagina.

Monitoraggio dello stato di distribuzione tramite commit

In qualsiasi momento del ciclo di vita dello sviluppo, è importante conoscere lo stato di implementazione di commit specifici, come correzioni di bug, nuove funzionalità o altre modifiche importanti. Considerate i seguenti scenari in cui la funzionalità di monitoraggio dello stato dell'implementazione è utile per i team di sviluppo:

- In qualità di sviluppatore, hai apportato una correzione per correggere un bug e desideri segnalare lo stato della sua implementazione negli ambienti di distribuzione del tuo team.
- In qualità di release manager, desideri visualizzare un elenco di commit distribuiti per monitorare e segnalare lo stato della distribuzione.

CodeCatalyst fornisce una visualizzazione che è possibile utilizzare per determinare a colpo d'occhio dove sono stati implementati i singoli commit o modifiche e in quale ambiente. Questa visualizzazione include:

- Un elenco di commit.
- Lo stato delle distribuzioni che includono i commit.
- Gli ambienti in cui i commit vengono distribuiti con successo.
- Lo stato di tutti i test eseguiti rispetto ai commit nel flusso di lavoro. CI/CD

La procedura seguente descrive in dettaglio come accedere a questa visualizzazione e utilizzarla per tenere traccia delle modifiche nel progetto.

Note

Il monitoraggio dello stato di distribuzione tramite commit è supportato solo con i [CodeCatalyst repository](#). Non è possibile utilizzare questa funzionalità con un [GitHub repository](#), un [repository Bitbucket](#) o un [repository](#) di progetto. GitLab

Per tenere traccia dello stato della distribuzione tramite commit

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Change tracking.
4. Nei due elenchi a discesa nella parte superiore del riquadro principale, scegli il repository di origine e il ramo che contengono i commit di cui desideri visualizzare lo stato di rilascio.
5. Scegli Visualizza modifiche.

Viene visualizzato un elenco di commit.

Per ogni commit, puoi visualizzare quanto segue:

- Conferma informazioni come ID, autore, messaggio e quando è stato eseguito il commit. Per ulteriori informazioni, consulta [Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst](#).
- Lo stato delle implementazioni in ogni ambiente. Per ulteriori informazioni, consulta [Implementazione in e Account AWS VPCs](#).
- Risultati dei test e della copertura del codice. Per ulteriori informazioni, consulta [Test con flussi di lavoro](#).

 Note

I risultati dell'analisi della composizione del software (SCA) non vengono visualizzati.

6. (Facoltativo) Per visualizzare ulteriori informazioni sulle modifiche relative a un commit specifico, tra cui la distribuzione più recente e la copertura dettagliata del codice e le informazioni sui test unitari, scegli Visualizza i dettagli per quel commit.

Visualizzazione dei log di distribuzione

Puoi visualizzare i log relativi a specifiche azioni di distribuzione per risolvere i problemi in Amazon CodeCatalyst

[Puoi visualizzare i log a partire da un flusso di lavoro o da un ambiente.](#)

Per visualizzare i registri di un'azione di distribuzione a partire da un flusso di lavoro

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Esecuzioni.
6. Scegli il flusso di lavoro eseguito che ha distribuito la tua applicazione.
7. Nel diagramma del flusso di lavoro, scegli l'azione di cui desideri visualizzare i log.
8. Scegli la scheda Registri ed espandi le sezioni per visualizzare i messaggi di registro.
9. Per visualizzare altri registri, scegli la scheda Riepilogo, quindi scegli Visualizza in CloudFormation (se disponibile) per visualizzarne altri. Potrebbe essere necessario accedere a AWS

Per visualizzare i log di un'azione di distribuzione a partire da un ambiente

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Scegliere il progetto.

3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
4. Scegli l'ambiente in cui è stata distribuita l'applicazione.
5. In Attività di distribuzione, trova la colonna Workflow Run ID e scegli l'esecuzione del flusso di lavoro che ha distribuito il tuo stack.
6. Nel diagramma del flusso di lavoro, scegli l'azione di cui desideri visualizzare i log.
7. Scegli la scheda Registri ed espandi le sezioni per visualizzare i messaggi di registro.
8. Per visualizzare altri registri, scegli la scheda Riepilogo, quindi scegli Visualizza in CloudFormation (se disponibile) per visualizzarne altri. Potrebbe essere necessario accedere a AWS.

Visualizzazione delle informazioni sulla distribuzione

Puoi visualizzare le seguenti informazioni su una distribuzione in Amazon CodeCatalyst:

- Attività di distribuzione, inclusi lo stato della distribuzione, l'ora di inizio, l'ora di fine, la cronologia e la durata degli eventi.
- Nome dello stack Regione AWS, ora dell'ultimo aggiornamento e flussi di lavoro associati.
- Commit e richieste pull.
- Informazioni specifiche sull'azione, ad esempio CloudFormation eventi e risultati.

[È possibile visualizzare le informazioni sulla distribuzione a partire da un flusso di lavoro, un ambiente o un'azione del flusso di lavoro.](#)

Per visualizzare le informazioni sulla distribuzione a partire da un flusso di lavoro

- Vai all'esecuzione del flusso di lavoro che ha distribuito l'applicazione. Per istruzioni, consulta [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro.](#)

Per visualizzare le informazioni sulla distribuzione a partire da un ambiente

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Ambienti.
4. Scegli l'ambiente in cui è stato distribuito lo stack, ad esempio. Production

5. Scegli Attività di distribuzione per visualizzare la cronologia di implementazione dei tuoi stack, lo stato delle distribuzioni (ad esempio, RIUSCITED o FAILED) e altre informazioni relative alla distribuzione.
6. Scegli Deployment target per visualizzare le informazioni sugli stack, i cluster o altri obiettivi distribuiti nell'ambiente. È possibile visualizzare informazioni come il nome dello stack, la regione, il provider e l'identificatore.

Per visualizzare le informazioni sulla distribuzione a partire da un'azione

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Nel diagramma del flusso di lavoro, scegli l'azione del flusso di lavoro che ha distribuito l'applicazione. Ad esempio, puoi scegliere. DeployCloudFormationStack
6. Esamina il contenuto nel riquadro a destra per informazioni sulla distribuzione specifiche dell'azione.

Creazione di un flusso di lavoro

Un flusso di lavoro è una procedura automatizzata che descrive come creare, testare e distribuire il codice come parte di un sistema di integrazione e distribuzione continua (CI/CD). Un flusso di lavoro definisce una serie di passaggi, o azioni, da eseguire durante l'esecuzione di un flusso di lavoro. Un flusso di lavoro definisce anche gli eventi, o trigger, che causano l'avvio del flusso di lavoro. Per configurare un flusso di lavoro, è necessario creare un file di definizione del flusso di lavoro utilizzando l'editor [visivo o YAML](#) della CodeCatalyst console.

Tip

Per una rapida occhiata a come potresti utilizzare i flussi di lavoro in un progetto, [crea un progetto con un blueprint](#). Ogni blueprint implementa un flusso di lavoro funzionante che puoi rivedere, eseguire e sperimentare.

Utilizzare la procedura seguente per creare un flusso di lavoro in CodeCatalyst. Il flusso di lavoro verrà archiviato come file YAML in una `~/.codecatalyst/workflows/` cartella nel repository di origine scelto. Facoltativamente, è possibile archiviare il flusso di lavoro in una sottocartella o anteporre al nome del `~/.codecatalyst/workflows/` file del flusso di lavoro un nome di cartella al momento del commit. Per ulteriori informazioni, consulta le seguenti istruzioni.

Per ulteriori informazioni sui flussi di lavoro, consulta [Crea, testa e distribuisce con i flussi di lavoro](#).

Visual

Per creare un flusso di lavoro utilizzando l'editor visivo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli Crea flusso di lavoro.

Viene visualizzata la finestra di dialogo Crea flusso di lavoro.

5. Nel campo Archivio di origine, scegliete un archivio di origine in cui risiederà il file di definizione del flusso di lavoro. [Se non esiste alcun repository di origine, creane uno.](#)
6. Nel campo Branch, scegliete un ramo in cui risiederà il file di definizione del flusso di lavoro.
7. Scegli Create (Crea).

Amazon CodeCatalyst salva le informazioni del repository e della filiale in memoria, ma il flusso di lavoro non è ancora stato eseguito.

8. Scegli Visual.
9. Crea il flusso di lavoro:
 - a. (Facoltativo) Nel diagramma del flusso di lavoro, selezionate la casella Sorgente e trigger. Viene visualizzato il riquadro Trigger. Scegli Aggiungi trigger per aggiungere un trigger. Per ulteriori informazioni, consulta [Aggiungere trigger ai flussi di lavoro](#).
 - b. Scegli + Azioni (in alto a sinistra). Viene visualizzato il catalogo Actions.
 - c. Scegli il segno più (+) all'interno di un'azione per aggiungerla al flusso di lavoro. Usa il riquadro a destra per configurare l'azione. Per ulteriori informazioni, consulta [Aggiungere un'azione a un flusso di lavoro](#).
 - d. (Facoltativo) Scegli le proprietà del flusso di lavoro (in alto a destra). Viene visualizzato il riquadro delle proprietà del flusso di lavoro. Configura il nome del flusso di lavoro, la

modalità di esecuzione e il calcolo. Per ulteriori informazioni, consultare [Configurazione del comportamento di accodamento delle esecuzioni](#) e [Configurazione delle immagini di calcolo e di runtime](#).

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit e, nella finestra di dialogo Commit workflow, effettuate le seguenti operazioni:
 - a. Per il nome del file Workflow, lasciate il nome predefinito o immettetene uno personalizzato. Il file verrà archiviato in una `~/.codecatalyst/workflows/` cartella nel repository e nel ramo di origine scelti. È possibile anteporre al nome del file una cartella o sottocartella. Esempi:
 - Se si specifica `my-workflow` (nessuna cartella), il file viene memorizzato come `~/.codecatalyst/workflows/my-workflow.yaml`
 - Specificando `folder/subfolder/my-workflow` memorizza il file come `~/.codecatalyst/workflows/folder/subfolder/my-workflow.yaml`
 - b. Per il messaggio di conferma, lascia il messaggio predefinito o inserisci il tuo.
 - c. Per Repository e Branch, scegliete il repository e il branch di origine per il file di definizione del flusso di lavoro. Questi campi devono essere impostati sul repository e sul branch specificati in precedenza nella finestra di dialogo Crea flusso di lavoro. Ora puoi cambiare il repository e il ramo, se lo desideri.

 Note

Dopo aver salvato il file di definizione del flusso di lavoro, non può essere associato a un altro repository o ramo, quindi assicurati di sceglierli con attenzione.

- d. Scegliete Commit per eseguire il commit del file di definizione del flusso di lavoro.

YAML

Per creare un flusso di lavoro utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.

3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli Crea flusso di lavoro.

Viene visualizzata la finestra di dialogo Crea flusso di lavoro.

5. Nel campo Archivio di origine, scegliete un archivio di origine in cui risiederà il file di definizione del flusso di lavoro. [Se non esiste alcun repository di origine, creane uno.](#)
6. Nel campo Branch, scegliete un ramo in cui risiederà il file di definizione del flusso di lavoro.
7. Scegli Create (Crea).

Amazon CodeCatalyst salva le informazioni del repository e della filiale in memoria, ma il flusso di lavoro non è ancora stato eseguito.

8. Scegli YAML.
9. Crea il flusso di lavoro:
 - a. (Facoltativo) Aggiungi un trigger al codice YAML. Per ulteriori informazioni, consulta [Aggiungere trigger ai flussi di lavoro.](#)
 - b. Scegli + Azioni (in alto a sinistra). Viene visualizzato il catalogo Actions.
 - c. Scegli il segno più (+) all'interno di un'azione per aggiungerla al flusso di lavoro. Usa il riquadro a destra per configurare l'azione. Per ulteriori informazioni, consulta [Aggiungere un'azione a un flusso di lavoro.](#)
 - d. (Facoltativo) Scegli le proprietà del flusso di lavoro (in alto a destra). Viene visualizzato il riquadro delle proprietà del flusso di lavoro. Configura il nome del flusso di lavoro, la modalità di esecuzione e il calcolo. Per ulteriori informazioni, consultare [Configurazione del comportamento di accodamento delle esecuzioni](#) e [Configurazione delle immagini di calcolo e di runtime.](#)
10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit e, nella finestra di dialogo Commit workflow, effettuate le seguenti operazioni:
 - a. Per il nome del file Workflow, lasciate il nome predefinito o immettetene uno personalizzato. Il file verrà archiviato in una ~/.codecatalyst/workflows/ cartella nel repository e nel ramo di origine scelti. È possibile anteporre al nome del file una cartella o sottocartella. Esempi:

- Se si specifica `my-workflow` (nessuna cartella), il file viene memorizzato come `~/.codecatalyst/workflows/my-workflow.yaml`
 - Specificando `folder/subfolder/my-workflow` memorizza il file come `~/.codecatalyst/workflows/folder/subfolder/my-workflow.yaml`
- b. Per il messaggio di conferma, lascia il messaggio predefinito o inserisci il tuo.
 - c. Per Repository e Branch, scegliete il repository e il branch di origine per il file di definizione del flusso di lavoro. Questi campi devono essere impostati sul repository e sul branch specificati in precedenza nella finestra di dialogo Crea flusso di lavoro. Ora puoi cambiare il repository e il ramo, se lo desideri.

 Note

Dopo aver salvato il file di definizione del flusso di lavoro, non può essere associato a un altro repository o ramo, quindi assicurati di sceglierli con attenzione.

- d. Scegliete Commit per eseguire il commit del file di definizione del flusso di lavoro.

Esecuzione di un workflow

Un'esecuzione è una singola iterazione di un flusso di lavoro. Durante un'esecuzione, CodeCatalyst esegue le azioni definite nel file di configurazione del flusso di lavoro e restituisce i log, gli artefatti e le variabili associati.

È possibile avviare un'esecuzione manualmente o automaticamente, tramite un trigger del flusso di lavoro. Un esempio di attivazione del flusso di lavoro potrebbe essere uno sviluppatore di software che invia un commit alla filiale principale.

Puoi anche interrompere manualmente un flusso di lavoro eseguito a metà dell'elaborazione se lo hai avviato per errore.

Se più esecuzioni del flusso di lavoro vengono avviate all'incirca nello stesso momento, è possibile configurare il modo in cui si desidera che queste esecuzioni vengano messe in coda. È possibile utilizzare il comportamento di accodamento predefinito, in cui le esecuzioni vengono messe in coda una dopo l'altra nell'ordine in cui sono state avviate, oppure è possibile fare in modo che un'esecuzione successiva sostituisca (o «sostituisca») una precedente per velocizzare l'esecuzione.

È inoltre possibile configurare le esecuzioni del flusso di lavoro in modo che avvengano in parallelo, in modo che nessuna esecuzione debba attendere altre esecuzioni.

Dopo aver avviato l'esecuzione di un workflow, manualmente o automaticamente, puoi visualizzare lo stato dell'esecuzione e altri dettagli. Ad esempio, puoi vedere quando è stato avviato, da chi è stato avviato e se è ancora in esecuzione.

Argomenti

- [Avvio manuale dell'esecuzione di un flusso di lavoro](#)
- [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#)
- [Configurazione dei trigger solo manuali](#)
- [Arresto dell'esecuzione di un flusso di lavoro](#)
- [Garantire l'esecuzione di un flusso di lavoro](#)
- [Richiesta di approvazioni per l'esecuzione del flusso di lavoro](#)
- [Configurazione del comportamento di accodamento delle esecuzioni](#)
- [Memorizzazione nella cache dei file tra le esecuzioni del workflow](#)
- [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#)

Avvio manuale dell'esecuzione di un flusso di lavoro

In Amazon CodeCatalyst, puoi avviare un flusso di lavoro eseguito manualmente dalla CodeCatalyst console.

Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, consulta [Esecuzione di un workflow](#).

Note

È inoltre possibile avviare un flusso di lavoro eseguito automaticamente [configurando un trigger](#).

Per avviare un flusso di lavoro, esegui manualmente

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Esegui.

L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger

Puoi avviare automaticamente un CodeCatalyst flusso di lavoro Amazon con un trigger del flusso di lavoro.

Un trigger del flusso di lavoro, o semplicemente un trigger, ti consente di avviare automaticamente un flusso di lavoro quando si verificano determinati eventi, ad esempio l'invio di un codice. Potresti voler configurare i trigger per evitare agli sviluppatori di software di dover avviare manualmente il flusso di lavoro tramite la CodeCatalyst console.

Puoi utilizzare tre tipi di trigger:

- Push: un trigger di invio di codice provoca l'avvio di un flusso di lavoro ogni volta che viene premuto un commit.
- Pull request: un trigger di pull request fa sì che un workflow venga avviato ogni volta che una pull request viene creata, rivista o chiusa.
- Pianificazione: un trigger di pianificazione fa sì che l'esecuzione di un workflow venga avviata in base a una pianificazione definita dall'utente. Prendi in considerazione l'utilizzo di un trigger di pianificazione per eseguire le build notturne del tuo software in modo che la build più recente sia pronta per essere utilizzata dagli sviluppatori il mattino successivo.

Puoi utilizzare i trigger push, pull request e schedule da soli o in combinazione nello stesso flusso di lavoro.

I trigger sono facoltativi: se non ne configuri nessuno, puoi avviare un flusso di lavoro solo manualmente.

Tip

Per vedere un trigger in azione, avvia un progetto con un blueprint. La maggior parte dei blueprint contiene un flusso di lavoro con un trigger. Cerca la `Trigger` proprietà nel file

di definizione del flusso di lavoro del blueprint. Per ulteriori informazioni sui piani, consulta [Creare un progetto con un blueprint](#).

Argomenti

- [Esempi: trigger nei flussi di lavoro](#)
- [Linee guida per l'utilizzo di trigger e filiali](#)
- [Aggiungere trigger ai flussi di lavoro](#)

Esempi: trigger nei flussi di lavoro

Gli esempi seguenti mostrano come aggiungere diversi tipi di trigger in un file di definizione del CodeCatalyst flusso di lavoro Amazon.

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Argomenti

- [Esempio: un semplice pulsante di attivazione tramite codice](#)
- [Esempio: un semplice trigger «push to main»](#)
- [Esempio: un semplice trigger di pull request](#)
- [Esempio: un semplice trigger di pianificazione](#)
- [Esempio: un trigger con una pianificazione e rami](#)
- [Esempio: un trigger con una pianificazione, un push e rami](#)
- [Esempio: un grilletto con trazione e ramificazioni](#)
- [Esempio: un trigger con pull, branch e un evento «CLOSED»](#)
- [Esempio: un trigger con push, branch e file](#)
- [Esempio: un trigger manuale](#)
- [Esempio: trigger in una configurazione CI/CD con più flussi di lavoro](#)

Esempio: un semplice pulsante di attivazione tramite codice

L'esempio seguente mostra un trigger che avvia un flusso di lavoro ogni volta che il codice viene inviato a qualsiasi ramo del repository di origine.

Quando questo trigger è attivato, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando i file nel ramo verso cui state inviando il push (ovvero il ramo di destinazione).

Ad esempio, se invii un commit `main`, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e altri file di origine. `main`

Come altro esempio, se si invia un commit `feature-branch-123`, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e altri file di origine. `feature-branch-123`

Triggers:
- Type: PUSH

Note

Se desideri che l'esecuzione di un flusso di lavoro inizi solo quando esegui il push `main`, consulta. [Esempio: un semplice trigger «push to main»](#)

Esempio: un semplice trigger «push to main»

L'esempio seguente mostra un trigger che avvia l'esecuzione di un flusso di lavoro ogni volta che il codice viene inviato al ramo, `main` e solo al ramo, del repository di origine. `main`

Triggers:
- Type: PUSH
Branches:
- `main`

Esempio: un semplice trigger di pull request

L'esempio seguente mostra un trigger che avvia l'esecuzione di un flusso di lavoro ogni volta che una richiesta pull viene creata o modificata nel repository di origine.

Quando questo trigger è attivato, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e altri file sorgente nel ramo da cui state estraendo (ovvero il ramo di origine).

Ad esempio, se crei una richiesta pull con un ramo di origine chiamato `feature-123` e un ramo di destinazione chiamato `main`, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e altri file di origine. `feature-123`

Triggers:

- Type: PULLREQUEST

Events:

- OPEN
- REVISION

Esempio: un semplice trigger di pianificazione

L'esempio seguente mostra un trigger che avvia un flusso di lavoro eseguito a mezzanotte (UTC+0) dal lunedì al venerdì.

Quando questo trigger è attivato, CodeCatalyst avvia un singolo flusso di lavoro per ogni ramo del repository di origine che contiene un file di definizione del flusso di lavoro con questo trigger.

Ad esempio, se nel repository di origine sono presenti tre rami,, `main` `release-v1feature-123`, e ognuno di questi rami contiene un file di definizione del flusso di lavoro con il trigger seguente, CodeCatalyst avvia tre esecuzioni del flusso di lavoro: una utilizzando i file in `main`, un'altra utilizzando i file in `release-v1` e un'altra utilizzando i file in `feature-123`

Triggers:

- Type: SCHEDULE

Expression: "`0 0 ? * MON-FRI *`"

Per altri esempi di espressioni cron che è possibile utilizzare nella Expression proprietà, vedere.

[Expression](#)

Esempio: un trigger con una pianificazione e rami

L'esempio seguente mostra un trigger che avvia un flusso di lavoro eseguito ogni giorno alle 18:15 (UTC+0).

Quando questo trigger è attivato, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando i file nel `main` ramo e avvia esecuzioni aggiuntive per ogni ramo che inizia con. `release-`

Ad esempio, se sono presenti rami denominati `main`, `release-v1bugfix-1`, e `bugfix-2` nel repository di origine, CodeCatalyst avvia due esecuzioni del flusso di lavoro: una utilizzando i file in `main` e l'altra utilizzando i file in `release-v1`. Non avvia le esecuzioni del flusso di lavoro per i `bugfix-1` rami `bugfix-1` and.

Triggers:

- Type: SCHEDULE

```
Expression: "15 18 * * ? *"  
Branches:  
- main  
- release\-.*
```

Per altri esempi di espressioni cron che è possibile utilizzare nella `Expression` proprietà, vedere [Expression](#).

Esempio: un trigger con una pianificazione, un push e rami

L'esempio seguente mostra un trigger che avvia un flusso di lavoro eseguito a mezzanotte (UTC+0) ogni giorno e ogni volta che il codice viene inviato alla filiale. `main`

In questo esempio:

- L'esecuzione di un flusso di lavoro inizia ogni giorno a mezzanotte. L'esecuzione del flusso di lavoro utilizza il file di definizione del flusso di lavoro e altri file di origine presenti nel `main` ramo.
- L'esecuzione di un flusso di lavoro viene inoltre avviata ogni volta che si invia un commit al `main` ramo. L'esecuzione del flusso di lavoro utilizza il file di definizione del flusso di lavoro e altri file di origine nel ramo di destinazione (`main`).

```
Triggers:  
- Type: SCHEDULE  
  Expression: "0 0 * * ? *"  
  Branches:  
    - main  
- Type: PUSH  
  Branches:  
    - main
```

Per altri esempi di espressioni cron che è possibile utilizzare nella `Expression` proprietà, vedere [Expression](#).

Esempio: un grilletto con trazione e ramificazioni

L'esempio seguente mostra un trigger che avvia un flusso di lavoro ogni volta che qualcuno apre o modifica una richiesta pull con un branch di destinazione chiamato `main`. Sebbene il ramo specificato nella `Triggers` configurazione sia `main`, l'esecuzione del flusso di lavoro utilizzerà il file di definizione del flusso di lavoro e altri file di origine nel ramo di origine (che è il ramo da cui state estraendo).

```
Triggers:
- Type: PULLREQUEST
  Branches:
    - main
  Events:
    - OPEN
    - REVISION
```

Esempio: un trigger con pull, branch e un evento «CLOSED»

L'esempio seguente mostra un trigger che avvia l'esecuzione di un flusso di lavoro ogni volta che una richiesta pull viene chiusa su un ramo che inizia con `main`.

In questo esempio:

- Quando si chiude una richiesta pull con un ramo di destinazione che inizia con `main`, l'esecuzione di un workflow viene avviata automaticamente utilizzando il file di definizione del flusso di lavoro e altri file di origine nel ramo di origine (ora chiuso).
- Se hai configurato il tuo repository di origine per eliminare automaticamente i branch dopo l'unione di una pull request, questi branch non avranno mai la possibilità di entrare nello `CLOSED` stato. Ciò significa che i rami uniti non attiveranno il trigger della pull request. `CLOSED` L'unico modo per attivare il `CLOSED` trigger in questo scenario è chiudere la pull request senza unirla.

```
Triggers:
- Type: PULLREQUEST
  Branches:
    - main.*
  Events:
    - CLOSED
```

Esempio: un trigger con push, branch e file

L'esempio seguente mostra un trigger che avvia l'esecuzione di un flusso di lavoro ogni volta che viene apportata una modifica al `filename.txt` file, o a qualsiasi file nella `src` directory, sul `main` ramo.

Quando questo trigger è attivato, CodeCatalyst avvia un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e altri file di origine nel `main` ramo.

```
Triggers:
```

```
- Type: PUSH
  Branches:
    - main
  FilesChanged:
    - filename.txt
    - src\/*.*
```

Esempio: un trigger manuale

Per configurare un trigger manuale, omettete la `Triggers` sezione dal file di definizione del flusso di lavoro. Senza questa sezione, gli utenti sono costretti ad avviare il flusso di lavoro manualmente scegliendo il pulsante `Esegui` nella CodeCatalyst console. Per ulteriori informazioni, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Esempio: trigger in una configurazione CI/CD con più flussi di lavoro

Questo esempio descrive come configurare i trigger quando desideri utilizzare CodeCatalyst flussi di lavoro Amazon separati per l'integrazione continua (CI) e la distribuzione continua (CD).

In questo scenario, configuri due flussi di lavoro:

- un flusso di lavoro CI: questo flusso di lavoro crea e verifica l'applicazione quando viene creata o modificata una pull request.
- un flusso di lavoro su CD: questo flusso di lavoro crea e distribuisce l'applicazione quando viene unita una pull request.

Il file di definizione del flusso di lavoro CI sarebbe simile al seguente:

```
Triggers:
  - Type: PULLREQUEST
    Branches:
      - main
    Events:
      - OPEN
      - REVISION
Actions:
  BuildAction:
    instructions-for-building-the-app
  TestAction:
    instructions-for-test-the-app
```

Il `Triggers` codice indica di avviare un flusso di lavoro eseguito automaticamente ogni volta che uno sviluppatore di software crea una pull request (o [ne modifica una](#)) chiedendo di unire il proprio feature branch al main branch. CodeCatalyst avvia il flusso di lavoro eseguito utilizzando il codice sorgente nel ramo sorgente (che è il ramo delle funzionalità).

Il file di definizione del flusso di lavoro del CD sarebbe simile al seguente:

```
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildAction:
    instructions-for-building-the-app
  DeployAction:
    instructions-for-deploying-the-app
```

Il `Triggers` codice indica di avviare automaticamente il flusso di lavoro quando si main verifica un'unione a. CodeCatalyst avvia il flusso di lavoro eseguito utilizzando il codice sorgente nel main ramo.

Linee guida per l'utilizzo di trigger e filiali

Questa sezione descrive alcune delle linee guida principali per la configurazione di CodeCatalyst trigger Amazon che includono filiali.

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

- Linea guida 1: per i trigger di richiesta push e pull, se intendi specificare un ramo, devi specificare il ramo di destinazione (o «a») nella configurazione del trigger. Non specificare mai il ramo di origine (o «da»).

Nell'esempio seguente, un push da qualsiasi ramo a main attiva il flusso di lavoro.

```
Triggers:
  - Type: PUSH
    Branches:
      - main
```

Nell'esempio seguente, una richiesta pull da qualsiasi filiale main attiva il flusso di lavoro.

Triggers:

- Type: PULLREQUEST

Branches:

- main

Events:

- OPEN
- REVISION

- Linea guida 2: Per i trigger push, dopo l'attivazione del flusso di lavoro, il flusso di lavoro verrà eseguito utilizzando il file di definizione del flusso di lavoro e i file sorgente nel ramo di destinazione.
- Linea guida 3: Per i trigger di pull request, dopo l'attivazione del flusso di lavoro, il flusso di lavoro verrà eseguito utilizzando il file di definizione del flusso di lavoro e i file sorgente nel ramo di origine (anche se è stato specificato il ramo di destinazione nella configurazione del trigger).
- Linea guida 4: Lo stesso trigger in un ramo potrebbe non essere eseguito in un altro ramo.

Considerate il seguente pulsante:

Triggers:

- Type: PUSH

Branches:

- main

Se il file di definizione del flusso di lavoro contenente questo trigger esiste in `main` e viene clonato `test`, il flusso di lavoro non verrà mai avviato automaticamente utilizzando i file in esso contenuti `test` (sebbene sia possibile avviare il flusso di lavoro manualmente per fare in modo che utilizzi i file in `test` esso contenuti). Consultate la Linea guida 2 per capire perché il flusso di lavoro non verrà mai eseguito automaticamente utilizzando i file in esso contenuti. `test`

Considerate anche il seguente trigger di pull request:

Triggers:

- Type: PULLREQUEST

Branches:

- main

Events:

- OPEN
- REVISION

Se il file di definizione del flusso di lavoro contenente questo trigger esiste in `main`, il flusso di lavoro non verrà mai eseguito utilizzando i file in `main`. (Tuttavia, se si crea un test ramo di `main`, il flusso di lavoro verrà eseguito utilizzando i file in `test`.) Consulta la Linea guida 3 per capire perché.

Aggiungere trigger ai flussi di lavoro

Utilizza le seguenti istruzioni per aggiungere un trigger push, pull o schedulato al tuo CodeCatalyst flusso di lavoro Amazon.

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Visual

Per aggiungere un trigger (editor visivo)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli la casella Source and Triggers.
8. Nel riquadro di configurazione, scegli Aggiungi trigger.
9. Nella finestra di dialogo Aggiungi trigger, fornisci le informazioni nei campi, come segue.

Tipo di trigger

Specificare il tipo di trigger. È possibile utilizzare uno dei seguenti valori:

- Push (editor visivo) o PUSH (editor YAML)

Un push trigger avvia l'esecuzione di un flusso di lavoro quando una modifica viene inviata all'archivio di origine. L'esecuzione del flusso di lavoro utilizzerà i file del ramo verso cui state inviando il push (ovvero il ramo di destinazione).

- Pull request (editor visuale) o PULLREQUEST (editor YAML)

Un trigger di pull request avvia un flusso di lavoro quando una pull request viene aperta, aggiornata o chiusa nel repository di origine. L'esecuzione del flusso di lavoro utilizzerà i file nel ramo da cui state estraendo (ovvero il ramo di origine).

- Pianificazione (editor visivo) o SCHEDULE (editor YAML)

Un trigger di pianificazione avvia l'esecuzione del flusso di lavoro in base a una pianificazione definita da un'espressione cron specificata dall'utente. Verrà avviato un flusso di lavoro separato per ogni ramo del repository di origine utilizzando i file del ramo. (Per limitare i rami su cui si attiva il trigger, usa il campo Branches (editor visivo) o la Branches proprietà (editor YAML).)

Quando configuri un trigger di pianificazione, segui queste linee guida:

- Utilizza solo un trigger di pianificazione per flusso di lavoro.
- Se hai definito più flussi di lavoro nel tuo CodeCatalyst spazio, ti consigliamo di programmarne non più di 10 per avviarli contemporaneamente.
- Assicurati di configurare l'espressione cron del trigger con un tempo adeguato tra le esecuzioni. Per ulteriori informazioni, consulta [Expression](#).

Per alcuni esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

Eventi per la pull request

Questo campo viene visualizzato solo se è stato selezionato il tipo di trigger della richiesta Pull.

Specificate il tipo di eventi di pull request che avvieranno l'esecuzione di un flusso di lavoro. Di seguito sono riportati i valori validi:

- Viene creata una pull request (editor visuale) o OPEN (editor YAML)

L'esecuzione del flusso di lavoro viene avviata quando viene creata una richiesta pull.

- La pull request è chiusa (editor visivo) o CLOSED (editor YAML)

L'esecuzione del flusso di lavoro viene avviata quando viene chiusa una pull request. Il comportamento dell'`CLOSE` evento è complicato e può essere compreso meglio attraverso un esempio. Per ulteriori informazioni, consulta [Esempio: un trigger con pull, branch e un evento «CLOSED»](#).

- Viene effettuata una nuova revisione per pull request (editor visivo) o `REVISION` (editor YAML)

L'esecuzione del workflow viene avviata quando viene creata una revisione di una pull request. La prima revisione viene creata quando viene creata la pull request. Dopodiché, viene creata una nuova revisione ogni volta che qualcuno invia un nuovo commit al ramo di origine specificato nella pull request. Se includi l'`REVISION` evento nel trigger della pull request, puoi omettere l'`OPEN` evento, poiché `REVISION` è un superset di `OPEN`.

È possibile specificare più eventi nello stesso trigger di pull request.

Per alcuni esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

Pianificazione

Questo campo viene visualizzato solo se è stato selezionato il tipo di trigger `Schedule`.

Specificate l'espressione cron che descrive quando desiderate che avvenga l'esecuzione del flusso di lavoro pianificato.

Le espressioni Cron CodeCatalyst utilizzano la seguente sintassi a sei campi, in cui ogni campo è separato da uno spazio:

minutes hours days-of-month month days-of-week year

Esempi di espressioni cron

Minuti	Ore	Giorni del mese	Mese	Giorni della settimana	Anno	Significato
0	0	?	*	LUN-VEN	*	Esegue un flusso di lavoro a mezzanotte (UTC +0) dal lunedì al venerdì.
0	2	*	*	?	*	Esegue un flusso di lavoro ogni giorno alle 2:00 (UTC +0).
15	22	*	*	?	*	Esegue un flusso di lavoro alle 22:15 (UTC +0) ogni giorno.

Minuti	Ore	Giorni del mese	Mese	Giorni della settimana	Anno	Significato
0/30	22-2	?	*	SAT-SUN	*	Esegue un flusso di lavoro ogni 30 minuti dal sabato alla domenica tra le 22:00 del giorno iniziale e le 2:00 del giorno successivo (UTC +0).
45	13	L	*	?	2023-2027	Esegue un flusso di lavoro alle 13:45 (UTC+0) dell'ultimo giorno del mese tra gli anni 2023 e 2027 inclusi.

Quando specifichi le espressioni cron in CodeCatalyst, assicurati di seguire queste linee guida:

- Specificate una singola espressione cron per trigger. SCHEDULE
- Racchiudi l'espressione cron tra virgolette doppie (") nell'editor YAML.
- Specificate l'ora in UTC (Coordinated Universal Time). Gli altri fusi orari non sono supportati.
- Configura almeno 30 minuti tra un'esecuzione e l'altra. Una cadenza più veloce non è supportata.
- Specificate il *days-of-week* campo *days-of-month* o, ma non entrambi. Se si specifica un valore o un asterisco (*) in uno dei campi, è necessario utilizzare un punto interrogativo (?) nell'altro. L'asterisco significa «tutti» e il punto interrogativo significa «qualsiasi».

Per altri esempi di espressioni cron e informazioni sui caratteri jolly come, e ? *L, consulta il [riferimento alle espressioni Cron nella Amazon EventBridge User Guide](#). Le espressioni Cron CodeCatalyst funzionano esattamente EventBridge allo stesso modo.

Per esempi di trigger di pianificazione, vedi. [Esempi: trigger nei flussi di lavoro](#)

Rami e schema delle filiali

(Facoltativo)

Specificate i rami del repository di origine monitorati dal trigger per sapere quando avviare l'esecuzione di un flusso di lavoro. È possibile utilizzare modelli regex per definire i nomi delle filiali. Ad esempio, utilizzare per `main.*` abbinare tutti i rami che iniziano con `main`.

I rami da specificare sono diversi a seconda del tipo di trigger:

- Per un trigger push, specifica i rami verso cui stai eseguendo il push, ovvero i rami di destinazione. Verrà avviata un'esecuzione del flusso di lavoro per ramo corrispondente, utilizzando i file nel ramo corrispondente.

Esempi: `main.*`, `mainline`

- Per un trigger di pull request, specifica i rami verso cui stai inviando il push, ovvero i rami di destinazione. Verrà avviata un'esecuzione del flusso di lavoro per ramo corrispondente,

utilizzando il file di definizione del flusso di lavoro e i file di origine nel ramo di origine (non nel ramo corrispondente).

Esempi: `main.*`, `mainline`, `v1\-.*` (corrisponde ai rami che iniziano con) `v1-`

- Per un trigger di pianificazione, specifica i rami che contengono i file che desideri vengano utilizzati dall'esecuzione pianificata. Verrà avviata un'esecuzione del flusso di lavoro per ramo corrispondente, utilizzando il file di definizione del flusso di lavoro e i file sorgente nel ramo corrispondente.

Esempi: `main.*`, `version\ -1\ .0`

Note

Se non specifichi i rami, il trigger monitora tutti i rami nel tuo repository di origine e avvierà un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e i file sorgente in:

- Il ramo verso cui stai eseguendo il push (per i trigger push). Per ulteriori informazioni, consulta [Esempio: un semplice pulsante di attivazione tramite codice](#).
- Il ramo da cui stai prelevando (per i trigger di pull request). Per ulteriori informazioni, consulta [Esempio: un semplice trigger di pull request](#).
- Tutte le filiali (per i trigger di pianificazione). Verrà avviata un'unica esecuzione del flusso di lavoro per ramo nel repository di origine. Per ulteriori informazioni, consulta [Esempio: un semplice trigger di pianificazione](#).

Per ulteriori informazioni su rami e trigger, consulta. [Linee guida per l'utilizzo di trigger e filiali](#)

Per ulteriori esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

File modificati

Questo campo viene visualizzato solo se è stato selezionato il tipo di trigger di richiesta Push o Pull.

Specificate i file o le cartelle nell'archivio di origine monitorati dal trigger per sapere quando avviare l'esecuzione di un flusso di lavoro. È possibile utilizzare espressioni regolari per abbinare i nomi o i percorsi dei file.

Per alcuni esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere un trigger (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Aggiungi una Triggers sezione e le proprietà sottostanti usando l'esempio seguente come guida. Per ulteriori informazioni, consultare [Triggers](#) nella [Definizione YAML del flusso di lavoro](#).

Un trigger con codice potrebbe avere il seguente aspetto:

```
Triggers:
- Type: PUSH
  Branches:
    - main
```

Un trigger di pull request potrebbe avere il seguente aspetto:

```
Triggers:
- Type: PULLREQUEST
  Branches:
    - main.*
  Events:
    - OPEN
    - REVISION
```

- CLOSED

Un trigger di pianificazione potrebbe essere simile al seguente:

Triggers:

- Type: SCHEDULE

Branches:

- main.*

Run the workflow at 1:15 am (UTC+0) every Friday until the end of 2023

Expression: "15 1 ? * FRI 2022-2023"

Per altri esempi di espressioni cron che è possibile utilizzare nella Expression proprietà, vedere [Expression](#).

Per altri esempi di trigger push, pull request e schedule, vedi. [Esempi: trigger nei flussi di lavoro](#)

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Configurazione dei trigger solo manuali

Puoi limitare un flusso di lavoro in modo che possa essere avviato solo manualmente dal tuo team utilizzando il pulsante Esegui nella CodeCatalyst console. Per configurare questa funzionalità, è necessario rimuovere la Triggers sezione nel file di definizione del flusso di lavoro. La Triggers sezione è inclusa per impostazione predefinita quando si crea un flusso di lavoro, ma è facoltativa e può essere rimossa.

Utilizzate le seguenti istruzioni per rimuovere la Triggers sezione dal file di definizione del flusso di lavoro in modo che il flusso di lavoro possa essere avviato solo manualmente.

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Per ulteriori informazioni sull'esecuzione dei flussi di lavoro, vedere [Esecuzione di un workflow](#).

Visual

Per rimuovere la sezione 'Triggers' (editor visuale)

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Scegliete la casella Origine nel diagramma del flusso di lavoro.
8. In Trigger, scegli l'icona del cestino per rimuovere la Triggers sezione dal flusso di lavoro.
9. (Facoltativo) Scegli Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per rimuovere la sezione 'Triggers' (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Trova la Triggers sezione e rimuovila.
8. (Facoltativo) Scegli Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Arresto dell'esecuzione di un flusso di lavoro

Utilizzare la procedura seguente per interrompere l'esecuzione di un flusso di lavoro in corso. Potresti voler interrompere un'esecuzione se è stata avviata accidentalmente.

Quando interrompi l'esecuzione di un flusso di lavoro, CodeCatalyst attende il completamento delle azioni in corso prima di contrassegnare l'esecuzione come Interrotta nella CodeCatalyst console. Tutte le azioni che non hanno avuto la possibilità di iniziare non verranno avviate e verranno contrassegnate come Abbandonate.

Note

Se un'esecuzione è in coda (ovvero non ha azioni in corso), l'esecuzione viene interrotta immediatamente.

Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, vedere [Esecuzione di un workflow](#)

Per interrompere l'esecuzione di un flusso di lavoro

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. In Flussi di lavoro, scegli Esecuzioni e scegli l'esecuzione in corso dall'elenco.
5. Scegli Stop (Arresta).

Garantire l'esecuzione di un flusso di lavoro

Un gate è un componente del flusso di lavoro che è possibile utilizzare per impedire che l'esecuzione di un flusso di lavoro proceda a meno che non vengano soddisfatte determinate condizioni. Un esempio di gate è l'Approval gate, in cui gli utenti devono inviare un'approvazione nella CodeCatalyst console prima che l'esecuzione del flusso di lavoro possa continuare.

È possibile aggiungere porte tra sequenze di azioni in un flusso di lavoro o prima della prima azione (che viene eseguita immediatamente dopo il download del codice sorgente). Puoi anche aggiungere porte dopo l'ultima azione, se necessario.

Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, consulta [Esecuzione di un workflow](#).

Argomenti

- [Tipi di porte](#)
- [Posso configurare un gate in modo che funzioni in parallelo a un'altra azione?](#)
- [Posso usare un gate per impedire l'avvio di un workflow?](#)
- [Limitazioni dei gate](#)
- [Aggiungere un gate a un flusso di lavoro](#)
- [Sequenziamento di porte e azioni](#)
- [Specificare la versione di un cancello](#)

Tipi di porte

Attualmente, Amazon CodeCatalyst supporta un tipo di cancello: il gate di approvazione. Per ulteriori informazioni, consulta [Richiesta di approvazioni per l'esecuzione del flusso di lavoro](#).

Posso configurare un gate in modo che funzioni in parallelo a un'altra azione?

No. Gates può funzionare solo prima o dopo un'azione. Per ulteriori informazioni, consulta [Sequenziamento di porte e azioni](#).

Posso usare un gate per impedire l'avvio di un workflow?

Sì, con qualifiche.

È possibile impedire l'esecuzione di attività a un flusso di lavoro, il che è leggermente diverso dall'impedirne l'avvio.

Per impedire a un flusso di lavoro di eseguire attività, aggiungi un gate prima della prima azione in un flusso di lavoro. In questo scenario, verrà avviata l'esecuzione di un flusso di lavoro, vale a dire che scaricherà i file del repository di origine, ma gli verrà impedito di eseguire attività finché il gate non verrà sbloccato.

Note

I flussi di lavoro che iniziano e poi vengono bloccati da un gate vengono comunque conteggiati nel numero massimo di flussi di lavoro simultanei per quota di spazio e altre quote. Per assicurarti di non superare le quote del flusso di lavoro, prendi in considerazione l'utilizzo di un trigger del flusso di lavoro per avviare un flusso di lavoro in modo condizionale

anziché utilizzare un gate. Considerate anche l'utilizzo di una regola di approvazione delle pull request anziché di un gate. Per ulteriori informazioni su quote, trigger e regole di approvazione delle pull request, consulta [Quote per i flussi di lavoro in CodeCatalyst](#) L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger, e. [Gestione dei requisiti per l'unione di una pull request con le regole di approvazione](#)

Limitazioni dei gate

I gate presentano le seguenti limitazioni:

- Gates non può essere utilizzato insieme alla funzionalità di condivisione del calcolo. Per ulteriori informazioni sull'utilizzo di questa caratteristica, consulta [Condivisione dell'elaborazione tra le azioni](#).
- I gate non possono essere utilizzati all'interno di gruppi di azione. Per ulteriori informazioni sui gruppi di azioni, vedere [Raggruppamento delle azioni in gruppi di azione](#).

Aggiungere un gate a un flusso di lavoro

In Amazon CodeCatalyst, puoi aggiungere un gate a un flusso di lavoro per impedirne lo svolgimento a meno che non vengano soddisfatte determinate condizioni. Utilizza le seguenti istruzioni per aggiungere un gate a un flusso di lavoro.

Per ulteriori informazioni sui gate, vedere [Garantire l'esecuzione di un flusso di lavoro](#).

Per aggiungere e configurare un cancello

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. A sinistra, scegli Gates.

8. Nel catalogo dei gate, cerca un gate, quindi scegli il segno più (+) per aggiungerlo al tuo flusso di lavoro.
9. Configura il cancello. Scegli Visual per usare l'editor visuale o YAML per usare l'editor YAML. Per istruzioni dettagliate, vedi:
 - [Aggiungere un cancello di «Approvazione»](#)
10. (Facoltativo) Scegli Convalida per assicurarti che il codice YAML sia valido.
11. Scegli Conferma per confermare le modifiche.

Sequenziamento di porte e azioni

In Amazon CodeCatalyst, puoi configurare un gate da eseguire prima o dopo un'azione del flusso di lavoro, un gruppo di azioni o un gate. Ad esempio, potresti impostare un Approval gate da eseguire prima di un'Deployazione. In questo caso, si dice che l'Deployazione dipenda dal Approval cancello.

Per impostare le dipendenze tra le porte e le azioni, configura la proprietà Depends on del cancello o dell'azione. Per istruzioni, consulta [Impostazione delle dipendenze tra le azioni](#). Le istruzioni a cui si fa riferimento si riferiscono alle azioni del flusso di lavoro ma si applicano anche ai gate.

Per un esempio di come configurare la proprietà Depends on con un gate, vedere [Esempio: un cancello di «approvazione»](#).

Per ulteriori informazioni sui cancelli, vedere [Garantire l'esecuzione di un flusso di lavoro](#).

Per ulteriori informazioni sulle azioni del flusso di lavoro, vedere [Configurazione delle azioni del flusso di lavoro](#).

Specificare la versione di un cancello

Per impostazione predefinita, quando aggiungi un gate a un workflow, CodeCatalyst aggiunge la versione completa al file di definizione del workflow utilizzando il formato:

vmajor.minor.patch

Per esempio:

```
My-Gate:
```

```
Identifier: aws/approval@v1
```

È possibile allungare la versione in modo che il flusso di lavoro utilizzi una versione principale o secondaria specifica del gate. Per istruzioni, consulta [Specificare la versione dell'azione da utilizzare](#). L'argomento a cui si fa riferimento si riferisce alle azioni del flusso di lavoro ma si applica anche ai gate.

Per ulteriori informazioni sui gate in CodeCatalyst, vedere [Garantire l'esecuzione di un flusso di lavoro](#).

Richiesta di approvazioni per l'esecuzione del flusso di lavoro

È possibile configurare l'esecuzione di un flusso di lavoro in modo che richieda un'approvazione prima di procedere. A tale scopo, è necessario aggiungere una [porta](#) di approvazione al flusso di lavoro. Un cancello di approvazione impedisce che un flusso di lavoro proceda finché un utente o un gruppo di utenti non invia una o più approvazioni nella console. CodeCatalyst Una volta concesse tutte le approvazioni, il gate viene «sbloccato» e l'esecuzione del flusso di lavoro può riprendere.

Utilizza un Approval gate nel tuo flusso di lavoro per dare ai team di sviluppo, operativi e di leadership la possibilità di esaminare le modifiche prima che vengano distribuite a un pubblico più ampio.

Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, consulta [Esecuzione di un workflow](#).

Argomenti

- [Come posso sbloccare un cancello di approvazione?](#)
- [Quando utilizzare la porta «Approvazione»](#)
- [Chi può fornire un'approvazione?](#)
- [Come faccio a notificare agli utenti che è necessaria un'approvazione?](#)
- [Posso utilizzare una porta di «Approvazione» per impedire l'avvio di un flusso di lavoro?](#)
- [Come funzionano le approvazioni dei flussi di lavoro con le modalità di esecuzione in coda, sostituita e parallela?](#)
- [Esempio: un cancello di «approvazione»](#)
- [Aggiungere un cancello di «Approvazione»](#)
- [Configurazione delle notifiche di approvazione](#)
- [Approvazione o rifiuto dell'esecuzione di un flusso di lavoro](#)

- [Porta «Approvazione» YAML](#)

Come posso sbloccare un cancello di approvazione?

Per sbloccare un cancello di approvazione, devono essere soddisfatte tutte le seguenti condizioni:

- Condizione 1: è necessario presentare il numero richiesto di approvazioni. Il numero richiesto di approvazioni è configurabile e ogni utente può inviare una singola approvazione.
- Condizione 2: Tutte le approvazioni devono essere presentate prima del timeout del gate. Il cancello scade 14 giorni dopo l'attivazione. Questo periodo non è configurabile.
- Condizione 3: nessuno deve rifiutare l'esecuzione del flusso di lavoro. Un singolo rifiuto causerà l'interruzione dell'esecuzione del flusso di lavoro.
- Condizione 4: (Si applica solo se si utilizza la modalità di esecuzione sostituita). L'esecuzione non deve essere sostituita da un'esecuzione successiva. Per ulteriori informazioni, consulta [Come funzionano le approvazioni dei flussi di lavoro con le modalità di esecuzione in coda, sostituita e parallela?](#).

Se una qualsiasi delle condizioni non viene soddisfatta, CodeCatalyst interrompe il flusso di lavoro e imposta lo stato di esecuzione su Non riuscito (nel caso delle condizioni da 1 a 3) o Sostituito (nel caso della Condizione 4).

Quando utilizzare la porta «Approvazione»

In genere, si utilizza un gate di approvazione in un flusso di lavoro che distribuisce applicazioni e altre risorse su un server di produzione o in qualsiasi ambiente in cui è necessario convalidare gli standard di qualità. Posizionando il gate prima dell'implementazione in produzione, offri ai revisori la possibilità di convalidare la nuova revisione del software prima che diventi disponibile al pubblico.

Chi può fornire un'approvazione?

Qualsiasi utente membro del progetto e con il ruolo di collaboratore o amministratore del progetto può fornire un'approvazione. Anche gli utenti con il ruolo di amministratore dello spazio che appartengono allo spazio del progetto possono fornire un'approvazione.

Note

Gli utenti con il ruolo Revisore non possono fornire approvazioni.

Come faccio a notificare agli utenti che è necessaria un'approvazione?

Per notificare agli utenti che è necessaria un'approvazione, devi:

- Ho CodeCatalyst inviato loro una notifica Slack. Per ulteriori informazioni, consulta [Configurazione delle notifiche di approvazione](#).
- Vai alla pagina della CodeCatalyst console in cui si trovano i pulsanti Approva e Rifiuta e incolla l'URL della pagina in un'applicazione di posta elettronica o di messaggistica indirizzata agli approvatori. Per ulteriori informazioni su come accedere a questa pagina, consulta [Approvazione o rifiuto dell'esecuzione di un flusso di lavoro](#)

Posso utilizzare una porta di «Approvazione» per impedire l'avvio di un flusso di lavoro?

Sì, con qualifche. Per ulteriori informazioni, consulta [Posso usare un gate per impedire l'avvio di un workflow?](#).

Come funzionano le approvazioni dei flussi di lavoro con le modalità di esecuzione in coda, sostituita e parallela?

[Quando si utilizza la modalità in coda, sostituita o di esecuzione parallela, la porta Approval funziona in modo simile alle azioni.](#) Ti consigliamo di leggere le [Informazioni sulla modalità di esecuzione parallela](#) sezioni [Informazioni sulla modalità di esecuzione in coda](#), [Informazioni sulla modalità di esecuzione sostituita](#), per acquisire familiarità con queste modalità di esecuzione. Una volta acquisita una conoscenza di base delle stesse, torna a questa sezione per scoprire come funzionano queste modalità di esecuzione quando è presente l'Approval gate.

Quando è presente l'Approval gate, le esecuzioni vengono elaborate come segue:

- Se utilizzi la [modalità di esecuzione in coda](#), le esecuzioni verranno messe in coda dopo l'esecuzione attualmente in attesa di approvazione al gate. Quando il gate viene sbloccato (ossia, tutte le approvazioni sono state concesse), la corsa successiva della coda avanza fino al gate e attende le approvazioni. Questo processo continua con l'elaborazione delle esecuzioni in coda attraverso il gate. one-by-one [Figure 1](#) illustra questo processo.
- Se si utilizza la modalità di [esecuzione sostituita, il comportamento è lo stesso della modalità di esecuzione in coda](#), tranne per il fatto che invece di avere delle esecuzioni accumulate nella coda all'ingresso, le esecuzioni più recenti sostituiscono (sostituiscono) le esecuzioni precedenti. Non ci

sono code e qualsiasi esecuzione attualmente in attesa di approvazione all'uscita verrà annullata e sostituita da una nuova esecuzione. [Figure 2](#) illustra questo processo.

- Se utilizzi la [modalità di esecuzione parallela](#), le esecuzioni iniziano in parallelo e non si formano code. Ogni esecuzione viene elaborata immediatamente dal gate poiché non ci sono esecuzioni precedenti. [Figure 3](#) illustra questo processo.

Figura 1: «modalità di esecuzione in coda» e un cancello di approvazione

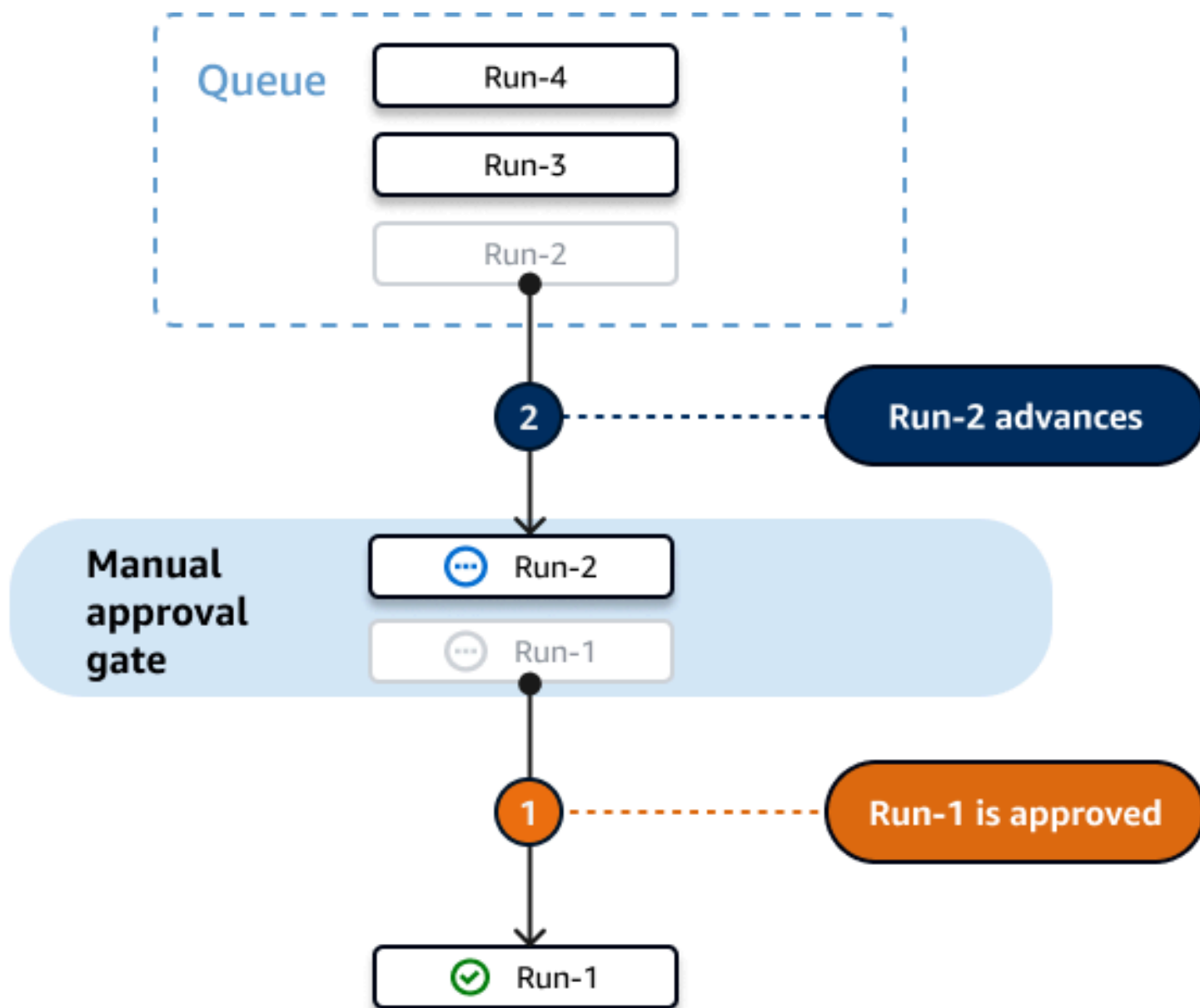


Figura 2: «Modalità di esecuzione sostituita» e porta di approvazione

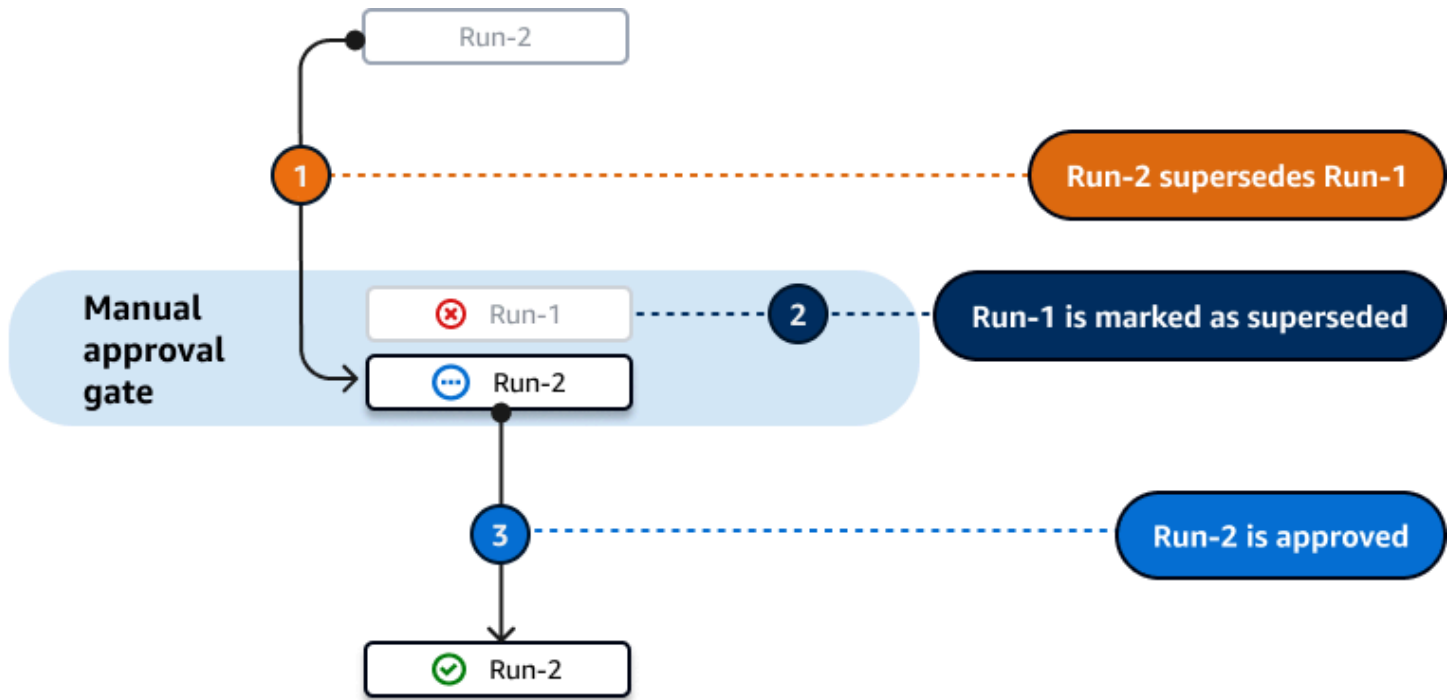
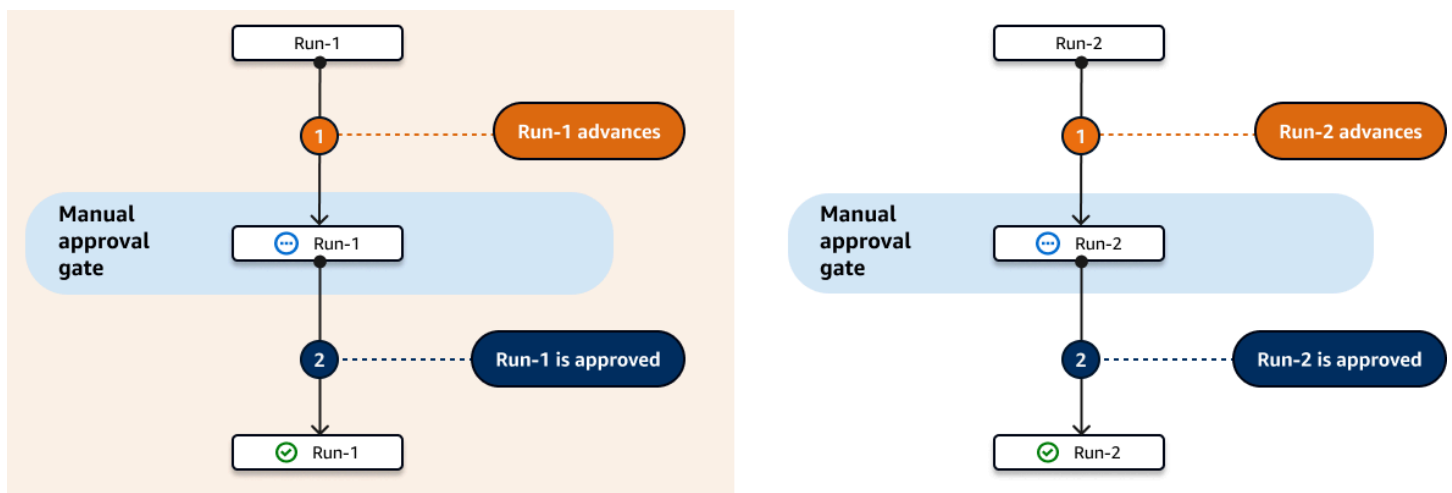


Figura 3: «modalità di esecuzione parallela» e porta di approvazione



Esempio: un cancello di «approvazione»

L'esempio seguente mostra come aggiungere un gate di approvazione chiamato `Approval_01` tra due azioni chiamate `Staging`, e `Production`. L'`Staging` azione viene eseguita per prima, il `Approval_01` gate dopo e l'`Production` azione per ultima. L'`Production` azione viene eseguita solo se il `Approval_01` cancello è sbloccato. La `DependsOn` proprietà garantisce che le `Production` fasi `StagingApproval_01`, e vengano eseguite in ordine sequenziale.

Per ulteriori informazioni sull'Approval gate, vedere [Richiesta di approvazioni per l'esecuzione del flusso di lavoro](#).

Actions:

```
Staging: # Deploy to a staging server
  Identifier: aws/ecs-deploy@v1
  Configuration:
  ...
Approval_01:
  Identifier: aws/approval@v1
  DependsOn:
    - Staging
  Configuration:
    ApprovalsRequired: 2
Production: # Deploy to a production server
  Identifier: aws/ecs-deploy@v1
  DependsOn:
    - Approval_01
  Configuration:
  ...
```

Aggiungere un cancello di «Approvazione»

Per configurare il flusso di lavoro in modo che richieda un'approvazione, è necessario aggiungere la porta di approvazione al flusso di lavoro. Utilizza le seguenti istruzioni per aggiungere una porta di approvazione al tuo flusso di lavoro.

Per ulteriori informazioni su questo cancello, vedi [Richiesta di approvazioni per l'esecuzione del flusso di lavoro](#).

Visual

Per aggiungere una porta di «Approvazione» a un flusso di lavoro (editor visivo)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.

5. Scegli Modifica.
6. In alto a sinistra, scegli Gates.
7. Nel catalogo Gates, in Approvazione, scegli il segno più (+).
8. Scegliete Input e, nel campo Dipende da, effettuate le seguenti operazioni.

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente affinché questo gate possa funzionare. Per impostazione predefinita, quando aggiungi un gate a un workflow, il gate è impostato in modo da dipendere dall'ultima azione del workflow. Se rimuovi questa proprietà, il gate non dipenderà da nulla e verrà eseguito per primo, prima di altre azioni.

 Note

Un gate deve essere configurato per essere eseguito prima o dopo un'azione, un gruppo di azioni o un gate. Non può essere configurato per funzionare in parallelo con altre azioni, gruppi di azioni e gate.

Per ulteriori informazioni sulla funzionalità Depends on, vedere [Sequenziamento di porte e azioni](#).

9. Scegli la scheda Configurazione.
10. Nel campo Gate name, effettuate le seguenti operazioni.

Specificate il nome che volete dare al cancello. Tutti i nomi delle porte devono essere univoci all'interno del flusso di lavoro. I nomi delle porte sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle porte.

11. (Facoltativo) Nel campo Numero di approvazioni, procedi come segue.

Specificate il numero minimo di approvazioni necessarie per sbloccare l'Approval gate. Il minimo è 1. Il massimo è 2. Se omissso, il valore predefinito è 1.

 Note

Se desideri omettere la `ApprovalsRequired` proprietà, rimuovi la `Configuration` sezione del gate dal file di definizione del flusso di lavoro.

12. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
13. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere una porta di «Approvazione» a un flusso di lavoro (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Aggiungi una Approval sezione e le proprietà sottostanti usando l'esempio seguente come guida. Per ulteriori informazioni, consultare [Porta «Approvazione» YAML](#) nella [Definizione YAML del flusso di lavoro](#).

```
Actions:
  MyApproval_01:
    Identifier: aws/approval@v1
    DependsOn:
      - PreviousAction
    Configuration:
      ApprovalsRequired: 2
```

Per un altro esempio, consulta [Esempio: un cancello di «approvazione»](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Configurazione delle notifiche di approvazione

Puoi fare in modo che venga CodeCatalyst inviata una notifica a un canale Slack per informare gli utenti che l'esecuzione di un flusso di lavoro richiede un'approvazione. Gli utenti visualizzano la notifica e fanno clic sul link al suo interno. Il link li porta a una pagina di CodeCatalyst approvazione in cui possono approvare o rifiutare il flusso di lavoro.

Puoi anche configurare le notifiche per informare gli utenti che un flusso di lavoro è stato approvato, rifiutato o che la richiesta di approvazione è scaduta.

Usa le seguenti istruzioni per configurare le notifiche di Slack.

Prima di iniziare

Assicurati di aver aggiunto un cancello di approvazione al tuo flusso di lavoro. Per ulteriori informazioni, consulta [Aggiungere un cancello di «Approvazione»](#).

Per inviare notifiche di approvazione del flusso di lavoro a un canale Slack

1. Configura CodeCatalyst con Slack. Per ulteriori informazioni, consulta [Guida introduttiva alle notifiche di Slack](#).
2. Nel CodeCatalyst progetto che contiene il flusso di lavoro che richiede un'approvazione, abilita le notifiche, se non sono già abilitate. Per abilitare le notifiche:
 - a. Vai al tuo progetto e nel riquadro di navigazione, scegli Impostazioni progetto.
 - b. In alto, scegli Notifiche.
 - c. In Eventi di notifica, scegli Modifica notifiche.
 - d. Attiva l'approvazione del flusso di lavoro in sospeso e scegli un canale Slack a cui inviare la CodeCatalyst notifica.
 - e. (Facoltativo) Attiva notifiche aggiuntive per avvisare gli utenti delle approvazioni approvate, rifiutate e scadute. Puoi attivare l'approvazione dell'esecuzione del flusso di lavoro, l'esecuzione del flusso di lavoro rifiutata, l'approvazione del flusso di lavoro sostituita e l'approvazione del flusso di lavoro è scaduta. Accanto a ciascuna notifica, scegli il canale Slack a cui inviare la CodeCatalyst notifica.
 - f. Seleziona Salva.

Approvazione o rifiuto dell'esecuzione di un flusso di lavoro

Le esecuzioni del flusso di lavoro che includono la porta di approvazione dovranno essere approvate o rifiutate. Gli utenti possono fornire la propria approvazione o rifiuto a partire da:

- la console CodeCatalyst
- un link fornito da un membro del team
- una notifica automatica su Slack

Dopo che un utente ha fornito l'approvazione o il rifiuto, questa decisione non può essere annullata.

Note

Solo alcuni utenti possono approvare o rifiutare l'esecuzione di un flusso di lavoro. Per ulteriori informazioni, consulta [Chi può fornire un'approvazione?](#).

Prima di iniziare

Assicurati di aver aggiunto un cancello di approvazione al tuo flusso di lavoro. Per ulteriori informazioni, consulta [Aggiungere un cancello di «Approvazione»](#).

Per approvare o rifiutare un flusso di lavoro, esegui a partire dalla console. CodeCatalyst

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Nel diagramma del flusso di lavoro, scegli la casella che rappresenta la porta di approvazione.

Viene visualizzato un pannello laterale.

Note

A questo punto, puoi inviare l'URL di questa pagina ad altri approvatori, se lo desideri.

6. In Revisione della decisione, scegli Approva o Rifiuta.
7. (Facoltativo) In Commento: facoltativo, inserisci un commento che indichi il motivo per cui hai approvato o rifiutato l'esecuzione del flusso di lavoro.
8. Scegli Invia.

Per approvare o rifiutare un flusso di lavoro, esegui a partire da un link fornito da un membro del team.

1. Scegli il link che ti è stato inviato dal membro del team. (Puoi chiedere al membro del tuo team di leggere la procedura precedente per ottenere il link.)
2. Accedi a CodeCatalyst, se richiesto.

Verrai reindirizzato alla pagina di approvazione dell'esecuzione del flusso di lavoro.

3. In Decisione di revisione, scegli Approva o Rifiuta.
4. (Facoltativo) In Commento: facoltativo, inserisci un commento che indichi il motivo per cui hai approvato o rifiutato l'esecuzione del flusso di lavoro.
5. Scegli Invia.

Per approvare o rifiutare un flusso di lavoro, esegui a partire da una notifica automatica di Slack

1. Assicurati che le notifiche Slack siano configurate. Per informazioni, consulta [Configurazione delle notifiche di approvazione](#).
2. In Slack, nel canale a cui è stata inviata la notifica di approvazione, scegli il link nella notifica di approvazione.
3. Accedi a CodeCatalyst, se richiesto.

Verrai reindirizzato alla pagina di esecuzione del flusso di lavoro.

4. Nel diagramma del flusso di lavoro, scegli la porta di approvazione.
5. In Decisione di revisione, scegli Approva o Rifiuta.
6. (Facoltativo) In Commento: facoltativo, inserisci un commento che indichi il motivo per cui hai approvato o rifiutato l'esecuzione del flusso di lavoro.
7. Scegli Invia.

Porta «Approvazione» YAML

Di seguito è riportata la definizione YAML della porta di approvazione. Per imparare a usare questo cancello, vedi [Richiesta di approvazioni per l'esecuzione del flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The 'Approval' gate definition starts here.
Approval:
  Identifier: aws/approval@v1
  DependsOn:
    - another-action
  Configuration:
    ApprovalsRequired: number
```

Approval

(Obbligatorio)

Specificate il nome che volete dare al cancello. Tutti i nomi delle porte devono essere univoci all'interno del flusso di lavoro. I nomi delle porte sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle porte.

Default: Approval_nn.

Interfaccia utente corrispondente: scheda di configurazione/nome del gate

Identifier

(*Approval*/Identifier)

(Obbligatorio)

Identifica la porta. La porta di approvazione supporta la versione 1.0.0. Non modificate questa proprietà a meno che non vogliate abbreviare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: aws/approval@v1.

Interfaccia utente corrispondente: Workflow diagram/ Approval _nn/ aws/approval @v1 label

DependsOn

(*Approval*/DependsOn)

(Facoltativo)

Specificare un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente affinché questo gate possa funzionare. Per impostazione predefinita, quando aggiungi un gate a un workflow, il gate è impostato in modo da dipendere dall'ultima azione del workflow. Se rimuovi questa proprietà, il gate non dipenderà da nulla e verrà eseguito per primo, prima di altre azioni.

 Note

Un gate deve essere configurato per essere eseguito prima o dopo un'azione, un gruppo di azioni o un gate. Non può essere configurato per funzionare in parallelo con altre azioni, gruppi di azioni e gate.

Per ulteriori informazioni sulla funzionalità Depends on, vedere [Sequenziamento di porte e azioni](#).

Interfaccia utente corrispondente: scheda Inputs/ Depends on

Configuration

(*Approval*/Configuration)

(Facoltativo)

Una sezione in cui è possibile definire le proprietà di configurazione del gate.

Interfaccia utente corrispondente: scheda Configurazione

ApprovalsRequired

(*Approval*/Configuration/ApprovalsRequired)

(Facoltativo)

Specificate il numero minimo di approvazioni necessarie per sbloccare la porta di approvazione. Il minimo è 1. Il massimo è 2. Se omissso, il valore predefinito è 1.

Note

Se desideri omettere la ApprovalsRequired proprietà, rimuovi la Configuration sezione del gate dal file di definizione del flusso di lavoro.

Interfaccia utente corrispondente: scheda di configurazione/ Numero di approvazioni

Configurazione del comportamento di accodamento delle esecuzioni

Per impostazione predefinita CodeCatalyst, in Amazon, quando si verificano più flussi di lavoro contemporaneamente, li mette in CodeCatalyst coda e li elabora uno per uno, nell'ordine in cui sono stati avviati. Puoi modificare questo comportamento predefinito specificando una modalità di esecuzione. Esistono alcune modalità di esecuzione:

- (Impostazione predefinita) Modalità di esecuzione in coda: CodeCatalyst i processi vengono eseguiti uno per uno
- Modalità di esecuzione sostituita: CodeCatalyst i processi vengono eseguiti uno per uno, con le esecuzioni più recenti che sostituiscono quelle precedenti
- Modalità di esecuzione parallela: CodeCatalyst i processi vengono eseguiti in parallelo

Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, vedere [Esecuzione di un workflow](#).

Argomenti

- [Informazioni sulla modalità di esecuzione in coda](#)
- [Informazioni sulla modalità di esecuzione sostituita](#)
- [Informazioni sulla modalità di esecuzione parallela](#)
- [Configurazione della modalità di esecuzione](#)

Informazioni sulla modalità di esecuzione in coda

In modalità di esecuzione in coda, le esecuzioni vengono eseguite in serie, con le esecuzioni in attesa che formano una coda.

Le code si formano nei punti di ingresso alle azioni e ai gruppi di azioni, quindi è possibile avere più code all'interno dello stesso flusso di lavoro (vedi). [Figure 1](#) Quando un'esecuzione in coda entra in un'azione, l'azione viene bloccata e non possono entrare altre esecuzioni. Quando la corsa termina ed esce dall'azione, l'azione viene sbloccata e pronta per l'esecuzione successiva.

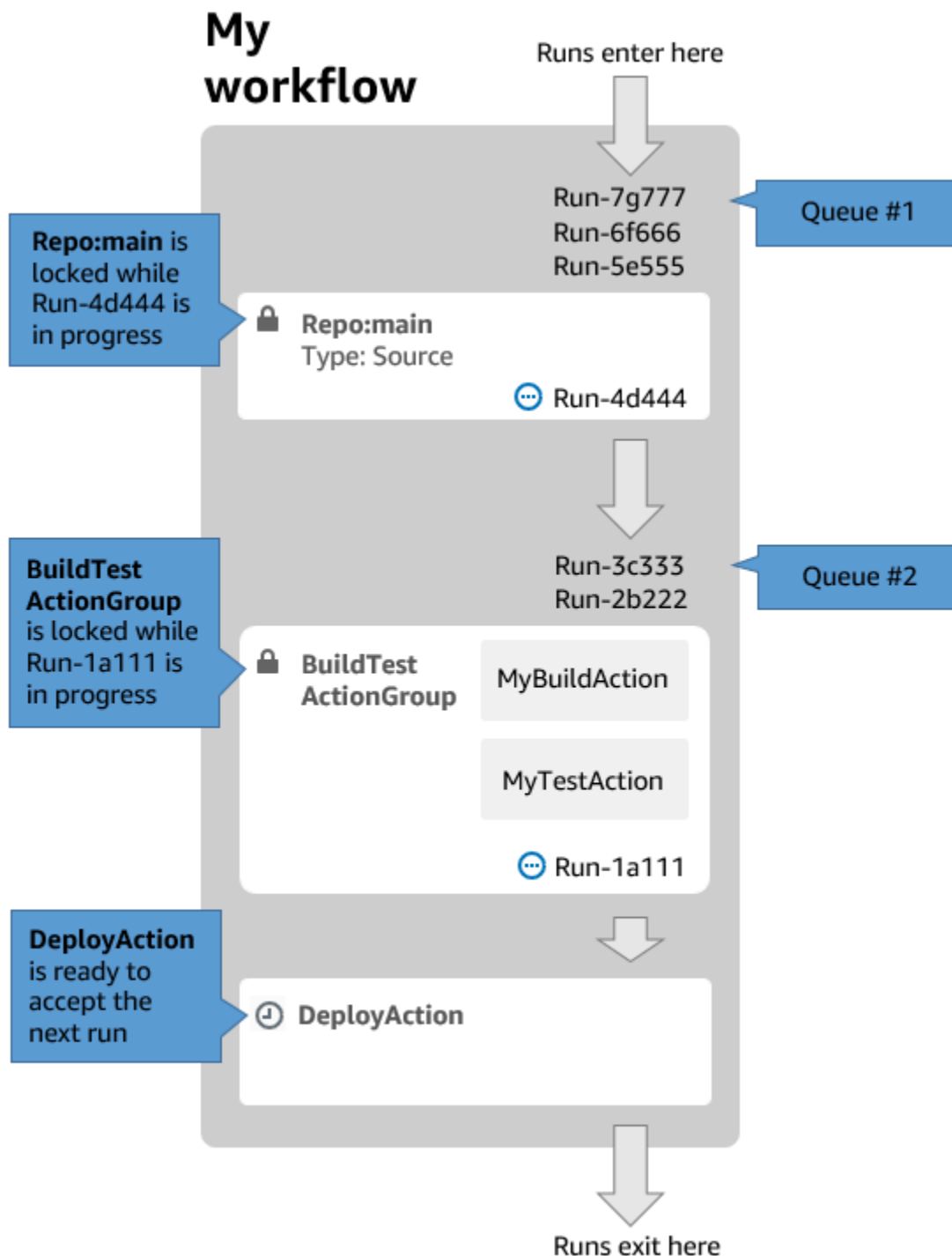
[Figure 1](#) illustra un flusso di lavoro configurato in modalità di esecuzione in coda. Mostra le seguenti informazioni:

- Sette esecuzioni si fanno strada nel flusso di lavoro.
- Due code: una all'esterno dell'ingresso della sorgente di input (Repo:main) e una all'esterno dell'ingresso dell'azione. BuildTestActionGroup
- Due blocchi bloccati: la sorgente di ingresso (Repo:main) e il. BuildTestActionGroup

Ecco come andranno le cose man mano che il flusso di lavoro viene eseguito e terminato l'elaborazione:

- Quando run-4D444 termina la clonazione del repository di origine, uscirà dalla sorgente di input e si unirà alla coda dietro RUN-3c333. Quindi, RUN-5e555 entrerà nella sorgente di input.
- Quando Run-1a111 terminerà la costruzione e il test, uscirà dall'azione ed entrerà nell'BuildTestActionGroupazione. DeployAction Quindi, Run-2b222 entrerà in azione. BuildTestActionGroup

Figura 1: Un flusso di lavoro configurato in «modalità di esecuzione in coda»



Utilizza la modalità di esecuzione in coda se:

- Desideri mantenere una one-to-one relazione tra funzionalità ed esecuzioni: queste funzionalità possono essere raggruppate quando si utilizza la modalità sostituita. Ad esempio, quando si unisce la funzionalità 1 nel commit 1, viene avviata l'esecuzione 1 e quando si unisce la funzionalità 2 nel commit 2, viene avviata l'esecuzione 2 e così via. Se dovessi utilizzare la modalità sostituita

anziché la modalità in coda, le tue funzionalità (e i commit) verranno raggruppati nella sequenza che sostituisce le altre.

- Vuoi evitare condizioni di gara e problemi imprevisti che possono verificarsi quando si utilizza la modalità parallela. Ad esempio, se due sviluppatori di software, Wang e Saanvi, avviano il flusso di lavoro all'incirca nello stesso momento per la distribuzione in un cluster Amazon ECS, l'esecuzione di Wang potrebbe avviare i test di integrazione sul cluster mentre l'esecuzione di Saanvi distribuisce un nuovo codice applicativo nel cluster, causando il fallimento dei test di Wang o il test del codice sbagliato. Come altro esempio, potresti avere un obiettivo che non dispone di un meccanismo di blocco, nel qual caso le due esecuzioni potrebbero sovrascrivere le reciproche modifiche in modi imprevisti.
- Vuoi limitare il carico sulle risorse di calcolo CodeCatalyst utilizzate per elaborare le tue esecuzioni. Ad esempio, se hai tre azioni nel tuo flusso di lavoro, puoi avere un massimo di tre esecuzioni contemporaneamente. L'imposizione di un limite al numero di esecuzioni che possono essere eseguite contemporaneamente rende il throughput di esecuzione più prevedibile.
- Si desidera limitare il numero di richieste effettuate a servizi di terze parti dal flusso di lavoro. Ad esempio, il flusso di lavoro potrebbe avere un'azione di compilazione che include istruzioni per estrarre un'immagine da Docker Hub. [Docker Hub limita il numero di pull request](#) che puoi effettuare a un determinato numero all'ora per account e, se superi il limite, non potrai più accedere. L'utilizzo della modalità di esecuzione in coda per rallentare il throughput di esecuzione avrà l'effetto di generare meno richieste all'ora verso Docker Hub, limitando così il potenziale di blocchi e i conseguenti errori di compilazione ed esecuzione.

Dimensione massima della coda: 50

Note sulla dimensione massima della coda:

- La dimensione massima della coda si riferisce al numero massimo di esecuzioni consentite in tutte le code del flusso di lavoro.
- Se una coda supera le 50 esecuzioni, CodeCatalyst interrompe la 51a esecuzione e le successive.

Comportamento di errore:

Se un'esecuzione non risponde mentre viene elaborata da un'azione, le esecuzioni successive vengono mantenute in coda fino al timeout dell'azione. Le azioni scadono dopo un'ora.

Se un'esecuzione fallisce all'interno di un'azione, la prima esecuzione in coda che l'ha preceduta può continuare.

Informazioni sulla modalità di esecuzione sostituita

La modalità di esecuzione sostituita è la stessa della modalità di esecuzione in coda, tranne per il fatto che:

- Se un'esecuzione in coda raggiunge un'altra esecuzione in coda, l'esecuzione successiva sostituisce (sostituisce) l'esecuzione precedente e l'esecuzione precedente viene annullata e contrassegnata come «sostituita».
- Come risultato del comportamento descritto nel primo bullet, una coda può includere una sola esecuzione quando viene utilizzata la modalità di esecuzione sostituita.

Utilizzando il workflow in [Figure 1](#) come guida, l'applicazione della modalità di esecuzione sostituita a questo flusso di lavoro comporterebbe quanto segue:

- Run-7g777 sostituirebbe le altre due esecuzioni nella sua coda e sarebbe l'unica esecuzione rimasta nella coda #1. RUN-6f666 e RUN-5e555 verrebbero annullati.
- Run-3c333 sostituirebbe RUN-2b222 e sarebbe l'unica esecuzione rimasta nella coda #2. Run-2b222 verrebbe annullata.

Usa la modalità di esecuzione sostituita se desideri:

- velocità di trasmissione migliore rispetto alla modalità in coda
- ancora meno richieste verso i servizi di terze parti rispetto alla modalità in coda; questo è vantaggioso se il servizio di terze parti ha limiti di tariffa, come Docker Hub

Informazioni sulla modalità di esecuzione parallela

In modalità di esecuzione parallela, le esecuzioni sono indipendenti l'una dall'altra e non attendono il completamento delle altre esecuzioni prima di iniziare. Non ci sono code e la velocità di esecuzione è limitata solo dalla velocità di completamento delle azioni all'interno del flusso di lavoro.

Utilizza la modalità di esecuzione parallela in ambienti di sviluppo in cui ogni utente ha il proprio ramo di funzionalità e si distribuisce su destinazioni non condivise da altri utenti.

Important

Se disponi di un target condiviso su cui più utenti possono eseguire l'implementazione, ad esempio una funzione Lambda in un ambiente di produzione, non utilizzare la modalità parallela, poiché potrebbero verificarsi condizioni di gara. Una condizione di gara si verifica quando l'esecuzione di un flusso di lavoro parallelo tenta di modificare contemporaneamente una risorsa condivisa, con risultati imprevedibili.

Numero massimo di esecuzioni parallele: 1000 per CodeCatalyst spazio

Configurazione della modalità di esecuzione

È possibile impostare la modalità di esecuzione in coda, sostituita o in parallelo. L'impostazione predefinita è in coda.

Quando si modifica la modalità di esecuzione da coda o sostituita a parallela CodeCatalyst , annulla le esecuzioni in coda e consente alle esecuzioni attualmente elaborate da un'azione di terminare prima di annullarle.

Quando si modifica la modalità di esecuzione da parallela a quella in coda o sostituita, consente il completamento di CodeCatalyst tutte le esecuzioni parallele attualmente in esecuzione. Tutte le esecuzioni avviate dopo aver cambiato la modalità di esecuzione in coda o sostituita utilizzano la nuova modalità.

Visual

Per modificare la modalità di esecuzione utilizzando l'editor visuale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. In alto a destra, scegli Proprietà del flusso di lavoro.

7. Espandi Avanzate e, in modalità Esegui, scegli una delle seguenti opzioni:
 - a. In coda: vedi [Informazioni sulla modalità di esecuzione in coda](#)
 - b. Sostituito: vedi [Informazioni sulla modalità di esecuzione sostituita](#)
 - c. Parallelo: vedi [Informazioni sulla modalità di esecuzione parallela](#)
8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per cambiare la modalità di esecuzione utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Aggiungi la RunMode proprietà, in questo modo:

```
Name: Workflow_6d39
SchemaVersion: "1.0"
RunMode: QUEUED/SUPERSEDED/PARALLEL
```

Per ulteriori informazioni, consulta la descrizione della RunMode proprietà nella [Proprietà di primo livello](#) sezione del [Definizione YAML del flusso di lavoro](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Memorizzazione nella cache dei file tra le esecuzioni del workflow

Quando la memorizzazione nella cache dei file è abilitata, le azioni di compilazione e test salvano i file su disco in una cache e li ripristinano da tale cache nelle successive esecuzioni del flusso di lavoro. La memorizzazione nella cache riduce la latenza causata dalla creazione o dal download di dipendenze che non sono cambiate tra un'esecuzione e l'altra. CodeCatalyst supporta anche le cache di fallback, che possono essere utilizzate per ripristinare cache parziali contenenti alcune delle dipendenze necessarie. Questo aiuta a ridurre l'impatto sulla latenza di una mancata memorizzazione della cache.

Note

La memorizzazione nella cache dei file è disponibile solo con le azioni di CodeCatalyst [build](#) e [test](#) di Amazon e solo quando sono configurate per utilizzare il tipo di EC2 [elaborazione](#).

Argomenti

- [Informazioni sulla memorizzazione nella cache dei file](#)
- [Creazione di una cache](#)
- [Vincoli di memorizzazione nella cache dei file](#)

Informazioni sulla memorizzazione nella cache dei file

La memorizzazione nella cache dei file consente di organizzare i dati in più cache, ognuna delle quali è referenziata nella proprietà. `FileCaching` Ogni cache salva una directory specificata da un determinato percorso. La directory specificata verrà ripristinata nelle future esecuzioni del flusso di lavoro. Di seguito è riportato un esempio di frammento YAML per la memorizzazione nella cache con più cache denominate `and.cacheKey1` `cacheKey2`

```
Actions:
  BuildMyNpmApp:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
```

```
- Run: npm install
- Run: npm run test
Caching:
  FileCaching:
    cacheKey1:
      Path: file1.txt
      RestoreKeys:
        - restoreKey1
    cacheKey2:
      Path: /root/repository
      RestoreKeys:
        - restoreKey2
        - restoreKey3
```

Note

CodeCatalyst utilizza la memorizzazione nella cache multilivello, che consiste in una cache locale e una cache remota. Quando le flotte di macchine o le macchine on-demand predisposte rilevano una perdita di cache in una cache locale, le dipendenze verranno ripristinate da una cache remota. Di conseguenza, alcune esecuzioni di azioni potrebbero presentare una latenza dovuta al download di una cache remota.

CodeCatalyst applica restrizioni di accesso alla cache per garantire che un'azione in un flusso di lavoro non possa modificare le cache di un flusso di lavoro diverso. Ciò protegge ogni flusso di lavoro da altri che potrebbero inviare dati errati con impatto sulle build o sulle implementazioni. Le restrizioni vengono applicate con cache-scopes che isolano le cache per ogni flusso di lavoro e associazione di filiali. Ad esempio, `workflow-A` in branch `feature-A` ha una cache di file diversa rispetto a quella del ramo gemello `workflow-A feature-B`.

Gli errori di cache si verificano quando un flusso di lavoro cerca una cache di file specificata e non riesce a trovarla. Ciò può verificarsi per diversi motivi, ad esempio quando viene creato un nuovo ramo o quando si fa riferimento a una nuova cache che non è ancora stata creata. Può verificarsi anche alla scadenza di una cache, che per impostazione predefinita si verifica 14 giorni dopo l'ultimo utilizzo. Per mitigare gli errori nella cache e aumentare la frequenza degli accessi alla cache, CodeCatalyst supporta le cache di fallback. Le cache di fallback sono cache alternative e offrono l'opportunità di ripristinare cache parziali, che possono essere una versione precedente di una cache. Una cache viene ripristinata cercando innanzitutto una corrispondenza nel nome della proprietà e, se non viene trovata, valuta `FileCaching RestoreKeys`. Se si verifica un errore nella cache sia per il

nome della proprietà che per tutto `RestoreKeys`, il flusso di lavoro continuerà a funzionare, poiché la memorizzazione nella cache è la soluzione migliore e non è garantita.

Creazione di una cache

Puoi usare le seguenti istruzioni per aggiungere una cache al tuo flusso di lavoro.

Visual

Per aggiungere una cache utilizzando l'editor visuale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione in cui desideri aggiungere la cache.
8. Scegliere Configuration (Configurazione).
9. In Memorizzazione nella cache dei file - opzionale, scegli Aggiungi cache e inserisci le informazioni nei campi, come segue:

Chiave

Specificate il nome della proprietà della cache principale. I nomi delle proprietà della cache devono essere univoci all'interno del flusso di lavoro. Ogni azione può contenere fino a cinque immissioni `FileCaching`.

Path

Specificate il percorso associato per la cache.

Chiavi di ripristino: opzionale

Specificare la chiave di ripristino da utilizzare come fallback quando non è possibile trovare la proprietà della cache principale. I nomi delle chiavi di ripristino devono essere univoci all'interno del flusso di lavoro. Ogni cache può contenere fino a cinque voci `RestoreKeys`.

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit, quindi scegliete nuovamente Commit.

YAML

Per aggiungere una cache utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione del flusso di lavoro, aggiungi un codice simile al seguente:

```
action-name:
  Configuration:
    Steps: ...
  Caching:
    FileCaching:
      key-name:
        Path: file-path
        # # Specify any additional fallback caches
        # RestoreKeys:
        # - restore-key
```

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Vincoli di memorizzazione nella cache dei file

Di seguito sono riportati i vincoli per il nome della proprietà e: RestoreKeys

- I nomi devono essere univoci all'interno di un flusso di lavoro.
- I nomi sono limitati a caratteri alfanumerici (A-Z, a-z, 0-9), trattini (-) e caratteri di sottolineatura (_).
- I nomi possono contenere fino a 180 caratteri.
- Ogni azione può contenere fino a cinque cache. `FileCaching`
- Ogni cache può contenere fino a cinque voci. `RestoreKeys`

Di seguito sono riportati i vincoli per i percorsi:

- Gli asterischi (*) non sono consentiti.
- I percorsi possono contenere fino a 255 caratteri.

Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro

IN Amazon CodeCatalyst, puoi visualizzare lo stato e i dettagli di un singolo flusso di lavoro eseguito o di più esecuzioni contemporaneamente.

Per un elenco dei possibili stati di esecuzione, consulta [Stati di esecuzione del flusso di lavoro](#).

Note

È inoltre possibile visualizzare lo stato del flusso di lavoro, che è diverso dallo stato di esecuzione del flusso di lavoro. Per ulteriori informazioni, consulta [Visualizzazione dello stato di un flusso di lavoro](#).

Per ulteriori informazioni sulle esecuzioni del flusso di lavoro, vedere [Esecuzione di un workflow](#).

Argomenti

- [Visualizzazione dello stato e dei dettagli di una singola esecuzione](#)
- [Visualizzazione dello stato e dei dettagli di tutte le esecuzioni del progetto](#)
- [Visualizzazione dello stato e dei dettagli di tutte le esecuzioni di uno specifico workflow](#)
- [Visualizzazione delle esecuzioni di un flusso di lavoro nel diagramma del flusso di lavoro](#)

Visualizzazione dello stato e dei dettagli di una singola esecuzione

Potresti voler visualizzare lo stato e i dettagli di un singolo flusso di lavoro eseguito per verificare se ha avuto esito positivo, per vedere a che ora è stato completato o per vedere chi o cosa l'ha avviato.

Per visualizzare lo stato e i dettagli di una singola esecuzione

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Sotto il nome del flusso di lavoro, scegli Esegue.
6. Nella cronologia di esecuzione, nella colonna Run ID, scegli una corsa. Ad esempio Run-95a4d.
7. Sotto il nome della corsa, esegui una delle seguenti operazioni:
 - Visuale per visualizzare un diagramma del flusso di lavoro che mostra le azioni eseguite dal flusso di lavoro e il relativo stato (vedi [Stati di esecuzione del flusso di lavoro](#)). Questa vista mostra anche il repository e il ramo di origine utilizzati durante l'esecuzione.

Nel diagramma del flusso di lavoro, scegli un'azione per visualizzare dettagli come log, report e output generati dall'azione durante l'esecuzione. Le informazioni mostrate dipendono dal tipo di azione selezionato. Per ulteriori informazioni sulla visualizzazione dei log di compilazione o distribuzione, consulta [Visualizzazione dei risultati di un'azione di compilazione](#) o [Visualizzazione dei log di distribuzione](#).

- YAML per visualizzare il file di definizione del flusso di lavoro utilizzato per l'esecuzione.
- Artefatti per visualizzare gli artefatti prodotti dal flusso di lavoro in esecuzione. Per ulteriori informazioni sugli artefatti, vedi [Condivisione di artefatti e file tra azioni](#).
- Rapporti per visualizzare i rapporti sui test e altri tipi di report prodotti dal flusso di lavoro eseguiti. Per ulteriori informazioni sui report, vedere [Tipi di report sulla qualità](#).
- Variabili per visualizzare le variabili di output prodotte dall'esecuzione del flusso di lavoro. Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

 Note

Se il flusso di lavoro principale dell'esecuzione è stato eliminato, nella parte superiore della pagina dei dettagli dell'esecuzione viene visualizzato un messaggio che indica questo fatto.

Visualizzazione dello stato e dei dettagli di tutte le esecuzioni del progetto

Potresti voler visualizzare lo stato e i dettagli di tutti i flussi di lavoro eseguiti all'interno del tuo progetto, capire quanta attività di flusso di lavoro è in corso nel tuo progetto e conoscere lo stato generale dei tuoi flussi di lavoro.

Per visualizzare lo stato e i dettagli di tutte le esecuzioni del progetto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. In Flussi di lavoro, scegli Esecuzioni.

Vengono visualizzate tutte le esecuzioni, per tutti i flussi di lavoro, in tutte le filiali, in tutti i repository del progetto.

La pagina include le seguenti colonne:

- Run ID: l'identificatore univoco della corsa. Scegli il link Run ID per visualizzare informazioni dettagliate sulla corsa.
- Stato: lo stato di elaborazione del workflow in esecuzione. Per ulteriori informazioni sugli stati di esecuzione, vedere [Stati di esecuzione del flusso di lavoro](#).
- Trigger: la persona, il commit, la pull request (PR) o la pianificazione che ha avviato l'esecuzione del flusso di lavoro. Per ulteriori informazioni, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Flusso di lavoro: il nome del flusso di lavoro per il quale è stata avviata un'esecuzione e l'archivio di origine e il ramo in cui risiede il file di definizione del flusso di lavoro. Potrebbe essere necessario espandere la larghezza della colonna per visualizzare queste informazioni.

 Note

Se questa colonna è impostata su Non disponibile, in genere è perché il flusso di lavoro associato è stato eliminato o spostato.

- Ora di inizio: l'ora in cui è iniziata l'esecuzione del flusso di lavoro.
- Durata: il tempo impiegato per l'elaborazione dell'esecuzione del flusso di lavoro. Durate molto lunghe o molto brevi potrebbero indicare problemi.
- Ora di fine: l'ora in cui è terminata l'esecuzione del flusso di lavoro.

Visualizzazione dello stato e dei dettagli di tutte le esecuzioni di uno specifico workflow

Potresti voler visualizzare lo stato e i dettagli di tutte le esecuzioni associate a un flusso di lavoro specifico per vedere se alcune esecuzioni stanno creando colli di bottiglia all'interno del flusso di lavoro o per vedere quali esecuzioni sono attualmente in corso o sono state completate.

Per visualizzare lo stato e i dettagli di tutte le esecuzioni di uno specifico workflow

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Sotto il nome del flusso di lavoro, scegli Esegui.

Vengono visualizzate le esecuzioni associate al flusso di lavoro scelto.

La pagina è divisa in due sezioni:

- Esecuzioni attive: visualizza le esecuzioni in corso. Queste esecuzioni si svolgeranno in uno dei seguenti stati: In corso.
- Cronologia delle esecuzioni: visualizza le esecuzioni completate (ovvero non in corso).

Per ulteriori informazioni sugli stati di esecuzione, vedere [Stati di esecuzione del flusso di lavoro](#).

Visualizzazione delle esecuzioni di un flusso di lavoro nel diagramma del flusso di lavoro

È possibile visualizzare lo stato di tutte le esecuzioni di un flusso di lavoro man mano che procedono insieme nel flusso di lavoro. Le esecuzioni vengono visualizzate all'interno del diagramma del flusso di lavoro (anziché in una visualizzazione a elenco). Ciò offre una rappresentazione visiva di quali esecuzioni vengono elaborate, con quali azioni e quali esecuzioni sono in attesa in coda.

Per visualizzare lo stato di più esecuzioni man mano che procedono insieme all'interno di un flusso di lavoro

Note

Questa procedura si applica solo se il flusso di lavoro utilizza la modalità di esecuzione in coda o sostituita. Per ulteriori informazioni, consulta [Configurazione del comportamento di accodamento delle esecuzioni](#).

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/) CodeCatalyst .
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.

Note

Assicurati di visualizzare una pagina del flusso di lavoro e non una pagina di esecuzione.

5. Scegli la scheda Stato più recente in alto a sinistra.

Viene visualizzato un diagramma del flusso di lavoro.

6. Esamina il diagramma del flusso di lavoro. Il diagramma mostra tutte le esecuzioni attualmente in corso all'interno del flusso di lavoro e le ultime esecuzioni terminate. Più specificamente:

- Le esecuzioni che appaiono in alto, prima di Sources, sono in coda e attendono di iniziare.
- Le esecuzioni che compaiono tra le azioni vengono messe in coda e attendono di essere elaborate dall'azione successiva.

- Le esecuzioni che compaiono all'interno di un'azione sono 1. attualmente in fase di elaborazione da parte dell'azione, 2. hanno terminato l'elaborazione da parte dell'azione o 3. non sono state elaborate dall'azione (in genere perché un'azione precedente non è riuscita).

Configurazione delle azioni del flusso di lavoro

Un'azione è l'elemento costitutivo principale di un flusso di lavoro e definisce un'unità logica di lavoro, o attività, da eseguire durante l'esecuzione di un flusso di lavoro. In genere, un flusso di lavoro include più azioni eseguite in sequenza o in parallelo a seconda di come le hai configurate.

Argomenti

- [Tipi di operazione](#)
- [Aggiungere un'azione a un flusso di lavoro](#)
- [Rimuovere un'azione da un flusso di lavoro](#)
- [Sviluppo di un'azione personalizzata](#)
- [Raggruppamento delle azioni in gruppi di azione](#)
- [Azioni di sequenziamento](#)
- [Condivisione di artefatti e file tra azioni](#)
- [Specificare la versione dell'azione da utilizzare](#)
- [Elenco delle versioni action disponibili](#)
- [Visualizzazione del codice sorgente di un'azione](#)
- [Integrazione con GitHub Actions](#)

Tipi di operazione

All'interno di un CodeCatalyst flusso di lavoro Amazon, puoi utilizzare i seguenti tipi di azioni.

Tipi di operazione

- [CodeCatalyst azioni](#)
- [CodeCatalyst Azioni di laboratorio](#)
- [GitHub Azioni](#)
- [Operazioni di terze parti](#)

CodeCatalyst azioni

Un'CodeCatalyst azione è un'azione creata, gestita e completamente supportata dal team di CodeCatalyst sviluppo.

Esistono CodeCatalyst azioni per la creazione, il test e la distribuzione di applicazioni, nonché per l'esecuzione di varie attività, come l'invocazione di una funzione. AWS Lambda

Sono disponibili le seguenti CodeCatalyst azioni:

- **Compila**

Questa azione crea i tuoi artefatti ed esegue i tuoi unit test in un contenitore Docker. Per ulteriori informazioni, consulta [Aggiungere l'azione di compilazione](#).

- **Test**

Questa azione esegue test di integrazione e di sistema rispetto all'applicazione o agli artefatti. Per ulteriori informazioni, consulta [Aggiungere l'azione di test](#).

- **Pubblicazione su Amazon S3**

Questa azione copia gli artefatti dell'applicazione in un bucket Amazon S3. Per ulteriori informazioni, consulta [Pubblicazione di file su Amazon S3 con un flusso di lavoro](#).

- **AWS CDK bootstrap**

Questa azione fornisce le risorse necessarie per distribuire l'AWS CDK app CDK. Per ulteriori informazioni, consulta [Avvio di un'AWS CDK app con un flusso di lavoro](#).

- **AWS CDK distribuire**

Questa azione sintetizza e distribuisce un'app. AWS Cloud Development Kit (AWS CDK) Per ulteriori informazioni, consulta [Implementazione di un'AWS CDK app con un flusso di lavoro](#).

- **AWS Lambda invocare**

Questa azione richiama una funzione. AWS Lambda Per ulteriori informazioni, consulta [Richiamo di una funzione Lambda utilizzando un flusso di lavoro](#).

- **GitHub Azioni**

Questa azione consente di CodeCatalyst eseguire GitHub azioni all'interno di un CodeCatalyst flusso di lavoro. Per ulteriori informazioni, consulta [Richiamo di una funzione Lambda utilizzando un flusso di lavoro](#).

- Distribuisci lo stack CloudFormation

Questa azione distribuisce CloudFormation gli stack. Per ulteriori informazioni, consulta [Implementazione di uno stack CloudFormation](#).

- Esegui la distribuzione su Amazon ECS

Questa azione registra una definizione di attività Amazon ECS e la distribuisce su un servizio Amazon ECS. Per ulteriori informazioni, consulta [Distribuzione su Amazon ECS con un flusso di lavoro](#).

- Esegui la distribuzione nel cluster Kubernetes

Questa azione distribuisce un'applicazione in un cluster Kubernetes. Per ulteriori informazioni, consulta [Implementazione su Amazon EKS con un flusso di lavoro](#).

- Renderizza la definizione delle attività di Amazon ECS

Questa azione inserisce un URI dell'immagine del contenitore in un file JSON di definizione delle attività di Amazon ECS, creando un nuovo file di definizione delle attività. Per ulteriori informazioni, consulta [Modifica della definizione di un'attività Amazon ECS](#).

La documentazione relativa CodeCatalyst alle azioni è disponibile in questa guida e nel file readme di ogni azione.

Per informazioni sulle CodeCatalyst azioni disponibili e su come aggiungerne una a un flusso di lavoro, consulta [Aggiungere un'azione a un flusso di lavoro](#).

CodeCatalyst Azioni di laboratorio

Un'azione CodeCatalyst Labs è un'azione che fa parte di Amazon CodeCatalyst Labs, un banco di prova per applicazioni sperimentali. CodeCatalyst Le azioni di laboratorio sono state sviluppate per mostrare le integrazioni con i servizi. AWS

Sono disponibili le seguenti azioni CodeCatalyst Labs:

- Distribuisci su Hosting AWS Amplify

Questa azione distribuisce un'applicazione su Amplify Hosting.

- Distribuisci su AWS App Runner

Questa azione distribuisce l'immagine più recente in un archivio di immagini di origine su App Runner.

- Esegui la distribuzione su Amazon CloudFront e Amazon S3

Questa azione distribuisce un'applicazione su Amazon S3. CloudFront

- Esegui la distribuzione con AWS SAM

Questa azione distribuisce la tua applicazione serverless con AWS Serverless Application Model ().AWS SAM

- Invalidare Amazon Cache CloudFront

Questa azione invalida una CloudFront cache per un determinato set di percorsi.

- Webhook in uscita

Questa azione consente agli utenti di inviare messaggi all'interno di un flusso di lavoro a un server Web arbitrario utilizzando una richiesta HTTPS.

- Pubblica su AWS CodeArtifact

Questa azione pubblica i pacchetti in un CodeArtifact repository.

- Pubblica su Amazon SNS

Questa azione consente agli utenti di integrarsi con Amazon SNS creando un argomento, pubblicandolo su un argomento o iscrivendosi a un argomento.

- Esegui il push su Amazon ECR

Questa azione crea e pubblica un'immagine Docker in un repository Amazon Elastic Container Registry (Amazon ECR).

- Scansione con Amazon CodeGuru Security

Questa azione crea un archivio zip di un percorso di codice configurato e utilizza CodeGuru Security per eseguire una scansione del codice.

- Terraform Community Edition

Questa azione esegue Terraform Community Edition plan e apply le operazioni.

La documentazione per le azioni CodeCatalyst di Labs è disponibile nel readme di ogni azione.

Per informazioni sull'aggiunta di un'azione CodeCatalyst Labs a un flusso di lavoro e sulla visualizzazione del relativo readme, consulta. [Aggiungere un'azione a un flusso di lavoro](#)

GitHub Azioni

Un'GitHub azione è molto simile a un'[CodeCatalyst azione](#), tranne per il fatto che è stata sviluppata per essere utilizzata con i GitHub flussi di lavoro. Per informazioni dettagliate sulle GitHub azioni, consulta la documentazione relativa [GitHub alle azioni](#).

È possibile utilizzare GitHub le azioni insieme alle CodeCatalyst azioni native in un CodeCatalyst flusso di lavoro.

Per comodità, la CodeCatalyst console fornisce l'accesso a diverse GitHub azioni popolari. Puoi anche utilizzare qualsiasi GitHub azione elencata nel [GitHub Marketplace](#) (soggetta ad alcune limitazioni).

La documentazione relativa GitHub alle azioni è disponibile nel file readme di ogni azione.

Per ulteriori informazioni, consulta [Integrazione con GitHub Actions](#).

Operazioni di terze parti

Un'azione di terze parti è un'azione creata da un fornitore terzo e resa disponibile nella CodeCatalyst console. Esempi di azioni di terze parti includono le azioni Mend SCA e SonarCloud Scan, create rispettivamente da Mend e Sonar.

La documentazione per le azioni di terze parti è disponibile nel file readme di ogni azione. Documentazione aggiuntiva potrebbe essere fornita anche dal fornitore terzo.

Per informazioni sull'aggiunta di un'azione di terze parti a un flusso di lavoro e sulla visualizzazione del relativo file readme, consulta. [Aggiungere un'azione a un flusso di lavoro](#)

Aggiungere un'azione a un flusso di lavoro

Utilizza le seguenti istruzioni per aggiungere un'azione a un flusso di lavoro e quindi configurarla.

Per aggiungere e configurare un'azione

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.

3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. In alto a sinistra, scegli + Azioni, viene visualizzato il catalogo Azioni.
7. Nell'elenco a discesa, esegui una delle seguenti operazioni:
 - Scegli Amazon CodeCatalyst per visualizzare azioni [CodeCatalyst](#), [CodeCatalyst Labs](#) o [di terze parti](#).
 - CodeCatalyst le azioni hanno un' AWS etichetta per.
 - CodeCatalyst Le azioni Labs hanno l' CodeCatalyst etichetta by Labs.
 - Le azioni di terze parti hanno un'*vendor* etichetta per, dove *vendor* è il nome del fornitore terzo.
 - Scegli GitHubdi visualizzare un [elenco curato di GitHub](#) azioni.
8. Nel catalogo delle azioni, cerca un'azione, quindi esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al tuo flusso di lavoro.
 - Scegli il nome dell'azione per visualizzarne il file readme.
9. Configura l'azione. Scegli Visual per usare l'editor visuale o YAML per usare l'editor YAML. Per istruzioni dettagliate, consulta i seguenti collegamenti.

Per istruzioni sull'aggiunta di [CodeCatalystazioni](#), consulta:

- [Aggiungere l'azione di compilazione](#)
- [Aggiungere l'azione di test](#)
- [Aggiungere l'azione «Deploy to Amazon ECS»](#)
- [Aggiungere l'azione «Deploy to Kubernetes cluster»](#)
- [Aggiungere l'azione «Deploy stack» CloudFormation](#)
- [Aggiungere l'azione 'AWS CDK deploy'](#)
- [Aggiungere l'azione 'AWS CDK bootstrap'](#)
- [Aggiungere l'azione «Amazon S3 publish»](#)
- [Aggiungere l'AWS Lambda azione «invoca»](#)
- [Aggiungere l'azione «Render Amazon ECS task definition»](#)

Per istruzioni sull'aggiunta di [azioni CodeCatalyst Labs](#), consulta:

- Il file readme è l'azione. Puoi trovare il file readme scegliendo il nome dell'azione nel catalogo delle azioni.

Per istruzioni sull'aggiunta di [GitHub azioni](#), consulta:

- [Integrazione con GitHub Actions](#)

Per istruzioni sull'aggiunta [di azioni di terze parti](#), consulta:

- Il file readme è l'azione. Puoi trovare il file readme scegliendo il nome dell'azione nel catalogo delle azioni.

10. (Facoltativo) Scegli Convalida per assicurarti che il codice YAML sia valido.

11. Scegli Conferma per confermare le modifiche.

Rimuovere un'azione da un flusso di lavoro

Utilizza le seguenti istruzioni per rimuovere un'azione da un flusso di lavoro.

Visual

Per rimuovere un'azione utilizzando l'editor visuale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, nell'azione che desideri rimuovere, scegli l'icona con i puntini di sospensione verticali e scegli Rimuovi.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per rimuovere un'azione utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Trova la sezione di YAML che contiene l'azione che desideri rimuovere.

Seleziona la sezione e premi il tasto Elimina sulla tastiera.

8. (Facoltativo) Scegli Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Sviluppo di un'azione personalizzata

Puoi sviluppare un'azione personalizzata da utilizzare nei tuoi flussi di lavoro utilizzando l'CodeCatalyst Action Development Kit (ADK). Puoi quindi pubblicare l'azione nel catalogo delle CodeCatalyst azioni, in modo che altri CodeCatalyst utenti possano visualizzarla e utilizzarla nei propri flussi di lavoro.

Per sviluppare, testare e pubblicare un'azione (attività di alto livello)

1. Installa gli strumenti e i pacchetti necessari per sviluppare un'azione.
2. Crea un CodeCatalyst repository per archiviare il codice d'azione.
3. Inizializza l'azione. Questo stabilisce i file sorgente richiesti dall'azione, incluso un file di definizione dell'azione (`action.yml`) che puoi aggiornare con il tuo codice.

4. Avvia il codice d'azione per ottenere gli strumenti e le librerie necessari per creare, testare e rilasciare il progetto d'azione.
5. Crea l'azione sul tuo computer locale e invia le modifiche al tuo CodeCatalyst repository.
6. Testa l'azione con i test unitari a livello locale ed esegui il flusso di lavoro generato da ADK in CodeCatalyst
7. Pubblica l'azione nel catalogo CodeCatalyst delle azioni scegliendo il pulsante Pubblica nella CodeCatalyst console.

Per i passaggi dettagliati, consulta la [Amazon CodeCatalyst Action Development Kit Development Kit Development Guide](#).

Raggruppamento delle azioni in gruppi di azione

Un gruppo di azioni contiene una o più azioni. Il raggruppamento delle azioni in gruppi di azioni consente di mantenere organizzato il flusso di lavoro e consente inoltre di configurare le dipendenze tra diversi gruppi.

Note

Non è possibile annidare i gruppi di azioni all'interno di altri gruppi di azioni o azioni.

Argomenti

- [Definizione di un gruppo di azioni](#)
- [Esempio: definizione di due gruppi di azioni](#)

Definizione di un gruppo di azioni

Utilizzate le seguenti istruzioni per definire un gruppo di CodeCatalyst azioni.

Visual

Non disponibile. Scegli YAML per visualizzare le istruzioni YAML.

YAML

Per definire un gruppo

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/) .

2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In Actions, aggiungi un codice simile al seguente:

```
Actions:
  action-group-name:
    Actions:
      action-1:
        Identifier: aws/build@v1
        Configuration:
          ...
      action-2:
        Identifier: aws/build@v1
        Configuration:
          ...
```

Per un altro esempio, consulta [Esempio: definizione di due gruppi di azioni](#). Per ulteriori informazioni, vedere la descrizione [Azioni](#) della `action-group-name` proprietà nel [Definizione YAML del flusso di lavoro](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Esempio: definizione di due gruppi di azioni

L'esempio seguente mostra come definire due gruppi di CodeCatalyst azioni Amazon: BuildAndTest e Deploy. Il BuildAndTest gruppo include due azioni (BuildTest) e il Deploy gruppo include anche due azioni (DeployCloudFormationStackDeployToECS).

```
Actions:
  BuildAndTest: # Action group 1
    Actions:
```

```
Build:
  Identifier: aws/build@v1
  Configuration:
    ...
Test:
  Identifier: aws/managed-test@v1
  Configuration:
    ...
Deploy: #Action group 2
Actions:
  DeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Configuration:
      ...
  DeployToECS:
    Identifier: aws/ecs-deploy@v1
    Configuration:
      ...
```

Azioni di sequenziamento

Per impostazione predefinita, quando aggiungi azioni a un flusso di lavoro, queste vengono aggiunte fianco a fianco nell'[editor visivo](#). Ciò significa che le azioni verranno eseguite in parallelo quando si avvia l'esecuzione di un flusso di lavoro. Se desideri che le azioni vengano eseguite in ordine sequenziale (e appaiano verticalmente nell'editor visivo), devi impostare le dipendenze tra di esse. Ad esempio, è possibile impostare un'azione in modo che dipenda dall'azione in modo che l'Buildazione di test venga eseguita dopo l'azione di creazione.

È possibile impostare dipendenze tra azioni e gruppi di azioni. Puoi anche configurare one-to-many le dipendenze in modo che un'azione dipenda da diverse altre per iniziare. Consulta le linee guida riportate in [Impostazione delle dipendenze tra le azioni](#) per assicurarti che la configurazione delle dipendenze sia conforme alla sintassi YAML del flusso di lavoro.

Argomenti

- [Esempi di come configurare le dipendenze tra le azioni](#)
- [Impostazione delle dipendenze tra le azioni](#)

Esempi di come configurare le dipendenze tra le azioni

Gli esempi seguenti mostrano come configurare le dipendenze tra azioni e gruppi nel file di definizione del flusso di lavoro.

Argomenti

- [Esempio: configurazione di una dipendenza semplice](#)
- [Esempio: configurazione di un gruppo di azioni in modo che dipenda da un'azione](#)
- [Esempio: configurazione di un gruppo di azioni in modo che dipenda da un altro gruppo di azioni](#)
- [Esempio: configurazione di un gruppo di azioni in modo che dipenda da più azioni](#)

Esempio: configurazione di una dipendenza semplice

L'esempio seguente mostra come configurare un'azione in modo che dipenda dall'azione di build che utilizza la `DependsOn` proprietà.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      ...
  Test:
    DependsOn:
    - Build
    Identifier: aws/managed-test@v1
    Configuration:
      ...
```

Esempio: configurazione di un gruppo di azioni in modo che dipenda da un'azione

L'esempio seguente mostra come configurare un gruppo di `DeployGroup` azioni in modo che dipenda dall'azione di prima. Notate che l'azione e il gruppo di azioni sono allo stesso livello.

```
Actions:
  FirstAction: #An action outside an action group
    Identifier: aws/github-actions-runner@v1
    Configuration:
      ...
  DeployGroup: #An action group containing two actions
    DependsOn:
    - FirstAction
    Actions:
      DeployAction1:
        ...
      DeployAction2:
```

...

Esempio: configurazione di un gruppo di azioni in modo che dipenda da un altro gruppo di azioni

L'esempio seguente mostra come configurare un gruppo di `DeployGroup` azioni in modo che dipenda dal gruppo di `BuildAndTestGroup` azioni. Notate che i gruppi di azione si trovano allo stesso livello.

```

Actions:
  BuildAndTestGroup: # Action group 1
    Actions:
      BuildAction:
      ...
      TestAction:
      ...
  DeployGroup: #Action group 2
    DependsOn:
      - BuildAndTestGroup
    Actions:
      DeployAction1:
      ...
      DeployAction2:
      ...

```

Esempio: configurazione di un gruppo di azioni in modo che dipenda da più azioni

L'esempio seguente mostra come configurare un gruppo di `DeployGroup` azioni in modo che dipenda dall'`FirstAction` azione, dall'`SecondAction` azione e dal gruppo di `BuildAndTestGroup` azioni. Notate che `DeployGroup` è allo stesso livello di `FirstAction`, `SecondAction`, e `BuildAndTestGroup`.

```

Actions:
  FirstAction: #An action outside an action group
  ...
  SecondAction: #Another action
  ...
  BuildAndTestGroup: #Action group 1
    Actions:
      Build:
      ...
      Test:

```

```
...
DeployGroup: #Action group 2
  DependsOn:
    - FirstAction
    - SecondAction
    - BuildAndTestGroup
  Actions:
    DeployAction1:
      ...
    DeployAction2:
      ...
```

Impostazione delle dipendenze tra le azioni

Utilizza le seguenti istruzioni per impostare le dipendenze tra le azioni in un flusso di lavoro.

Quando configuri le dipendenze, segui queste linee guida:

- Se un'azione è all'interno di un gruppo, tale azione può dipendere solo da altre azioni all'interno dello stesso gruppo.
- Le azioni e i gruppi di azioni possono dipendere da altre azioni e gruppi di azioni allo stesso livello nella gerarchia YAML, ma non a un livello diverso.

Visual

Per impostare le dipendenze utilizzando l'editor visuale

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione che dipenderà da un'altra azione.
8. Scegli la scheda Input.

9. In Depends on - opzionale, procedi come segue:

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente affinché questa azione venga eseguita.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

10. (Facoltativo) Scegli Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per impostare le dipendenze utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione che dipenderà da un'altra, aggiungi un codice simile al seguente:

```
action-name:
  DependsOn:
    - action-1
```

Per ulteriori esempi, consulta [Esempi di come configurare le dipendenze tra le azioni](#).

Per linee guida generali, vedi [Impostazione delle dipendenze tra le azioni](#). Per ulteriori informazioni, consulta la descrizione della DependsOn proprietà nella sezione [Definizione YAML del flusso di lavoro](#) Per le tue azioni.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Condivisione di artefatti e file tra azioni

Un artefatto è l'output di un'azione del flusso di lavoro e in genere è costituito da una cartella o da un archivio di file. Gli artefatti sono importanti perché consentono di condividere file e informazioni tra le azioni.

Ad esempio, potresti avere un'azione di compilazione che genera un *sam-template.yml* file, ma desideri un'azione di distribuzione per utilizzarla. In questo scenario, si utilizzerebbe un artefatto per consentire all'azione di compilazione di condividere il *sam-template.yml* file con l'azione di distribuzione. Il codice potrebbe essere simile al seguente:

```
Actions:
  BuildAction:
    Identifier: aws/build@v1
    Steps:
      - Run: sam package --output-template-file sam-template.yml
    Outputs:
      Artifacts:
        - Name: MYARTIFACT
          Files:
            - sam-template.yml
  DeployAction:
    Identifier: aws/cfn-deploy@v1
    Inputs:
      Artifacts:
        - MYARTIFACT
    Configuration:
      template: sam-template.yml
```

Nel codice precedente, l'azione build (BuildAction) genera un *sam-template.yml* file e quindi lo aggiunge a un artefatto di output chiamato MYARTIFACT. Una successiva deploy action (DeployAction) specifica MYARTIFACT come input per l'accesso al file *sam-template.yml*.

Argomenti

- [Posso condividere gli artefatti senza specificarli come output e input?](#)
- [Posso condividere artefatti tra flussi di lavoro?](#)
- [Esempi di manufatti](#)
- [Definizione di un artefatto di output](#)
- [Definizione di un artefatto di input](#)

- [Riferimento ai file in un artefatto](#)
- [Scaricamento di artefatti](#)

Posso condividere gli artefatti senza specificarli come output e input?

Sì, puoi condividere artefatti tra azioni senza specificarli nelle sezioni `Outputs` e `Inputs` del codice YAML delle tue azioni. Per fare ciò, devi attivare la condivisione del calcolo. Per ulteriori informazioni sulla condivisione del calcolo e su come specificare gli artefatti quando è attivata, consulta [Condivisione dell'elaborazione tra le azioni](#)

Note

Sebbene la funzionalità di condivisione del calcolo consenta di semplificare il codice YAML del flusso di lavoro eliminando la necessità delle sezioni `Outputs` e `Inputs`, è necessario conoscere alcune limitazioni prima di attivarla. Per informazioni su queste limitazioni, consulta [Considerazioni sulla condivisione del calcolo](#)

Posso condividere artefatti tra flussi di lavoro?

No, non puoi condividere artefatti tra flussi di lavoro diversi; tuttavia, puoi condividere artefatti tra azioni all'interno dello stesso flusso di lavoro.

Esempi di manufatti

I seguenti esempi mostrano come generare, inserire e fare riferimento agli artefatti nel file di definizione del CodeCatalyst flusso di lavoro di Amazon.

Argomenti

- [Esempio: output di un artefatto](#)
- [Esempio: inserimento di un artefatto generato da un'altra azione](#)
- [Esempio: riferimento a file in più artefatti](#)
- [Esempio: fare riferimento a un file in un singolo artefatto](#)
- [Esempio: fare riferimento a un file in un artefatto quando è presente un `WorkflowSource`](#)
- [Esempio: fare riferimento a un file in un artefatto quando è presente un gruppo di azioni](#)

Esempio: output di un artefatto

L'esempio seguente mostra come generare un artefatto che include due file.jar.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ARTIFACT1
          Files:
            - build-output/file1.jar
            - build-output/file2.jar
```

Esempio: inserimento di un artefatto generato da un'altra azione

L'esempio seguente mostra come generare un artefatto chiamato e ARTIFACT4 inserirlo inBuildActionA. BuildActionB

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ARTIFACT4
          Files:
            - build-output/file1.jar
            - build-output/file2.jar
  BuildActionB:
    Identifier: aws/build@v1
    Inputs:
      Artifacts:
        - ARTIFACT4
    Configuration:
```

Esempio: riferimento a file in più artefatti

L'esempio seguente mostra come generare due artefatti denominati ART5 e ART6 in BuildActionC e quindi fare riferimento a due file denominati file5.txt (in artifactART5) e (in artifact) in file6.txt (under). ART6 BuildActionD Steps

 Note

Per ulteriori informazioni sulla referenziazione dei file, vedere. [Riferimento ai file in un artefatto](#)

 Note

Sebbene l'esempio mostri il `$CATALYST_SOURCE_DIR_ART5` prefisso utilizzato, è possibile ometterlo. Questo perché ART5 è l'input principale. Per ulteriori informazioni sull'input principale, consulta [Riferimento ai file in un artefatto](#).

Actions:

BuildActionC:

Identifier: aws/build@v1

Outputs:

Artifacts:

- Name: ART5

Files:

- build-output/file5.txt

- Name: ART6

Files:

- build-output/file6.txt

BuildActionD:

Identifier: aws/build@v1

Inputs:

Artifacts:

- ART5

- ART6

Configuration:

Steps:

- run: cd \$CATALYST_SOURCE_DIR_ART5/build-output && cat file5.txt

- run: cd \$CATALYST_SOURCE_DIR_ART6/build-output && cat file6.txt

Esempio: fare riferimento a un file in un singolo artefatto

L'esempio seguente mostra come generare un elemento denominato ART7 in e quindi fare riferimento file7.txt (nell'BuildActionEartefattoART7) in (sotto). BuildActionF Steps

Notate come il riferimento non richieda il `$CATALYST_SOURCE_DIR_ artifact-name` prefisso davanti alla build-output directory come accadeva in. [Esempio: riferimento a file in più artefatti](#) Questo perché c'è solo un elemento specificato in `Inputs`.

Note

Per ulteriori informazioni sui file di riferimento, vedere [Riferimento ai file in un artefatto](#).

Actions:

BuildActionE:

Identifier: aws/build@v1

Outputs:

Artifacts:

- Name: ART7

Files:

- build-output/file7.txt

BuildActionF:

Identifier: aws/build@v1

Inputs:

Artifacts:

- ART7

Configuration:

Steps:

- run: cd build-output && cat file7.txt

Esempio: fare riferimento a un file in un artefatto quando è presente un `WorkflowSource`

L'esempio seguente mostra come generare un elemento denominato ART8 in e quindi fare riferimento `file8.txt` (nell'`BuildActionG` artefatto ART8) in (sotto). `BuildActionH` Steps

Notate come il riferimento richieda il `$CATALYST_SOURCE_DIR_ artifact-name` prefisso, come accadeva in. [Esempio: riferimento a file in più artefatti](#) Questo perché ci sono più elementi specificati in `Inputs` (una fonte e un artefatto), quindi è necessario il prefisso per indicare dove cercare il file.

Note

Per ulteriori informazioni sulla referenziazione dei file, vedere. [Riferimento ai file in un artefatto](#)

```
Actions:
  BuildActionG:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ART8
          Files:
            - build-output/file8.txt
  BuildActionH:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
      Artifacts:
        - ART8
    Configuration:
      Steps:
        - run: cd $CATALYST_SOURCE_DIR_ART8/build-output && cat file8.txt
```

Esempio: fare riferimento a un file in un artefatto quando è presente un gruppo di azioni

L'esempio seguente mostra come generare un artefatto denominato ART9 in e quindi fare riferimento file9.txt (in ActionGroup1 un artefatto) in. ActionI ART9 ActionJ

Per ulteriori informazioni sulla referenziazione dei file, vedere. [Riferimento ai file in un artefatto](#)

```
Actions:
  ActionGroup1:
    Actions:
      ActionI:
        Identifier: aws/build@v1
        Outputs:
          Artifacts:
            - Name: ART9
              Files:
                - build-output/file9.yml
      ActionJ:
        Identifier: aws/cfn-deploy@v1
        Inputs:
          Sources:
            - WorkflowSource
          Artifacts:
            - ART9
```

Configuration:

```
template: /artifacts/ActionGroup1@ActionJ/ART9/build-output/file9.yml
```

Definizione di un artefatto di output

Utilizza le seguenti istruzioni per definire un artefatto che desideri venga generato da CodeCatalyst un'azione Amazon. Questo artefatto diventa quindi disponibile per altre azioni da utilizzare.

Note

Non tutte le azioni supportano gli artefatti di output. Per determinare se la tua azione li supporta, segui le istruzioni dell'editor visivo che seguono e verifica se l'azione include un pulsante relativo agli artefatti di output nella scheda Output. Se sì, gli artefatti di output sono supportati.

Visual

Per definire un artefatto di output utilizzando l'editor visuale

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione che produrrà l'artefatto.
8. Seleziona la scheda Outputs (Output).
9. In Artefatti, scegli Aggiungi artefatto.
10. Scegli Aggiungi artefatto e inserisci le informazioni nei campi, come segue.

Crea il nome dell'artefatto

Specificate il nome di un artefatto generato dall'azione. I nomi degli Artifact devono essere univoci all'interno di un flusso di lavoro e sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9) e

caratteri di sottolineatura (_). Spazi, trattini (-) e altri caratteri speciali non sono consentiti. Non è possibile utilizzare le virgolette per abilitare spazi, trattini e altri caratteri speciali nei nomi degli artefatti di output.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

File prodotti da build

Specificate i file CodeCatalyst inclusi nell'artefatto generato dall'azione. Questi file vengono generati dall'azione del flusso di lavoro durante l'esecuzione e sono disponibili anche nell'archivio di origine. I percorsi dei file possono risiedere in un repository di origine o in un elemento di un'azione precedente e sono relativi al repository di origine o alla radice dell'elemento. È possibile utilizzare modelli a globo per specificare i percorsi. Esempi:

- Per specificare un singolo file che si trova nella radice della posizione di creazione o della posizione del repository di origine, usa `my-file.jar`
- Per specificare un singolo file in una sottodirectory, usa `directory/my-file.jar` o `directory/subdirectory/my-file.jar`
- Per specificare tutti i file, utilizzare `**/*`. Il pattern a `**` glob indica che deve corrispondere a un numero qualsiasi di sottodirectory.
- Per specificare tutti i file e le directory in una directory denominata, usa `directory "directory/**/*"` Il pattern a `**` glob indica che deve corrispondere a un numero qualsiasi di sottodirectory.
- Per specificare tutti i file in una directory denominata `directory`, ma non nessuna delle sue sottodirectory, usa `"directory/*"`

Note

Se il percorso del file include uno o più asterischi (*) o un altro carattere speciale, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere. [Linee guida e convenzioni sulla sintassi](#)

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

 Note

Potrebbe essere necessario aggiungere un prefisso al percorso del file per indicare in quale elemento o fonte trovarlo. Per ulteriori informazioni, consultare [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per definire un artefatto di output utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/) CodeCatalyst .
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione del flusso di lavoro, aggiungi un codice simile al seguente:

```
action-name:
  Outputs:
    Artifacts:
      - Name: artifact-name
        Files:
          - file-path-1
          - file-path-2
```

Per ulteriori esempi, consulta [Esempi di manufatti](#). Per ulteriori informazioni, consulta la sezione [Definizione YAML del flusso di lavoro](#) dedicata alla tua azione.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.

9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Definizione di un artefatto di input

Se desideri utilizzare un artefatto generato da un'altra CodeCatalyst azione Amazon, devi specificarlo come input per l'azione corrente. Potresti essere in grado di specificare più artefatti come input, dipende dall'azione. Per ulteriori informazioni, consulta la sezione dedicata all'azione. [Definizione YAML del flusso di lavoro](#)

Note

Non è possibile fare riferimento agli artefatti di altri flussi di lavoro.

Utilizzate le seguenti istruzioni per specificare un artefatto di un'altra azione come input per l'azione corrente.

Prerequisito

Prima di iniziare, assicuratevi di aver generato l'artefatto dall'altra azione. Per ulteriori informazioni, consulta [Definizione di un artefatto di output](#). L'emissione dell'artefatto lo rende disponibile per altre azioni da utilizzare.

Visual

Specificare un artefatto come input per un'azione (editor visivo)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegliete l'azione in cui desiderate specificare un artefatto come input.

8. Scegliete Ingressi.
9. In Artifacts, facoltativo, procedi come segue:

Specificate gli artefatti delle azioni precedenti che desiderate fornire come input per questa azione. Questi artefatti devono essere già definiti come artefatti di output nelle azioni precedenti.

Se non si specifica alcun artefatto di input, è necessario specificare almeno un repository di origine in. *action-name*/Inputs/Sources

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Note

Se l'elenco a discesa Artifacts - opzionale non è disponibile (editor visivo) o se si verificano errori durante la convalida di YAML (editor YAML), è possibile che l'azione supporti un solo input. In questo caso, prova a rimuovere l'input sorgente.

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per specificare un artefatto come input per un'azione (editor YAML)

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell'azione in cui desiderate specificare l'artefatto come input, aggiungete un codice simile al seguente:

```
action-name:  
  Inputs:  
    Artifacts:  
      - artifact-name
```

Per ulteriori esempi, consulta [Esempi di manufatti](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima del commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Riferimento ai file in un artefatto

Se hai un file che si trova all'interno di un artefatto e devi fare riferimento a questo file in una delle azioni del CodeCatalyst flusso di lavoro Amazon, completa la seguente procedura.

Note

Consulta anche [Riferimento ai file del repository di origine](#).

Visual

Non disponibile. Scegli YAML per visualizzare le istruzioni YAML.

YAML

Per fare riferimento ai file in un artefatto (editor YAML)

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/) CodeCatalyst .
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell'azione in cui desideri fare riferimento a un file, aggiungi un codice simile al seguente:

```

Actions:
  My-action:
    Inputs:
      Sources:
        - WorkflowSource
      Artifacts:
        - artifact-name
    Configuration:
      template: artifact-path/path/to/file.yml

```

Nel codice precedente, sostituisci:

- *artifact-name* con il nome del manufatto.
- *artifact-path* con un valore della tabella seguente.

Se stai aggiungendo il riferimento a...	Sostituisci <i>artifact-path</i> con...
Un'azione di compilazione o un'azione di test	<code>\$CATALYST_SOURCE_DIR_ <i>artifact-name</i> /</code>
Tutte le altre azioni	<code>\$CATALYST_SOURCE_DIR_ <i>artifact-name</i> /</code> oppure <code>/artifacts/ <i>current-action-name</i> /<i>artifact-name</i> /</code> oppure Se l'azione corrente fa parte di un gruppo di azioni : <code>/artifacts/ <i>current-action-group@current-action-name</i> /<i>artifact-name</i> /</code>

Per alcuni esempi, consulta [Esempi di manufatti](#).

Note

È possibile omettere *artifact-path* e specificare semplicemente il percorso del file relativo alla directory principale dell'artefatto se:

- L'azione in cui includi il riferimento include solo un elemento sottostante `Inputs` (ad esempio, include un elemento di input e nessuna fonte).
- Il file a cui vuoi fare riferimento si trova nell'input principale. L'input principale è `WorkflowSource`, o il primo elemento di input elencato, se non esiste.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Scaricamento di artefatti

Puoi scaricare e ispezionare gli artefatti generati dalle azioni del tuo CodeCatalyst flusso di lavoro Amazon per la risoluzione dei problemi. Esistono due tipi di artefatti che puoi scaricare:

- Artefatti di origine: un artefatto che contiene un'istantanea del contenuto del repository di origine così com'era all'avvio dell'esecuzione.
- Artefatti del flusso di lavoro: un artefatto definito nella proprietà del file di configurazione del flusso di lavoro. `Outputs`

Per scaricare gli artefatti generati dal flusso di lavoro

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.

5. Sotto il nome del flusso di lavoro, scegli Esegue.
6. Nella cronologia di esecuzione, nella colonna Run ID, scegli una corsa. Ad esempio Run-95a4d.
7. Sotto il nome della corsa, scegli Artefatti.
8. Accanto a un artefatto, scegli Scarica. Viene scaricato un file di archivio. Il nome del file è composto da sette caratteri casuali.
9. Estrai l'archivio utilizzando un'utilità di estrazione dell'archivio di tua scelta.

Specificare la versione dell'azione da utilizzare

Per impostazione predefinita, quando aggiungi un'azione a un flusso di lavoro, Amazon CodeCatalyst aggiunge la versione completa al file di definizione del flusso di lavoro utilizzando il formato:

*v***major.minor.patch**

Esempio:

```
My-Build-Action:
  Identifier: aws/build@v1.0.0
```

Puoi abbreviare la versione completa nella `Identifier` proprietà in modo che il flusso di lavoro utilizzi sempre l'ultima versione secondaria o patch dell'azione.

Ad esempio, se si specifica:

```
My-CloudFormation-Action:
  Identifier: aws/cfn-deploy@v1.0
```

... e l'ultima versione della patch è `1.0.4`, allora l'azione utilizzerà `1.0.4`. Se viene rilasciata una versione successiva, ad esempio `1.0.5`, l'azione verrà utilizzata `1.0.5`. Se viene rilasciata una versione secondaria, ad esempio `1.1.0`, l'azione continuerà a essere utilizzata `1.0.5`.

Per istruzioni dettagliate su come specificare le versioni, consultate uno dei seguenti argomenti.

Utilizza le seguenti istruzioni per indicare la versione di un'azione che desideri venga utilizzata dal flusso di lavoro. È possibile specificare la versione principale o secondaria più recente o una versione di patch specifica.

Si consiglia di utilizzare la versione secondaria o la patch più recente di un'azione.

Visual

Non disponibile. Scegli YAML per visualizzare le istruzioni YAML.

YAML

Per configurare un flusso di lavoro per utilizzare la versione più recente di un'azione o una versione di patch specifica

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Trova l'azione di cui desideri modificare la versione.
8. Trova la Identifier proprietà dell'azione e imposta la versione su una delle seguenti:
 - action-identifier @v*major*: utilizzate questa sintassi per fare in modo che il flusso di lavoro utilizzi una versione principale specifica e consentire la scelta automatica delle versioni secondarie e delle patch più recenti.
 - identificatore di azione @v. *major minor*— Utilizzate questa sintassi per fare in modo che il flusso di lavoro utilizzi una versione secondaria specifica e consentite la scelta automatica dell'ultima versione della patch.
 - identificatore di azione @v. *major minor. patch* — Utilizzate questa sintassi per fare in modo che il flusso di lavoro utilizzi una versione di patch specifica.

Note

Se non sei sicuro delle versioni disponibili, consulta [Elenco delle versioni action disponibili](#).

 Note

Non puoi omettere la versione principale.

9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Elenco delle versioni action disponibili

Utilizza le seguenti istruzioni per determinare quali versioni di un'azione sono disponibili per l'uso in un flusso di lavoro.

Visual

Per determinare quali versioni di azione sono disponibili

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Trova l'azione di cui desideri visualizzare le versioni:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - b. Scegli il nome di qualsiasi flusso di lavoro o creane uno. Per informazioni sulla creazione di un flusso di lavoro, consulta [Creazione di un flusso di lavoro](#).
 - c. Scegli Modifica.
 - d. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
 - e. Nell'elenco a discesa, scegli Amazon CodeCatalyst per visualizzare CodeCatalyst, CodeCatalyst Labs e le azioni di terze parti oppure scegli di visualizzare le azioni GitHubcurate. GitHub
 - f. Cerca un'azione e scegline il nome. Non scegliere il segno più (+).

Vengono visualizzati i dettagli sull'azione.

4. Nella finestra di dialogo dei dettagli dell'azione, in alto a destra, scegli l'elenco a discesa Versioni per visualizzare un elenco delle versioni disponibili dell'azione.

YAML

Non disponibile. Scegli «visivo» per visualizzare le istruzioni dell'editor visivo.

Visualizzazione del codice sorgente di un'azione

Puoi visualizzare il codice sorgente di un'azione per assicurarti che non contenga codice rischioso, vulnerabilità di sicurezza o altri difetti.

Utilizza le seguenti istruzioni per visualizzare il codice sorgente di un'ActionCodeCatalystazione di CodeCatalyst Labs o di terze parti.

Note

Per visualizzare il codice sorgente di un'[GitHubazione](#), vai alla pagina dell'azione nel [GitHub Marketplace](#). La pagina include un link all'archivio dell'azione, dove puoi trovare il codice sorgente dell'azione.

Note

Non è possibile visualizzare il codice sorgente delle seguenti CodeCatalyst azioni: [build](#), [test](#), [GitHub Actions](#).

Note

AWS non supporta né garantisce il codice d'azione di GitHub Actions o le azioni di terze parti.

Per visualizzare il codice sorgente di un'azione

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Trova l'azione di cui desideri visualizzare il codice:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

- b. Scegli il nome di qualsiasi flusso di lavoro o creane uno. Per informazioni sulla creazione di un flusso di lavoro, consulta [Creazione di un flusso di lavoro](#).
- c. Scegli Modifica.
- d. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
- e. Nell'elenco a discesa, scegli Amazon CodeCatalyst to view CodeCatalyst, CodeCatalyst Labs e azioni di terze parti.
- f. Cerca un'azione e scegline il nome. Non scegliere il segno più (+).

Vengono visualizzati i dettagli sull'azione.

4. Nella finestra di dialogo dei dettagli dell'azione, nella parte inferiore, scegli Scarica.

Viene visualizzata una pagina che mostra il bucket Amazon S3 in cui risiede il codice sorgente dell'azione. Per informazioni su Amazon S3, consulta [What is Amazon S3?](#) nella Guida per l'utente di Amazon Simple Storage Service.

5. Controlla il codice per assicurarti che soddisfi le tue aspettative in termini di qualità e sicurezza.

Integrazione con GitHub Actions

Un'GitHub azione è molto simile a un'[CodeCatalyst azione](#), tranne per il fatto che è stata sviluppata per essere utilizzata con i flussi GitHub di lavoro. Per informazioni dettagliate sulle GitHub azioni, consulta la documentazione relativa [GitHub alle azioni](#).

È possibile utilizzare GitHub le azioni insieme alle CodeCatalyst azioni native in un CodeCatalyst flusso di lavoro.

Esistono due modi per aggiungere un' GitHub azione a un CodeCatalyst flusso di lavoro:

- Puoi selezionare l' GitHub azione da un elenco curato nella CodeCatalyst console. Sono disponibili diverse GitHub azioni popolari. Per ulteriori informazioni, consulta [Aggiungere un'azione curata GitHub](#).
- Se l' GitHub azione che desideri utilizzare non è disponibile nella CodeCatalyst console, puoi aggiungerla utilizzando un'azione GitHub Azioni.

Un'azione GitHub Actions è un'CodeCatalyst azione che racchiude un' GitHub azione e la rende compatibile con CodeCatalyst i flussi di lavoro.

[Ecco un esempio di azione GitHub Actions che completa l'azione Super-Linter:](#) GitHub

```
Actions:
  GitHubAction:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        - name: Lint Code Base
          uses: github/super-linter@v4
          env:
            VALIDATE_ALL_CODEBASE: "true"
            DEFAULT_BRANCH: main
```

Nel codice precedente, l'azione CodeCatalyst GitHub Actions (identificata da `aws/github-actions-runner@v1`) racchiude l'azione Super-Linter (identificata da `github/super-linter@v4`) facendola funzionare in un flusso di lavoro. CodeCatalyst

Per ulteriori informazioni, consulta [Aggiungere l'GitHub azione «Azioni»](#).

Tutte le GitHub azioni, curate e non, devono essere racchiuse all'interno di un'GitHub azione Actions (`actions/github-actions-runner@v1`), come mostrato nell'esempio precedente. Il wrapper è necessario per il corretto funzionamento dell'azione.

Argomenti

- [In che modo le GitHub azioni sono diverse dalle CodeCatalyst azioni?](#)
- [GitHub Le azioni possono interagire con altre CodeCatalyst azioni nel flusso di lavoro?](#)
- [Quali GitHub azioni posso usare?](#)
- [Limitazioni delle GitHub azioni in CodeCatalyst](#)
- [Come faccio ad aggiungere un' GitHub azione \(passaggi di alto livello\)?](#)
- [L' GitHub azione viene eseguita GitHub?](#)
- [Posso usare anche i GitHub flussi di lavoro?](#)
- [Immagine di runtime utilizzata dall'GitHub azione «Azioni»](#)
- [Tutorial: codice Lint usando un' GitHubazione](#)
- [Aggiungere l'GitHub azione «Azioni»](#)
- [Aggiungere un'azione curata GitHub](#)
- [Esportazione dei parametri GitHub di output](#)
- [Parametri di GitHub output di riferimento](#)

- [GitHub Azione 'Azioni' YAML](#)

In che modo le GitHub azioni sono diverse dalle CodeCatalyst azioni?

GitHub Le azioni utilizzate all'interno di un CodeCatalyst flusso di lavoro non hanno lo stesso livello di accesso e integrazione AWS e CodeCatalyst le stesse funzionalità (come [ambienti](#) e [problemi](#)) CodeCatalyst delle azioni.

GitHub Le azioni possono interagire con altre CodeCatalyst azioni nel flusso di lavoro?

Sì. Ad esempio, GitHub le azioni possono utilizzare variabili prodotte da altre CodeCatalyst azioni come input e possono anche condividere parametri e artefatti di output con CodeCatalyst le azioni. Per ulteriori informazioni, consultare [Esportazione dei parametri GitHub di output](#) e [Parametri di GitHub output di riferimento](#).

Quali GitHub azioni posso usare?

Puoi utilizzare qualsiasi GitHub azione disponibile tramite la CodeCatalyst console e qualsiasi GitHub azione disponibile nel [GitHub Marketplace](#). Se decidi di utilizzare un' GitHub azione dal Marketplace, tieni presente le seguenti [limitazioni](#).

Limitazioni delle GitHub azioni in CodeCatalyst

- GitHub Le azioni non possono essere utilizzate con il tipo di [calcolo CodeCatalyst Lambda](#).
- GitHub Le azioni vengono eseguite sull'immagine Docker dell'ambiente di runtime di [novembre 2022](#), che include strumenti precedenti. Per ulteriori informazioni sull'immagine e sugli strumenti, consulta. [Specificazione delle immagini dell'ambiente di runtime](#)
- GitHub Le azioni che si basano internamente sul [githubcontesto](#) o che fanno riferimento GitHub a risorse specifiche non funzioneranno. CodeCatalyst Ad esempio, le seguenti azioni non funzioneranno in: CodeCatalyst
 - Azioni che tentano di aggiungere, modificare o aggiornare GitHub risorse. Gli esempi includono azioni che aggiornano le richieste pull o creano problemi in GitHub.
 - Quasi tutte le azioni elencate in <https://github.com/actions>.
- GitHub Le azioni che sono [azioni del contenitore Docker](#) funzioneranno, ma devono essere eseguite dall'utente Docker predefinito (root). Non eseguire l'azione come utente 1001. (Al

momento della scrittura, l'utente 1001 lavora in GitHub, ma non in CodeCatalyst.) Per ulteriori informazioni, consulta l'argomento [USER](#) nel [supporto di Dockerfile](#) per Actions. GitHub

Per un elenco delle GitHub azioni disponibili tramite la CodeCatalyst console, consulta. [Aggiungere un'azione curata GitHub](#)

Come faccio ad aggiungere un' GitHub azione (passaggi di alto livello)?

I passaggi di alto livello per aggiungere un' GitHub azione a un CodeCatalyst flusso di lavoro sono i seguenti:

1. Nel tuo CodeCatalyst progetto, crei un flusso di lavoro. Il flusso di lavoro è il luogo in cui definisci come creare, testare e distribuire l'applicazione. Per ulteriori informazioni, consulta [Guida introduttiva ai flussi di lavoro](#).
2. Nel flusso di lavoro, aggiungi un'azione curata o aggiungi l' GitHub azione GitHub Azioni.
3. Effettuate una delle seguenti operazioni:
 - Se hai scelto di aggiungere un'azione curata, configurala. Per ulteriori informazioni, consulta [Aggiungere un'azione curata GitHub](#).
 - Se scegli di aggiungere un'azione non curata, all'interno dell'azione GitHubAzioni, incolli il codice YAML dell' GitHub azione. Puoi trovare questo codice nella pagina dei dettagli dell' GitHubazione scelta nel [GitHubMarketplace](#). Probabilmente dovrai modificare leggermente il codice per farlo funzionare CodeCatalyst. Per ulteriori informazioni, consulta [Aggiungere l'GitHub azione «Azioni»](#).
4. (Facoltativo) All'interno del flusso di lavoro, aggiungi altre azioni come le azioni di compilazione e test. Per ulteriori informazioni, consulta [Crea, testa e distribuisce con i flussi di lavoro](#).
5. Il flusso di lavoro viene avviato manualmente o automaticamente tramite un trigger. Il flusso di lavoro esegue l' GitHub azione e qualsiasi altra azione del flusso di lavoro. Per ulteriori informazioni, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Per i passaggi dettagliati, consulta:

- [Aggiungere un'azione curata GitHub](#).
- [Aggiungere l'GitHub azione «Azioni»](#).

L' GitHub azione viene eseguita GitHub?

No. L' GitHub azione viene eseguita utilizzando CodeCatalyst l'[immagine CodeCatalyst dell'ambiente di runtime](#).

Posso usare anche i GitHub flussi di lavoro?

No.

Immagine di runtime utilizzata dall'GitHub azione «Azioni»

L'azione CodeCatalyst GitHub Actions viene eseguita su un'[immagine del novembre 2022](#). Per ulteriori informazioni, consulta [Immagini attive](#).

Tutorial: codice Lint usando un' GitHubazione

In questo tutorial, aggiungi l' [GitHub azione Super-Linter](#) a un flusso di lavoro Amazon. CodeCatalyst L'azione Super-Linter ispeziona il codice, trova le aree in cui il codice presenta errori, problemi di formattazione e costrutti sospetti, quindi invia i risultati alla console). CodeCatalyst Dopo aver aggiunto il linter al flusso di lavoro, esegui il flusso di lavoro per lint un'applicazione Node.js di esempio (). app . js Quindi risolvi i problemi segnalati ed eseguite nuovamente il flusso di lavoro per verificare se le correzioni hanno funzionato.

Tip

[Prendi in considerazione l'utilizzo di Super-Linter per linkare file YAML, come i modelli.CloudFormation](#)

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un archivio di sorgenti](#)
- [Passaggio 2: aggiungere un file app.js](#)
- [Passaggio 3: Creare un flusso di lavoro che esegua l'azione Super-Linter](#)
- [Passaggio 4: Risolvi i problemi rilevati da Super-Linter](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare, avrai bisogno di:

- Uno CodeCatalyst spazio con un connesso Account AWS. Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Un progetto vuoto nel tuo CodeCatalyst spazio chiamato `codecatalyst-linter-project`. Scegli l'opzione Inizia da zero per creare questo progetto.

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

Fase 1: Creare un archivio di sorgenti

In questo passaggio, si crea un repository di origine in CodeCatalyst. Utilizzerai questo repository per archiviare il file sorgente dell'applicazione di esempio `app.js`, per questo tutorial.

Per ulteriori informazioni sui repository dei sorgenti, consulta [Creazione di un repository di sorgenti](#)

Per creare un archivio di sorgenti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto, `codecatalyst-linter-project`.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Crea repository.
5. Nel nome del repository, inserisci:

`codecatalyst-linter-source-repository`

6. Scegli Create (Crea) .

Passaggio 2: aggiungere un file `app.js`

In questo passaggio, aggiungi un `app.js` file al tuo repository di origine. `app.js` Contiene un codice di funzione che contiene alcuni errori che il linter troverà.

Per aggiungere il file `app.js`

1. Nella CodeCatalyst console, scegli il tuo progetto, `codecatalyst-linter-project`.

2. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
3. Dall'elenco dei repository di origine, scegli il tuo repository, `codecatalyst-linter-source-repository`
4. In File, scegli Crea file.
5. Nella casella di testo, inserisci il seguente codice:

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;
/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
 */
exports.lambdaHandler = async (event, context) => {
  try {
    // const ret = await axios(url);
    response = {
      statusCode: 200,
      'body': JSON.stringify({
        message: 'hello world'
        // location: ret.data.trim()
      })
    }
  } catch (err) {
    console.log(err)
    return err
  }

  return response
}
```

6. Per Nome file, immettere `app.js`. Mantieni le altre opzioni predefinite.
7. Scegli Applica.

Ora hai creato un file chiamato `app.js`.

Passaggio 3: Creare un flusso di lavoro che esegua l'azione Super-Linter

In questo passaggio, crei un flusso di lavoro che esegua l'azione Super-Linter quando invii il codice al tuo repository di origine. Il flusso di lavoro è costituito dai seguenti elementi costitutivi, definiti in un file YAML:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione GitHub «Azioni»: all'attivazione, l'azione GitHub Azioni esegue l'azione Super-Linter, che a sua volta ispeziona tutti i file nel repository di origine. Se il linter rileva un problema, l'azione del flusso di lavoro fallisce.

Per creare un flusso di lavoro che esegua l'azione Super-Linter

1. Nella CodeCatalyst console, scegli il tuo progetto, `codecatalyst-linter-project`.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli Crea flusso di lavoro.
4. Per Source repository, scegli `codecatalyst-linter-source-repository`.
5. Per Branch, scegli `main`.
6. Scegli Create (Crea).
7. Elimina il codice di esempio YAML.
8. Aggiungi il seguente YAML:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
```

```
SuperLinterAction:
  Identifier: aws/github-actions-runner@v1
  Configuration:
    Steps:
      github-action-code
```

Nel codice precedente, sostituiscilo *github-action-code* con il codice d'azione Super-Linter, come indicato nei seguenti passaggi di questa procedura.

9. Vai alla [pagina Super-Linter](#) nel Marketplace. GitHub
10. In `steps :` (minuscolo), trova il codice e incollalo nel CodeCatalyst flusso di lavoro sotto `Steps :` (maiuscolo).

Modifica il codice GitHub Action per renderlo conforme agli CodeCatalyst standard, come mostrato nel codice seguente.

Il tuo CodeCatalyst flusso di lavoro ora è simile a questo:

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  SuperLinterAction:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        - name: Lint Code Base
          uses: github/super-linter@v4
          env:
            VALIDATE_ALL_CODEBASE: "true"
            DEFAULT_BRANCH: main
```

11. (Facoltativo) Scegli Convalida per assicurarti che il codice YAML sia valido prima di eseguire il commit.
12. Scegli Commit, inserisci un messaggio di Commit, seleziona il tuo **codecatalyst-linter-source-repository** Repository e scegli nuovamente Commit.

Ora hai creato un flusso di lavoro. L'esecuzione di un flusso di lavoro viene avviata automaticamente a causa del trigger definito nella parte superiore del flusso di lavoro.

Per visualizzare il flusso di lavoro in corso

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il flusso di lavoro che hai appena creato: `codecatalyst-linter-workflow`
3. Nel diagramma del flusso di lavoro, scegli SuperLintAction.
4. Attendi che l'azione fallisca. Questo errore è previsto perché il linter ha riscontrato problemi nel codice.
5. Lascia la CodeCatalyst console aperta e vai a [Passaggio 4: Risolvi i problemi rilevati da Super-Linter](#)

Passaggio 4: Risolvi i problemi rilevati da Super-Linter

Super-Linter dovrebbe aver riscontrato problemi nel `app.js` codice e nel `README.md` file incluso nel repository dei sorgenti.

Per risolvere i problemi rilevati da Linter

1. Nella CodeCatalyst console, scegli la scheda Registri, quindi scegli Lint Code Base.

Vengono visualizzati i registri generati dall'azione Super-Linter.

2. Nei log di Super-Linter, scorri verso il basso fino alla riga 90, dove trovi l'inizio dei problemi. Sembrano simili ai seguenti:

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.  
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.  
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body'  
found.  
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but  
found 4.  
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not  
found.
```

3. Correggi `app.js` e `README.md` inserisci le modifiche nel tuo archivio di origine e conferma le modifiche.

Tip

Per risolvere il problema `README.md`, aggiungi markdown al blocco di codice, in questo modo:

```
```markdown
Setup examples:
...
```
```

Le modifiche avviano automaticamente un altro flusso di lavoro. Attendi il completamento del flusso di lavoro. Se hai risolto tutti i problemi, il flusso di lavoro dovrebbe avere esito positivo.

Eliminazione

Pulisci CodeCatalyst per rimuovere le tracce di questo tutorial dal tuo ambiente.

Per pulire CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Elimina `codecatalyst-linter-source-repository`.
3. Elimina `codecatalyst-linter-workflow`.

In questo tutorial, hai imparato come aggiungere l' GitHub azione Super-Linter a un CodeCatalyst flusso di lavoro per inserire del codice.

Aggiungere l'GitHub azione «Azioni»

Un'azione GitHub Actions è un'CodeCatalyst azione che completa un' GitHub azione e la rende compatibile con i flussi di lavoro. CodeCatalyst

Per ulteriori informazioni, consulta [Integrazione con GitHub Actions](#).

Per aggiungere l'azione GitHub Azioni a un flusso di lavoro, segui questi passaggi.

Tip

Per un tutorial che mostra come utilizzare l'azione GitHub Azioni, consulta [Tutorial: codice Lint usando un' GitHubazione](#).

Visual

Per aggiungere l'GitHub azione «Azioni» utilizzando l'editor visuale

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli. GitHub
9. Cercate l'azione GitHub Azioni ed effettuate una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli GitHub Azioni. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, consulta. [GitHub Azione 'Azioni' YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere l'azione «GitHub Azioni» utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/ CodeCatalyst) .
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli. GitHub
9. Cercate l'azione GitHub Azioni ed effettuate una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli GitHub Azioni. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [GitHub Azione 'Azioni' YAML](#)
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Definizione GitHub dell'azione 'Azioni'

L'azione GitHub Actions è definita come un insieme di proprietà YAML all'interno del file di definizione del flusso di lavoro. Per informazioni su queste proprietà, vedere [GitHub Azione 'Azioni' YAML](#) in.

[Definizione YAML del flusso di lavoro](#)

Aggiungere un'azione curata GitHub

Un' GitHub azione curata è un' GitHub azione resa disponibile nella CodeCatalyst console e funge da esempio di come utilizzare un' GitHub azione all'interno di un CodeCatalyst flusso di lavoro.

GitHub Le azioni curate sono racchiuse nell'[azione CodeCatalyst -authored GitHub Actions](#), identificata dall'identificatore. `aws/github-actions-runner@v1` [Ad esempio, ecco come si presenta la versione curata di GitHub Action, OSS: TruffleHog](#)

```
Actions:
  TruffleHogOSS_e8:
    Identifier: aws/github-actions-runner@v1
    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this Workflow as a
source
    Configuration:
      Steps:
        - uses: truffelsecurity/trufflehog@v3.16.0
          with:
            path: ' ' # Required; description: Repository path
            base: ' ' # Required; description: Start scanning from here (usually main
branch).
            head: ' ' # Optional; description: Scan commits until here (usually dev
branch).
            extra_args: ' ' # Optional; description: Extra args to be passed to the
trufflehog cli.
```

Nel codice precedente, l'azione CodeCatalyst GitHub Actions (identificata da `aws/github-actions-runner@v1`) racchiude l'azione TruffleHog OSS (identificata da `truffelsecurity/trufflehog@v3.16.0`), facendola funzionare in un flusso di lavoro. CodeCatalyst

Per configurare questa azione, devi sostituire le stringhe vuote sotto `with:` con i tuoi valori. Esempio:

```
Actions:
```

```
TruffleHogOSS_e8:
  Identifier: aws/github-actions-runner@v1
  Inputs:
    Sources:
      - WorkflowSource # This specifies that the action requires this Workflow as a
source
  Configuration:
    Steps:
      - uses: trufflesecurity/trufflehog@v3.16.0
        with:
          path: ./
          base: main # Required; description: Start scanning from here (usually main
branch).
          head: HEAD # Optional; description: Scan commits until here (usually dev
branch).
          extra_args: '--debug --only-verified' # Optional; description: Extra args
to be passed to the trufflehog cli.
```

Per aggiungere un' GitHub azione curata a un flusso di lavoro, utilizzate la procedura seguente. Per informazioni generali sull'utilizzo GitHub delle azioni in un CodeCatalyst flusso di lavoro, consulta [Integrazione con GitHub Actions](#).

Note

Se non vedi la tua GitHub azione nell'elenco delle azioni curate, puoi comunque aggiungerla al flusso di lavoro utilizzando l'azione GitHub Azioni. Per ulteriori informazioni, consulta [Aggiungere l'GitHub azione «Azioni»](#).

Visual

Per aggiungere un' GitHub azione curata utilizzando l'editor visivo

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.

6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli. GitHub
9. Sfoglia o cerca un' GitHub azione ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli il nome dell' GitHub azione. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Nelle schede Ingressi, Configurazione e Uscite, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, vedi. [GitHub Azione 'Azioni' YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) disponibile per l'azione GitHubActions, così come appare sia nell'editor YAML che in quello visivo.

Per informazioni sulle opzioni di configurazione disponibili per l'Action curata, consulta la relativa documentazione. GitHub

11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere un' GitHub azione curata utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli. GitHub
9. Sfoglia o cerca un' GitHub azione ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli il nome dell' GitHub azione. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
 - 10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile per l'azione GitHub Azioni è fornita in. [GitHub Azione 'Azioni' YAML](#)
- Per informazioni sulle opzioni di configurazione disponibili per l' GitHubAction curata, consulta la relativa documentazione.
11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Esportazione dei parametri GitHub di output

È possibile utilizzare [i parametri GitHub di output](#) nei CodeCatalyst flussi di lavoro.

Note

Un'altra parola per indicare il parametro di output è variabile. Poiché GitHub utilizza il termine parametro di output nella sua documentazione, useremo anche questo termine.

Utilizzate le seguenti istruzioni per esportare un parametro GitHub di output da un' GitHub azione in modo che sia disponibile per l'uso da parte di altre azioni CodeCatalyst del flusso di lavoro.

Per esportare un parametro GitHub di output

1. Apri un flusso di lavoro e scegli Modifica. Per ulteriori informazioni, consulta [Creazione di un flusso di lavoro](#).
2. Nell'azione GitHub Azioni che genera il parametro di output che desideri esportare, aggiungi una Outputs sezione con una Variables proprietà sottostante simile alla seguente:

```
Actions:
  MyGitHubAction:
    Identifier: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'step-id_output-name'
```

Sostituisci:

- **step-id** con il valore della id: proprietà nella steps sezione dell' GitHub azione.
- **output-name** con il nome del parametro GitHub di output.

Esempio

L'esempio seguente mostra come esportare un parametro GitHub di output chiamato SELECTEDCOLOR.

```
Actions:
  MyGitHubAction:
    Identifier: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'random-color-generator_SELECTEDCOLOR'
    Configuration:
      Steps:
        - name: Set selected color
          run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
          id: random-color-generator
```

Parametri di GitHub output di riferimento

Utilizzate le seguenti istruzioni per fare riferimento a un parametro GitHub di output.

Per fare riferimento a un parametro GitHub di output

1. Completa le fasi descritte in [Esportazione dei parametri GitHub di output](#).

Il parametro GitHub di output è ora disponibile per l'uso in altre azioni.

2. Annotate il Variables valore del parametro di output. Include un carattere di sottolineatura (_).
3. Fate riferimento al parametro di output utilizzando la seguente sintassi:

```
${action-name.output-name}
```

Sostituisci:

- *action-name* con il nome dell' CodeCatalystGitHub azione che produce il parametro di output (non utilizzare l'nomeo dell' GitHub azioneid).
- *output-name* con il Variables valore del parametro di output che hai notato in precedenza.

Esempio

```
BuildActionB:
  Identifier: aws/build@v1
  Configuration:
    Steps:
      - Run: echo ${MyGitHubAction.random-color-generator_SELECTEDCOLOR}
```

Esempio con contesto

L'esempio seguente mostra come impostare una SELECTEDCOLOR variabileGitHubActionA, emetterla e quindi farvi riferimentoBuildActionB.

```
Actions:
  GitHubActionA:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
```

```

    - name: Set selected color
      run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
      id: random-color-generator
Outputs:
Variables:
  - 'random-color-generator_SELECTEDCOLOR'

BuildActionB:
Identifier: aws/build@v1
Configuration:
Steps:
  - Run: echo ${GitHubActionA.random-color-generator_SELECTEDCOLOR}

```

GitHub Azione 'Azioni' YAML

Di seguito è riportata la definizione YAML dell'GitHubazione Actions.

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Scegliete una proprietà YAML nel codice seguente per vederne una descrizione.

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visuale. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```

# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
action-name:
  Identifier: aws/github-actions-runner@v1
  DependsOn:

```

- *dependent-action-name-1*

Compute:

Fleet: *fleet-name*

Timeout: *timeout-minutes*

Environment:

Name: *environment-name*

Connections:

- Name: *account-connection-name*

Role: *iam-role-name*

Inputs:

Sources:

- *source-name-1*

- *source-name-2*

Artifacts:

- *artifact-name*

Variables:

- Name: *variable-name-1*

Value: *variable-value-1*

- Name: *variable-name-2*

Value: *variable-value-2*

Outputs:

Artifacts:

- Name: *output-artifact-1*

Files:

- github-output/artifact-1.jar

- "github-output/build*"

- Name: *output-artifact-2*

Files:

- github-output/artifact-2.1.jar

- github-output/artifact-2.2.jar

Variables:

- *variable-name-1*

- *variable-name-2*

AutoDiscoverReports:

Enabled: *true | false*

ReportNamePrefix: *AutoDiscovered*

IncludePaths:

- *"/**/*"*

ExcludePaths:

- *node_modules/cdk/junit.xml*

SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

Vulnerabilities:Severity: *CRITICAL|HIGH|MEDIUM|LOW|INFORMATIONAL*Number: *whole-number*Reports:report-name-1:Format: *format*IncludePaths:- *"*.xml"*ExcludePaths:- *report2.xml*- *report3.xml*SuccessCriteria:PassRate: *percent*LineCoverage: *percent*BranchCoverage: *percent*Vulnerabilities:Severity: *CRITICAL|HIGH|MEDIUM|LOW|INFORMATIONAL*Number: *whole-number*ConfigurationSteps:- *github-actions-code*

nome dell'azione

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Interfaccia utente corrispondente: scheda Configurazione/ *action-name*

Identifier

(*action-name*/Identifier)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Usa `aws/github-actions-runner@v1` per GitHub Actions.

Interfaccia utente corrispondente: diagramma del flusso di lavoro//*action-name*aws/ @v1 label
github-actions-runner

DependsOn

(*action-name*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente per l'esecuzione di questa azione.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Ingressi/Dipende da - opzionale

Compute

(*action-name*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Fleet

(*action-name*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand:,. Linux.x86-64.Large Linux.x86-64.XLarge Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere. [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omissso, l'impostazione predefinita è. `Linux.x86-64.Large`

Interfaccia utente corrispondente: scheda di configurazione/parco di calcolo - opzionale

Timeout

(*action-name*/Timeout)

(Facoltativo)

Specificate il periodo di tempo, in minuti (editor YAML) o in ore e minuti (editor visivo), che l'azione può essere eseguita prima CodeCatalyst che termini l'azione. Il minimo è 5 minuti e il massimo è descritto in. [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Environment

(*action-name*/Environment)

(Facoltativo)

Specificare l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs e Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*action-name*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*action-name*/Environment/Connections)

(Facoltativo)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere. [Consentire l'accesso alle AWS risorse con connessione Account AWS](#) Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: tab/Environment/What la configurazione è attiva *my-environment*? /menu a tre punti/ Cambia ruolo

Name

(*action-name*/Environment/Connections/Name)

(Obbligatorio se incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: tab/Environment/What la configurazione è attiva *my-environment*? /menu a tre punti/ Cambia ruolo

Role

(*action-name*/Environment/Connections/Role)

(Obbligatorio se incluso [Connections](#))

Specificate il nome del ruolo IAM utilizzato da questa azione per accedere e operare in AWS servizi come Amazon S3 e Amazon ECR. Assicurati che questo ruolo venga aggiunto alla tua Account AWS connessione nel tuo spazio. Per aggiungere un ruolo IAM alla connessione di un account, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

Note

Puoi usare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo con questa azione. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Warning

Limita le autorizzazioni a quelle richieste dall'azione AzioneGitHub . L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

Interfaccia utente corrispondente: la configurazione tab/Environment/What è attiva? *my-environment* /menu a tre punti/ Cambia ruolo

Inputs

(*action-name*/Inputs)

(Facoltativo)

La **Inputs** sezione definisce i dati necessari a un'azione durante l'esecuzione di un flusso di lavoro.

 Note

È consentito un massimo di quattro input (una fonte e tre artefatti) per GitHub azione Actions. Le variabili non vengono conteggiate per questo totale.

Se è necessario fare riferimento a file che risiedono in input diversi (ad esempio una sorgente e un artefatto), l'input di origine è l'input principale e l'artefatto è l'input secondario. I riferimenti ai file negli input secondari richiedono un prefisso speciale per distinguerli dal primario. Per informazioni dettagliate, vedi [Esempio: riferimento a file in più artefatti](#).

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*action-name*/Inputs/Sources)

(Facoltativo)

Specificate le etichette che rappresentano i repository di origine che saranno necessari all'azione. Attualmente, l'unica etichetta supportata è `WorkflowSource`, che rappresenta l'archivio di origine in cui è archiviato il file di definizione del flusso di lavoro.

Se si omette una fonte, è necessario specificare almeno un elemento di input in *action-name*/Inputs/Artifacts

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Ingressi/Sorgenti - opzionale

Artifacts - input

(*action-name*/Inputs/Artifacts)

(Facoltativo)

Specificate gli artefatti delle azioni precedenti che desiderate fornire come input per questa azione. Questi artefatti devono essere già definiti come artefatti di output nelle azioni precedenti.

Se non si specifica alcun artefatto di input, è necessario specificare almeno un repository di origine in `action-name/Inputs/Sources`

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere [Condivisione di artefatti e file tra azioni](#)

 Note

Se l'elenco a discesa Artifacts - opzionale non è disponibile (editor visivo) o se si verificano errori durante la convalida di YAML (editor YAML), è possibile che l'azione supporti un solo input. In questo caso, prova a rimuovere l'input sorgente.

Interfaccia utente corrispondente: scheda Ingressi/Artefatti - opzionale

Variables - input

`(action-name/Inputs/Variables)`

(Facoltativo)

Specificate una sequenza di name/value coppie che definisca le variabili di input che desiderate rendere disponibili all'azione. I nomi delle variabili sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle variabili.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Variabili - opzionale

Outputs

`(action-name/Outputs)`

(Facoltativo)

Definisce i dati emessi dall'azione durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Output

Artifacts - output

`(action-name/Outputs/Artifacts)`

(Facoltativo)

Specificate il nome di un artefatto generato dall'azione. I nomi degli Artifact devono essere univoci all'interno di un flusso di lavoro e sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9) e caratteri di sottolineatura (_). Spazi, trattini (-) e altri caratteri speciali non sono consentiti. Non è possibile utilizzare le virgolette per abilitare spazi, trattini e altri caratteri speciali nei nomi degli artefatti di output.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Outputs/Artifacts

Name

(*action-name*/Outputs/Artifacts/Name)

(Richiesto se incluso) [Artifacts - output](#)

Specificate il nome di un artefatto generato dall'azione. I nomi degli Artifact devono essere univoci all'interno di un flusso di lavoro e sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9) e caratteri di sottolineatura (_). Spazi, trattini (-) e altri caratteri speciali non sono consentiti. Non è possibile utilizzare le virgolette per abilitare spazi, trattini e altri caratteri speciali nei nomi degli artefatti di output.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: restituisce il nome dell'artefatto/crea l'tab/Artifacts/Addartefatto

Files

(*action-name*/Outputs/Artifacts/Files)

(Richiesto se incluso) [Artifacts - output](#)

Specificate i file CodeCatalyst inclusi nell'artefatto generato dall'azione. Questi file vengono generati dall'azione del flusso di lavoro durante l'esecuzione e sono disponibili anche nell'archivio di origine. I percorsi dei file possono risiedere in un repository di origine o in un elemento di un'azione precedente e sono relativi al repository di origine o alla radice dell'elemento. È possibile utilizzare modelli a globo per specificare i percorsi. Esempi:

- Per specificare un singolo file che si trova nella radice della posizione di creazione o della posizione del repository di origine, usa. `my-file.jar`

- Per specificare un singolo file in una sottodirectory, usa `directory/my-file.jar` o `directory/subdirectory/my-file.jar`
- Per specificare tutti i file, utilizzare `**/*`. Il pattern a `**` glob indica che deve corrispondere a un numero qualsiasi di sottodirectory.
- Per specificare tutti i file e le directory in una directory denominata, usa `directory` `"directory/**/*"` Il pattern a `**` glob indica che deve corrispondere a un numero qualsiasi di sottodirectory.
- Per specificare tutti i file in una directory denominata `directory`, ma non nessuna delle sue sottodirectory, usa `"directory/*"`

Note

Se il percorso del file include uno o più asterischi (*) o un altro carattere speciale, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere. [Linee guida e convenzioni sulla sintassi](#)

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Note

Potrebbe essere necessario aggiungere un prefisso al percorso del file per indicare in quale elemento o fonte trovarlo. Per ulteriori informazioni, consultare [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

Interfaccia utente corrispondente: restituisce `tab/Artifacts/Add artefatti/file` prodotti dalla build

Variables - output

(*action-name*/Outputs/Variables)

(Facoltativo)

Specificate le variabili che desiderate che l'azione esporti in modo che siano disponibili per l'uso da parte delle azioni successive.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Output/Variabili/ Aggiungi variabile

nome-variabile-1

(nome-variabile-1) **action-name** /Outputs/Variables

(Facoltativo)

Specificate il nome di una variabile che desiderate che l'azione esporti. Questa variabile deve essere già definita nella Steps sezione Inputs o della stessa azione.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: tab/Variables/Add variabile/nome degli output

AutoDiscoverReports

(**action-name**/Outputs/AutoDiscoverReports)

(Facoltativo)

Definisce la configurazione per la funzione di individuazione automatica.

Quando abiliti l'individuazione automatica, tutte le CodeCatalyst ricerche Inputs passano all'azione e tutti i file generati dall'azione stessa, alla ricerca di report di test, copertura del codice e analisi della composizione del software (SCA). Per ogni report trovato, lo CodeCatalyst trasforma in un report.

CodeCatalyst Un CodeCatalyst report è un report completamente integrato nel CodeCatalyst servizio e che può essere visualizzato e manipolato tramite la CodeCatalyst console.

Note

Per impostazione predefinita, la funzione di individuazione automatica ispeziona tutti i file. È possibile limitare i file da ispezionare utilizzando le [IncludePaths](#) proprietà o. [ExcludePaths](#)

Interfaccia utente corrispondente: nessuna

Enabled

(**action-name**/Outputs/AutoDiscoverReports/Enabled)

(Facoltativo)

Abilita o disabilita la funzione di rilevamento automatico.

I valori validi sono `true` e `false`.

Se `Enabled` viene omissso, l'impostazione predefinita è `true`

Interfaccia utente corrispondente: scheda Outputs/Reports/ Rileva automaticamente i report

ReportNamePrefix

(*action-name*/Outputs/AutoDiscoverReports/ReportNamePrefix)

(Richiesto se [AutoDiscoverReports](#) è incluso e abilitato)

Specificate un prefisso che CodeCatalyst precede tutti i report trovati per assegnare un nome ai report associati. CodeCatalyst Ad esempio, se si specifica un prefisso di `AutoDiscovered` e CodeCatalyst rileva automaticamente due rapporti di test `TestSuiteOne.xml` e `TestSuiteTwo.xml`, i CodeCatalyst report associati verranno denominati `e. AutoDiscoveredTestSuiteOne` `AutoDiscoveredTestSuiteTwo`

Interfaccia utente corrispondente: Outputs tab/Reports/Automatically discover reports/ Report prefix

IncludePaths

(*action-name*/Outputs/AutoDiscoverReports/IncludePaths)

Or

(*action-name*/Outputs/Reports/*report-name-1*/IncludePaths)

(Richiesto se [AutoDiscoverReports](#) è incluso e abilitato o se è incluso) [Reports](#)

Specificate i file e i percorsi dei file da CodeCatalyst includere nella ricerca di report non elaborati. Ad esempio, se si specifica `/test/report/*`, CodeCatalyst cerca la directory nell'intera [immagine di build](#) utilizzata dall'azione per cercare la `/test/report/*` directory. Quando trova quella directory, cerca CodeCatalyst i report in quella directory.

Note

Se il percorso del file include uno o più asterischi (*) o altri caratteri speciali, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere.

[Linee guida e convenzioni sulla sintassi](#)

Se questa proprietà viene omessa, l'impostazione predefinita è "**/*", ovvero la ricerca include tutti i file in tutti i percorsi.

Note

Per i report configurati manualmente, IncludePaths deve essere presente uno schema a globo che corrisponda a un singolo file.

Interfaccia utente corrispondente:

- Percorsi di tab/Reports/Automatically discover reports/'Include/exclude uscita/Includi percorsi
- Gli output tab/Reports/Manually configurano i report/ '/'Includi/escludi percorsi'/ Include i percorsi **report-name-1**

ExcludePaths

(**action-name**/Outputs/AutoDiscoverReports/ExcludePaths)

Or

(**action-name**/Outputs/Reports/**report-name-1**/ExcludePaths)

(Facoltativo)

Specificate i file e i percorsi dei file da escludere durante la ricerca di report non elaborati.

CodeCatalyst Ad esempio, se si specifica "/test/my-reports/**/*", non CodeCatalyst cercherà i file nella /test/my-reports/ directory. Per ignorare tutti i file in una directory, utilizzate il pattern **/* glob.

Note

Se il percorso del file include uno o più asterischi (*) o altri caratteri speciali, racchiudi il percorso tra virgolette doppie (). "" Per ulteriori informazioni sui caratteri speciali, vedere.

[Linee guida e convenzioni sulla sintassi](#)

Interfaccia utente corrispondente:

- Percorsi di tab/Reports/Automatically discover reports/'Include/exclude uscita/ Esclude percorsi

- Gli output tab/Reports/Manually configurano i report/ /'Includi/escludi percorsi'/ Escludi i percorsi ***report-name-1***

SuccessCriteria

(***action-name***/Outputs/AutoDiscoverReports/SuccessCriteria)

Or

(***action-name***/Outputs/Reports/***report-name-1***/SuccessCriteria)

(Facoltativo)

Specificate i criteri di successo per i report di test, copertura del codice, analisi della composizione del software (SCA) e analisi statica (SA).

Per ulteriori informazioni, consulta [Configurazione dei criteri di successo per i report](#).

Interfaccia utente corrispondente:

- I risultati tab/Reports/Automatically scoprono i report/criteri di successo
- Gli output tab/Reports/Manually configurano i report/Criteri di successo ***report-name-1***

PassRate

(***action-name***/Outputs/AutoDiscoverReports/SuccessCriteria/PassRate)

Or

(***action-name***/Outputs/Reports/***report-name-1***/SuccessCriteria/PassRate)

(Facoltativo)

Specificate la percentuale di test da superare in un rapporto di prova affinché il CodeCatalyst rapporto associato venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri relativi alla percentuale di superamento vengono applicati solo ai rapporti di prova. Per ulteriori informazioni sui rapporti di prova, vedere [Rapporti dei test](#).

Interfaccia utente corrispondente:

- tab/Reports/Automatically discover reports/SuccessCriteri di uscita/Frequenza di passaggio

- Gli output tab/Reports/Manually configurano i report/ /Criteri di successo/ Percentuale di successo ***report-name-1***

LineCoverage

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/LineCoverage)

Or

(*action-name*/Outputs/Reports/***report-name-1***/SuccessCriteria/LineCoverage)

(Facoltativo)

Specificate la percentuale di righe da coprire in un rapporto sulla copertura del codice affinché il rapporto associato CodeCatalyst venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri di copertura delle linee vengono applicati solo ai report sulla copertura del codice. Per ulteriori informazioni sui rapporti sulla copertura del codice, vedere [Report di copertura del codice](#).

Interfaccia utente corrispondente:

- tab/Reports/Automatically discover reports/SuccessCriteri di uscita/Copertura di linea
- Gli output tab/Reports/Manually configurano i report//Criteri di successo/ Copertura di linea ***report-name-1***

BranchCoverage

(*action-name*/Outputs/AutoDiscoverReports/SuccessCriteria/BranchCoverage)

Or

(*action-name*/Outputs/Reports/***report-name-1***/SuccessCriteria/BranchCoverage)

(Facoltativo)

Specificate la percentuale di filiali in un rapporto sulla copertura del codice che deve essere coperta affinché il rapporto associato CodeCatalyst venga contrassegnato come superato. I valori validi includono i numeri decimali. Ad esempio, 50, 60.5. I criteri di copertura delle filiali vengono applicati solo ai report sulla copertura del codice. Per ulteriori informazioni sui rapporti sulla copertura del codice, vedere [Report di copertura del codice](#).

Interfaccia utente corrispondente:

- tab/Reports/Automatically discover reports/SuccessCriteri di uscita/Copertura delle filiali
- Gli output tab/Reports/Manually configurano i report//Criteri di successo/ Copertura della filiale **report-name-1**

Vulnerabilities

(**action-name**/Outputs/AutoDiscoverReports/SuccessCriteria/Vulnerabilities)

Or

(**action-name**/Outputs/Reports/**report-name-1**/SuccessCriteria/Vulnerabilities)

(Facoltativo)

Specificare il numero e la gravità massimi di vulnerabilità consentiti nel rapporto SCA affinché il rapporto associato venga contrassegnato come superato. CodeCatalyst Per specificare le vulnerabilità, è necessario specificare:

- La gravità minima delle vulnerabilità da includere nel conteggio. I valori validi, dal più grave al meno grave, sono: CRITICAL, HIGH,, MEDIUMLOW, INFORMATIONAL.

Ad esempio, se si sceglie HIGH, verranno HIGH CRITICAL conteggiate tutte le vulnerabilità.

- Il numero massimo di vulnerabilità della gravità specificata che desideri consentire. Il superamento di questo numero fa sì che il CodeCatalyst rapporto venga contrassegnato come fallito. I valori validi sono numeri interi.

I criteri di vulnerabilità vengono applicati solo ai report SCA. Per ulteriori informazioni sui report SCA, vedere. [Rapporti di analisi della composizione del software](#)

Per specificare la severità minima, utilizzare la Severity proprietà. Per specificare il numero massimo di vulnerabilità, utilizzare la Number proprietà.

Per ulteriori informazioni sui report SCA, vedere. [Tipi di report sulla qualità](#)

Interfaccia utente corrispondente:

- tab/Reports/Automatically discover reports/SuccessCriteri di uscita/vulnerabilità

- Gli output configurano i report/ /Criteri di successo/ tab/Reports/Manually Vulnerabilità **report-name-1**

Reports

(**action-name**/Outputs/Reports)

(Facoltativo)

Una sezione che specifica la configurazione per i report di test.

Interfaccia utente corrispondente: scheda Output/Report

nome-rapporto-1

(nome-relazione-1) **action-name** /Outputs/Reports/

(Obbligatorio se incluso) [Reports](#)

Il nome che vuoi dare al CodeCatalyst rapporto che verrà generato dai tuoi report non elaborati.

Interfaccia utente corrispondente: Outputs tab/Reports/Manually configura report/nome del report

Format

(**action-name**/Outputs/Reports/**report-name-1**/Format)

(Obbligatorio se incluso [Reports](#))

Specificate il formato di file che state utilizzando per i report. I valori possibili sono i seguenti.

- Per i rapporti sui test:
 - Per Cucumber JSON, specifica Cucumber (editor visivo) o CUCUMBERJSON (editor YAML).
 - Per JUnit XML, specifica JUnit(editor visivo) o JUNITXML (editor YAML).
 - Per NUnit XML, specificare NUnit(editor visuale) o NUNITXML (editor YAML).
 - Per NUnit 3 XML, specificare NUnit3(editor visuale) o NUNIT3XML (editor YAML).
 - Per Visual Studio TRX, specifica Visual Studio TRX (editor visivo) o VISUALSTUDIOTRX (editor YAML).
 - Per TestNg XML, specificare TestNg (editor visivo) o **TESTNGXML** (editor YAML).

- Per i report sulla copertura del codice:
 - Per Clover XML, specifica Clover (editor visivo) o CLOVERXML (editor YAML).
 - Per Cobertura XML, specifica Cobertura (editor visivo) o (editor YAML). COBERTURAXML
 - Per JaCoCo XML, specifica JaCoCo(editor visivo) o (editor YAML). JACOCOXML
 - Per SimpleCov JSON generato da [simplecov](#), non da [simplecov-json](#), specifica [Simplecov](#) (editor visivo) o (editor YAML). SIMPLECOV
- Per i report di analisi della composizione del software (SCA):
 - Per SARIF, specifica SARIF (editor visivo) o SARIFSCA (editor YAML).

Interfaccia utente corrispondente: restituisce il tipo di rapporto e il formato del tab/Reports/Manually configure reports/Add rapporto **report-name-1**

Configuration

(**action-name**/Configuration)

(Obbligatorio) Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

Steps

(**action-name**/Configuration/Steps)

(Obbligatorio)

Specificate il codice GitHub dell'azione così come appare nella pagina dei dettagli dell'azione in [GitHub Marketplace](#). Aggiungi il codice seguendo queste linee guida:

1. Incolla il codice dalla **steps** : sezione GitHub Azione nella **Steps** : sezione del CodeCatalyst flusso di lavoro. Il codice inizia con un trattino (-) e ha un aspetto simile al seguente.

GitHub codice da incollare:

```
- name: Lint Code Base
  uses: github/super-linter@v4
  env:
    VALIDATE_ALL_CODEBASE: false
    DEFAULT_BRANCH: master
```

```
GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

2. Rivedi il codice appena incollato e modificalo se necessario in modo che sia conforme agli CodeCatalyst standard. Ad esempio, con il blocco di codice precedente, è possibile rimuovere il codice e aggiungerlo in ***red italics*** grassetto.

CodeCatalyst flusso di lavoro yaml:

Steps:

```
- name: Lint Code Base
  uses: github/super-linter@v4
  env:
    VALIDATE_ALL_CODEBASE: false
    DEFAULT_BRANCH: mastermain
    GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

3. Per il codice aggiuntivo incluso nell' GitHub Action ma che non esiste all'interno della steps : sezione, aggiungilo al CodeCatalyst flusso di lavoro utilizzando il codice CodeCatalyst -equivalent. Puoi consultarla [Definizione YAML del flusso di lavoro](#) per avere informazioni dettagliate su come potresti trasferire il tuo GitHub codice. CodeCatalyst Le fasi dettagliate della migrazione non rientrano nell'ambito di questa guida.

Ecco un esempio di come specificare i percorsi dei file in un'azione GitHub Actions:

Steps:

```
- name: Lint Code Base
  uses: github/super-linter@v4
  ...
- run: cd /sources/WorkflowSource/MyFolder/ && cat file.txt
- run: cd /artifacts/MyGitHubAction/MyArtifact/MyFolder/ && cat file2.txt
```

Per ulteriori informazioni sulla specificazione dei percorsi dei file, vedere [Riferimento ai file del repository di origine](#) e [Riferimento ai file in un artefatto](#).

Interfaccia utente corrispondente: scheda di configurazione/ GitHub Azioni YAML

Configurazione delle immagini di calcolo e di runtime

In un CodeCatalyst flusso di lavoro, puoi specificare l'immagine dell'ambiente di calcolo e di runtime da CodeCatalyst utilizzare per eseguire le azioni del flusso di lavoro.

Compute si riferisce al motore di elaborazione (CPU, memoria e sistema operativo) gestito e mantenuto da CodeCatalyst per eseguire le azioni del flusso di lavoro.

Note

Se il calcolo è definito come una proprietà del flusso di lavoro, non può essere definito come una proprietà di alcuna azione in quel flusso di lavoro. Allo stesso modo, se il calcolo è definito come una proprietà di qualsiasi azione, non può essere definito nel flusso di lavoro.

Un'immagine di ambiente di runtime è un contenitore Docker all'interno del quale vengono CodeCatalyst eseguite le azioni del flusso di lavoro. Il contenitore Docker viene eseguito sulla piattaforma di elaborazione prescelta e include un sistema operativo e strumenti aggiuntivi di cui potrebbe aver bisogno un'azione di workflow, come Node.js AWS CLI e tar.

Argomenti

- [Tipi di calcolo](#)
- [Flotte di calcolo](#)
- [Proprietà del parco veicoli su richiesta](#)
- [Proprietà del parco veicoli assegnate](#)
- [Creazione di una flotta predisposta](#)
- [Modifica di una flotta predisposta](#)
- [Eliminazione di una flotta fornita](#)
- [Assegnazione di una flotta o di un calcolo a un'azione](#)
- [Condivisione dell'elaborazione tra le azioni](#)
- [Specificazione delle immagini dell'ambiente di runtime](#)

Tipi di calcolo

CodeCatalyst offre i seguenti tipi di elaborazione:

- Amazon EC2
- AWS Lambda

Amazon EC2 offre una flessibilità ottimizzata durante le operazioni in azione e Lambda offre velocità di avvio ottimizzate. Lambda supporta esecuzioni più rapide del flusso di lavoro grazie a una minore latenza di avvio. Lambda consente di eseguire flussi di lavoro di base in grado di creare, testare e distribuire applicazioni serverless con runtime comuni. Questi runtime includono Node.js, Python, Java, .NET e Go. Tuttavia, ci sono alcuni casi d'uso che Lambda non supporta e, se hanno un impatto su di te, usa il tipo di elaborazione Amazon: EC2

- Lambda non supporta immagini di ambiente di runtime da un registro specifico.
- Lambda non supporta strumenti che richiedono i permessi di root. Per strumenti come `yum` o `rpm`, usa il tipo di EC2 calcolo Amazon o altri strumenti che non richiedono le autorizzazioni di root.
- Lambda non supporta le build o le esecuzioni di Docker. Le seguenti azioni che utilizzano immagini Docker non sono supportate: Deploy AWS CloudFormation stack, Deploy to Amazon ECS, Amazon S3 AWS CDK publish, AWS CDK bootstrap, `deploy`, `invoke` e Actions. AWS Lambda GitHub Inoltre, GitHub le azioni basate su Docker eseguite all'interno dell'azione CodeCatalyst GitHub Actions non sono supportate con Lambda Compute. Puoi usare alternative che non richiedono i permessi di root, come Podman.
- Lambda non supporta la scrittura su file esterni. `/tmp` Quando configuri le azioni del flusso di lavoro, puoi riconfigurare gli strumenti su cui installare o scrivere. `/tmp` Se hai un'azione di compilazione che si installa su `npm`, assicurati di configurarla per l'installazione su `/tmp`
- Lambda non supporta tempi di esecuzione superiori a 15 minuti.

Flotte di calcolo

CodeCatalyst offre le seguenti flotte di elaborazione:

- flotte su richiesta
- Flotte attrezzate

Con le flotte on-demand, quando inizia un'azione del flusso di lavoro, il flusso di lavoro fornisce le risorse di cui ha bisogno. Le macchine vengono distrutte al termine dell'azione. Paghi solo per il numero di minuti in cui esegui le tue azioni. Le flotte on-demand sono completamente gestite e includono funzionalità di scalabilità automatica per gestire i picchi di domanda.

CodeCatalyst offre anche flotte predisposte che contengono macchine alimentate da Amazon EC2 e gestite da CodeCatalyst. Con le flotte predisposte, configuri un set di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare

immediatamente le azioni. Con le flotte predisposte, le macchine sono sempre in funzione e, per tutta la durata del loro approvvigionamento, comportano dei costi.

Per creare, aggiornare o eliminare una flotta, è necessario disporre del ruolo di amministratore dello spazio o del ruolo di amministratore del progetto.

Proprietà del parco veicoli su richiesta

CodeCatalyst fornisce le seguenti flotte on-demand:

Name	Sistema operativo	Architettura	v CPUs	Memoria (GiB)	Spazio su disco	Tipi di elaborazione supportati
Linux.Arm64.Large	Amazon Linux 2	Arm 64	2	4	64 GB	Amazon EC2
					10 GB	Lambda
Linux.Arm64.XLarge	Amazon Linux 2	Arm 64	4	8	128 GB	Amazon EC2
					10 GB	Lambda
Linux.Arm64.2XLarge	Amazon Linux 2	Arm 64	8	16	128 GB	Amazon EC2
Linux.x86-64.Large	Amazon Linux 2	x86-64	2	4	64 GB	Amazon EC2
					10 GB	Lambda
Linux.x86-64.XLarge	Amazon Linux 2	x86-64	4	8	128 GB	Amazon EC2
					10 GB	Lambda

Name	Sistema operativo	Architettura	v CPUs	Memoria (GiB)	Spazio su disco	Tipi di elaborazione supportati
Linux.x86-64.2XLarge	Amazon Linux 2	x86-64	8	16	128 GB	Amazon EC2

Note

Le specifiche per le flotte on-demand variano a seconda del livello di fatturazione. Per ulteriori informazioni, consultare [Prezzi](#).

Se non è selezionata alcuna flotta, utilizza. CodeCatalyst Linux.x86-64.Large

Proprietà del parco veicoli assegnate

Una flotta predisposta contiene le seguenti proprietà:

Sistema operativo

Il sistema operativo Sono disponibili i seguenti sistemi operativi:

- Amazon Linux 2
- Windows Server 2022

Note

Le flotte Windows sono supportate solo nell'azione di compilazione. Al momento, le altre azioni non supportano Windows.

Architecture

L'architettura del processore. Sono disponibili le seguenti architetture:

- x86_64
- Arm64

Tipo di macchina

Il tipo di macchina per ogni istanza. Sono disponibili i seguenti tipi di macchine:

v CPUs	Memoria (GiB)	Spazio su disco	Sistema operativo
2	4	64 GB	Amazon Linux 2
4	8	128 GB	Amazon Linux 2
			Windows Server 2022
8	16	128 GB	Amazon Linux 2
			Windows Server 2022

Capacity

Il numero iniziale di macchine assegnate alla flotta, che definisce il numero di azioni che possono essere eseguite in parallelo.

Modalità di ridimensionamento

Definisce il comportamento quando il numero di azioni supera la capacità della flotta.

Fornisci capacità aggiuntiva su richiesta

Su richiesta vengono configurate macchine aggiuntive che si ridimensionano automaticamente in risposta alle nuove azioni in esecuzione e quindi si riducono alla capacità di base al termine delle azioni. Ciò può comportare costi aggiuntivi, poiché si paga al minuto per ogni macchina in funzione.

Attendi che sia disponibile una capacità aggiuntiva del parco veicoli

Le azioni eseguite vengono messe in coda fino a quando non è disponibile una macchina. Ciò limita i costi aggiuntivi perché non vengono allocate macchine aggiuntive.

Creazione di una flotta predisposta

Utilizza le seguenti istruzioni per creare una flotta predisposta.

 Note

Le flotte fornite verranno disattivate dopo 2 settimane di inattività. Se riutilizzate, verranno riattivate automaticamente, ma questa riattivazione potrebbe causare una latenza.

Per creare una flotta predisposta

1. Nel pannello di navigazione, scegli CI/CD, quindi scegli Compute.
2. Scegli Crea flotta previsionata.
3. Nel campo di testo Provisioned fleet name, inserisci un nome per la tua flotta.
4. Dal menu a discesa Sistema operativo, scegli il sistema operativo.
5. Dal menu a discesa Tipo di macchina, scegli il tipo di macchina per la tua macchina.
6. Nel campo di testo Capacità, inserisci il numero massimo di macchine del parco macchine.
7. Dal menu a discesa della modalità Scaling, scegliete il comportamento di overflow desiderato. Per ulteriori informazioni sui campi, consulta [Proprietà del parco veicoli assegnate](#).
8. Scegli Create (Crea).

Dopo aver creato la flotta assegnata, sei pronto per assegnarla a un'azione. Per ulteriori informazioni, consulta [Assegnazione di una flotta o di un calcolo a un'azione](#).

Modifica di una flotta predisposta

Utilizza le seguenti istruzioni per modificare una flotta predisposta.

 Note

Le flotte fornite verranno disattivate dopo 2 settimane di inattività. Se riutilizzate, verranno riattivate automaticamente, ma questa riattivazione potrebbe causare una latenza.

Per modificare una flotta predisposta

1. Nel pannello di navigazione, scegli CI/CD, quindi scegli Compute.
2. Nell'elenco Provisioned fleet, scegli il parco veicoli che desideri modificare.
3. Scegli Modifica.

4. Nel campo di testo Capacità, inserisci il numero massimo di macchine del parco macchine.
5. Dal menu a discesa della modalità Scaling, scegliete il comportamento di overflow desiderato. Per ulteriori informazioni sui campi, consulta [Proprietà del parco veicoli assegnate](#).
6. Scegli Save (Salva).

Eliminazione di una flotta fornita

Utilizza le seguenti istruzioni per eliminare una flotta predisposta.

Per eliminare una flotta predisposta

Warning

Prima di eliminare una flotta assegnata, rimuovila da tutte le azioni eliminando la `Fleet` proprietà dal codice YAML dell'azione. Qualsiasi azione che continui a fare riferimento a una flotta predisposta dopo la sua eliminazione avrà esito negativo alla successiva esecuzione dell'azione.

1. Nel pannello di navigazione, scegli CI/CD, quindi scegli Compute.
2. Nell'elenco Provisioned fleet, scegli il parco veicoli che desideri eliminare.
3. Scegli Elimina.
4. Inserisci **delete** per confermare l'eliminazione.
5. Scegli Delete (Elimina).

Assegnazione di una flotta o di un calcolo a un'azione

Per impostazione predefinita, le azioni del flusso di lavoro utilizzano la flotta `Linux.x86-64.Large` on-demand con un tipo di EC2 elaborazione Amazon. Per utilizzare invece una flotta predisposta o per utilizzare una flotta on-demand diversa, ad esempio `Linux.x86-64.2XLarge`, utilizza le seguenti istruzioni.

Visual

Prima di iniziare

- Se desideri assegnare una flotta predisposta, devi prima creare la flotta predisposta. Per ulteriori informazioni, consulta [Creazione di una flotta predisposta](#).

Per assegnare una flotta predisposta o un tipo di flotta diverso a un'azione

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione a cui vuoi assegnare la flotta assegnata o il nuovo tipo di flotta.
8. Scegli la scheda Configurazione.
9. In Compute fleet, procedi come segue:

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand: `Linux.x86-64.Large` `Linux.x86-64.XLarge` Per ulteriori informazioni sulle flotte on-demand, vedere [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omissa, l'impostazione predefinita è `Linux.x86-64.Large`

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.

11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Prima di iniziare

- Se desideri assegnare una flotta predisposta, devi prima creare la flotta predisposta. Per ulteriori informazioni, consulta [Creazione di una flotta predisposta](#).

Per assegnare una flotta predisposta o un tipo di flotta diverso a un'azione

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Trova l'azione a cui vuoi assegnare la tua flotta accantonata o il nuovo tipo di flotta.
8. Nell'azione, aggiungi una Compute proprietà e imposta Fleet il nome della tua flotta o il tipo di flotta su richiesta. Per ulteriori informazioni, consulta la descrizione della Fleet proprietà nella sezione [Crea e testa azioni YAML](#) dedicata all'azione.
9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Condivisione dell'elaborazione tra le azioni

Per impostazione predefinita, le azioni in un flusso di lavoro vengono eseguite su istanze separate in un parco veicoli. Questo comportamento fornisce azioni con isolamento e prevedibilità sullo stato degli input. Il comportamento predefinito richiede una configurazione esplicita per condividere il contesto, ad esempio file e variabili, tra le azioni.

La condivisione del calcolo è una funzionalità che consente di eseguire tutte le azioni in un flusso di lavoro sulla stessa istanza. L'utilizzo della condivisione del calcolo può fornire tempi di esecuzione del flusso di lavoro più rapidi perché viene dedicato meno tempo al provisioning delle istanze. È inoltre possibile condividere file (artefatti) tra azioni senza ulteriori configurazioni del flusso di lavoro.

Quando un flusso di lavoro viene eseguito utilizzando la condivisione del calcolo, un'istanza del parco dati predefinito o specificato è riservata per la durata di tutte le azioni in quel flusso di lavoro. Al termine dell'esecuzione del flusso di lavoro, la prenotazione dell'istanza viene rilasciata.

Argomenti

- [Esecuzione di più azioni su elaborazione condivisa](#)
- [Considerazioni sulla condivisione del calcolo](#)
- [Attivazione della condivisione del calcolo](#)
- [Esempi](#)

Esecuzione di più azioni su elaborazione condivisa

È possibile utilizzare l'Computeattributo nella definizione YAML a livello di flusso di lavoro per specificare sia la flotta che le proprietà di condivisione del calcolo delle azioni. Puoi anche configurare le proprietà di calcolo utilizzando l'editor visivo in. CodeCatalyst Per specificare una flotta, imposta il nome di una flotta esistente, imposta il tipo di elaborazione su e attiva la EC2condivisione delle risorse di calcolo.

Note

La condivisione del calcolo è supportata solo se il tipo di calcolo è impostato su e non è supportata per il sistema operativo Windows Server 2022. EC2 Per ulteriori informazioni sulle flotte di elaborazione, i tipi di calcolo e le proprietà, consulta. [Configurazione delle immagini di calcolo e di runtime](#)

Note

Se utilizzi il livello Free e specifichi manualmente il `Linux.x86-64.XLarge` o `Linux.x86-64.2XLarge` fleet nella definizione del flusso di lavoro YAML, l'azione verrà comunque eseguita sulla flotta predefinita (). `Linux.x86-64.Large` Per ulteriori

informazioni sulla disponibilità e sui prezzi di elaborazione, consulta la [tabella delle opzioni relative ai livelli](#).

Quando la condivisione dell'elaborazione è attivata, la cartella contenente l'origine del flusso di lavoro viene copiata automaticamente tra le azioni. Non è necessario configurare gli artefatti di output e farvi riferimento come artefatti di input in una definizione del flusso di lavoro (file YAML). In qualità di autore di flussi di lavoro, devi collegare le variabili di ambiente utilizzando input e output, proprio come faresti senza utilizzare la condivisione del calcolo. Se desideri condividere cartelle tra azioni esterne all'origine del flusso di lavoro, prendi in considerazione la memorizzazione nella cache dei file. Per ulteriori informazioni, consulta [Condivisione di artefatti e file tra azioni](#) e [Memorizzazione nella cache dei file tra le esecuzioni del workflow](#).

L'archivio di origine in cui si trova il file di definizione del flusso di lavoro è identificato dall'etichetta. `WorkflowSource` Durante l'utilizzo della condivisione di calcolo, l'origine del flusso di lavoro viene scaricata nella prima azione che vi fa riferimento e resa automaticamente disponibile per le azioni successive del flusso di lavoro da utilizzare. Qualsiasi modifica apportata alla cartella contenente l'origine del flusso di lavoro mediante un'azione, come l'aggiunta, la modifica o la rimozione di file, è visibile anche nelle azioni successive del flusso di lavoro. Puoi fare riferimento ai file che si trovano nella cartella di origine del flusso di lavoro in qualsiasi azione del flusso di lavoro, proprio come puoi fare senza utilizzare la condivisione del calcolo. Per ulteriori informazioni, consulta [Riferimento ai file del repository di origine](#).

Note

I flussi di lavoro di condivisione del calcolo devono specificare una sequenza rigorosa di azioni, quindi non è possibile impostare azioni parallele. Sebbene gli artefatti di output possano essere configurati in qualsiasi azione della sequenza, gli artefatti di input non sono supportati.

Considerazioni sulla condivisione del calcolo

Puoi eseguire flussi di lavoro con condivisione dell'elaborazione per accelerare l'esecuzione del flusso di lavoro e condividere il contesto tra le azioni in un flusso di lavoro che utilizzano la stessa istanza. Considera quanto segue per determinare se l'utilizzo della condivisione del calcolo è appropriato per il tuo scenario:

	Condivisione del calcolo	Senza condivisione del calcolo
Tipo di calcolo	Amazon EC2	Amazon EC2, AWS Lambda
Fornitura di istanze	Le azioni vengono eseguite sulla stessa istanza	Le azioni vengono eseguite su istanze separate
Sistema operativo	Amazon Linux 2	Amazon Linux 2, Windows Server 2022 (solo azioni di compilazione)
File di riferimento	<code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code>	<code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code>
Struttura flusso di lavoro	Le azioni possono essere eseguite solo in sequenza	Le azioni possono essere eseguite in parallelo
Accesso ai dati attraverso le azioni del flusso di lavoro	Accedi alla fonte del flusso di lavoro memorizzata nella cache () <code>WorkflowSource</code>	Accedi agli output degli artefatti condivisi (richiede una configurazione aggiuntiva)

Attivazione della condivisione del calcolo

Utilizza le seguenti istruzioni per attivare la condivisione del calcolo per un flusso di lavoro.

Visual

Per attivare la condivisione del calcolo utilizzando l'editor visivo

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.

7. Scegli le proprietà del flusso di lavoro.
8. Dal menu a discesa Tipo di calcolo, scegli. EC2
9. (Facoltativo) Dal menu a discesa Compute fleet, opzionale, scegli una flotta che desideri utilizzare per eseguire le azioni del flusso di lavoro. Puoi scegliere una flotta on-demand o creare e scegliere una flotta predisposta. Per ulteriori informazioni, consulta [Creazione di una flotta predisposta](#) e [Assegnazione di una flotta o di un calcolo a un'azione](#).
10. Attiva l'interruttore per attivare la condivisione del calcolo e far sì che le azioni nel flusso di lavoro vengano eseguite sullo stesso parco macchine.
11. (Facoltativo) Scegli la modalità di esecuzione per il flusso di lavoro. Per ulteriori informazioni, consulta [Configurazione del comportamento di accodamento delle esecuzioni](#).
12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per attivare la condivisione del calcolo utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Attiva la condivisione del calcolo impostando il SharedInstance campo su e suTRUE. Type EC2 Imposta Fleet una flotta di elaborazione che desideri utilizzare per eseguire azioni sul flusso di lavoro. Puoi scegliere una flotta su richiesta o creare e scegliere una flotta predisposta. Per ulteriori informazioni, consulta [Creazione di una flotta predisposta](#) e [Assegnazione di una flotta o di un calcolo a un'azione](#).

In un flusso di lavoro YAML, aggiungi un codice simile al seguente:

```
Name: MyWorkflow
SchemaVersion: "1.0"
Compute: # Define compute configuration.
  Type: EC2
  Fleet: MyFleet # Optionally, choose an on-demand or provisioned fleet.
```

```
SharedInstance: true # Turn on compute sharing. Default is False.
Actions:
  BuildFirst:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ...
        ...
```

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Esempi

Argomenti

- [Esempio: Amazon S3 Publish](#)

Esempio: Amazon S3 Publish

I seguenti esempi di flussi di lavoro mostrano come eseguire l'azione Amazon S3 Publish in due modi: prima utilizzando artefatti di input e poi utilizzando la condivisione di calcolo. Con la condivisione del calcolo, gli artefatti di input non sono necessari poiché puoi accedere alla cache. `WorkflowSource` Inoltre, l'artefatto di output nell'azione Crea non è più necessario. L'azione S3 Publish è configurata per utilizzare la `DependsOn` proprietà esplicita per mantenere azioni sequenziali; l'azione Build deve essere eseguita correttamente per poter eseguire l'azione S3 Publish.

- Senza la condivisione del calcolo, è necessario utilizzare artefatti di input e condividere gli output con azioni successive:

```
Name: S3PublishUsingInputArtifact
SchemaVersion: "1.0"
Actions:
  Build:
    Identifier: aws/build@v1
    Outputs:
```



```

    Artifacts:
      - Name: ArtifactToPublish
        Files: [output.zip]
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ./build.sh # Build script that generates output.zip
PublishToS3:
  Identifier: aws/s3-publish@v1
  Inputs:
    Artifacts:
      - ArtifactToPublish
  Environment:
    Connections:
      - Role: codecatalyst-deployment-role
        Name: dev-deployment-role
    Name: dev-connection
  Configuration:
    SourcePath: output.zip
    DestinationBucketName: amzn-s3-demo-bucket

```

- Quando si utilizza la condivisione del calcolo SharedInstance impostando su TRUE, è possibile eseguire più azioni sulla stessa istanza e condividere artefatti specificando un'unica origine del flusso di lavoro. Gli artefatti di input non sono obbligatori e non possono essere specificati:

```

Name: S3PublishUsingComputeSharing
SchemaVersion: "1.0"
Compute:
  Type: EC2
  Fleet: dev-fleet
  SharedInstance: TRUE
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:

```

```
- Run: ./build.sh # Build script that generates output.zip
PublishToS3:
  Identifier: aws/s3-publish@v1
  DependsOn:
    - Build
  Environment:
    Connections:
      - Role: codecatalyst-deployment-role
        Name: dev-deployment-role
      Name: dev-connection
  Configuration:
    SourcePath: output.zip
    DestinationBucketName: amzn-s3-demo-bucket
```

Specificazione delle immagini dell'ambiente di runtime

Un'immagine di ambiente di runtime è un contenitore Docker all'interno del quale vengono CodeCatalyst eseguite le azioni del flusso di lavoro. Il contenitore Docker viene eseguito sulla piattaforma di elaborazione prescelta e include un sistema operativo e strumenti aggiuntivi di cui potrebbe aver bisogno un'azione di workflow, come Node.js AWS CLI e tar.

Per impostazione predefinita, le azioni del flusso di lavoro verranno eseguite su una delle [immagini attive](#) fornite e gestite da CodeCatalyst. Solo le azioni di compilazione e test supportano immagini personalizzate. Per ulteriori informazioni, consulta [Assegnazione di un'immagine Docker in un ambiente di runtime personalizzato a un'azione](#).

Argomenti

- [Immagini attive](#)
- [Cosa succede se un'immagine attiva non include gli strumenti di cui ho bisogno?](#)
- [Assegnazione di un'immagine Docker in un ambiente di runtime personalizzato a un'azione](#)
- [Esempi](#)

Immagini attive

Le immagini attive sono immagini di ambiente di runtime che sono completamente supportate CodeCatalyst e includono strumenti preinstallati. Attualmente esistono due set di immagini attive: uno rilasciato a marzo 2024 e l'altro rilasciato a novembre 2022.

Il fatto che un'azione utilizzi un'immagine di marzo 2024 o novembre 2022 dipende dall'azione:

- [Le azioni di creazione e test che vengono aggiunte a un flusso di lavoro a partire dal 26 marzo 2024 includeranno una Container sezione nella loro definizione YAML che specifica esplicitamente un'immagine di marzo 2024. Facoltativamente, puoi rimuovere la Container sezione per tornare all'immagine di novembre 2022.](#)
- [Le azioni di compilazione e test aggiunte a un flusso di lavoro prima del 26 marzo 2024 non includeranno una Container sezione nella loro definizione YAML e di conseguenza utilizzeranno un'immagine di novembre 2022.](#) Puoi conservare l'immagine di novembre 2022 o aggiornarla. Per aggiornare l'immagine, apri l'azione nell'editor visivo, scegli la scheda Configurazione, quindi seleziona l'immagine di marzo 2024 dall'elenco a discesa delle immagini docker dell'ambiente Runtime. Questa selezione aggiungerà una Container sezione alla definizione YAML dell'azione che viene compilata con l'immagine appropriata di marzo 2024.
- [Tutte le altre azioni utilizzeranno un'immagine di novembre 2022 o un'immagine di marzo 2024.](#) Per ulteriori informazioni, consulta la documentazione dell'azione.

Argomenti

- [Immagini di marzo 2024](#)
- [Immagini di novembre 2022](#)

Immagini di marzo 2024

Le immagini di marzo 2024 sono le ultime immagini fornite da CodeCatalyst. Esiste un'immagine del marzo 2024 per combinazione di elaborazione type/fleet.

La tabella seguente mostra gli strumenti installati su ogni immagine di marzo 2024.

strumenti di immagine di marzo 2024

Strumento	CodeCatalyst Amazon EC2 per Linux x86_64 - CodeCatalystLinux_x86_64:2024_03	CodeCatalyst Lambda per Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2024_03	CodeCatalyst Amazon EC2 per Linux Arm64 - CodeCatalystLinux_Arm64:2024_03	CodeCatalyst Lambda per Arm64 - CodeCatalystLinux_Arm64:2024_03
AWS CLI	2.15.17	2,15,17	2,15,17	2,15,17
AWS CLI Copilot	1.32.1	1,32,1	1,32,1	1,32,1
Docker	24,9	N/D	24,9	N/D
Docker Compose	2.23.3	N/D	2.23.3	N/D
Git	2,43,0	2,43,0	2,43,0	2,43,0
Go	1,21,5	1,21,5	1,21,5	1,21,5
Gradle	8,5	8,5	8,5	8,5
Java	Corretto 17	Corretto 17	Corretto 17	Corretto 17
Maven	3.9.6	3,9,6	3,9,6	3,9,6
Node.js	18,19,0	18,19,0	18,19,0	18,19,0
npm	10,2,3	10,2,3	10,2,3	10,2,3
Python	3,9,18	3,9,18	3,9,18	3,9,18
Python3	3,11,6	311,6	311,6	311,6
pip	22,31	22,31	22,31	22,31
.NET	8,0100	8,0100	8,0100	8,0100

Immagini di novembre 2022

È disponibile un'immagine del novembre 2022 per type/fleet combinazione di elaborazione. È disponibile anche un'immagine Windows di novembre 2022 con l'azione di creazione se hai configurato una flotta [predisposta](#).

La tabella seguente mostra gli strumenti installati in ogni immagine di novembre 2022.

strumenti di immagine di novembre 2022

Strumento	CodeCatalyst Amazon EC2 per Linux x86_64 - CodeCatalystLinux_x86_64:2022_11	CodeCatalyst Lambda per Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2022_11	CodeCatalyst Amazon EC2 per Linux Arm64 - CodeCatalystLinux_Arm64:2022_11	CodeCatalyst Lambda per Linux Arm64 - CodeCatalystLinux_Arm64:2022_11
AWS CLI	2.15.17	2,15,17	2,15,17	2,15,17
AWS CLI Copilot	0.6.0	0,6,0	N/D	N/D
Docker	23,01	N/D	23,1	N/D
Docker Compose	2.16.0	N/D	2.16.0	N/D
Git	2.40.0	2.40.0	2,39,2	2,339,2
Go	1.20.2	1.20.2	1,20.1	1,20.1
Gradle	80,2	80,2	8.0.1	8.0.1
Java	Corretto 17	Corretto 17	Corretto 17	Corretto 17
Maven	3.9.4	3,9,4	3.9.0	3.9.0
Node.js	16,20,2	16,20,2	16,19,1	16,14,2
npm	8,19,4	8,19,4	8,19,3	8,5,0
Python	3,9,15	2,7,18	3.11.2	2,7,18

Strumento	CodeCatalyst Amazon EC2 per Linux x86_64 - CodeCatalystLinux_ x86_64:2022_11	CodeCatalyst Lambda per Linux x86_64 - CodeCatalystLinuxL ambda_x86_64:2022_11	CodeCatalyst Amazon EC2 per Linux Arm64 - CodeCatalystLinux_ Arm64:2022_11	CodeCatalyst Lambda per Arm64 - CodeCatalystLinux_ ambda_Arm64:2022_11
Python3	N/D	3,9,15	N/D	3.11.2
pip	222,2	222,2	23,1	23,1
.NET	6,0,407	6,0,407	6,0,406	6,0,406

Cosa succede se un'immagine attiva non include gli strumenti di cui ho bisogno?

Se nessuna delle [immagini attive](#) fornite da CodeCatalyst include gli strumenti necessari, hai un paio di opzioni:

- Puoi fornire un'immagine Docker per un ambiente di runtime personalizzato che includa gli strumenti necessari. Per ulteriori informazioni, consulta [Assegnazione di un'immagine Docker in un ambiente di runtime personalizzato a un'azione](#).

Note

Se vuoi fornire un'immagine Docker per un ambiente di runtime personalizzato, assicurati che nell'immagine personalizzata sia installato Git.

- Puoi fare in modo che l'azione di compilazione o test del tuo flusso di lavoro installi gli strumenti di cui hai bisogno.

Ad esempio, puoi includere le seguenti istruzioni nella Steps sezione del codice YAML dell'azione di compilazione o test:

```
Configuration:
  Steps:
    - Run: ./setup-script
```

L'**setup-script**istruzione eseguirebbe quindi il seguente script per installare il gestore di pacchetti Node (npm):

```
#!/usr/bin/env bash
echo "Setting up environment"

touch ~/.bashrc
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
nvm install v16.1.0
source ~/.bashrc
```

Per ulteriori informazioni sull'azione di compilazione YAML, vedere. [Crea e testa azioni YAML](#)

Assegnazione di un'immagine Docker in un ambiente di runtime personalizzato a un'azione

Se non desideri utilizzare un'[immagine Active fornita da CodeCatalyst](#), puoi fornire un'immagine Docker per un ambiente di runtime personalizzato. Se vuoi fornire un'immagine personalizzata, assicurati che abbia Git installato al suo interno. L'immagine può risiedere in Docker Hub, Amazon Elastic Container Registry o in qualsiasi archivio pubblico.

Per sapere come creare un'immagine Docker personalizzata, consulta [Containerizzare un'applicazione](#) nella documentazione Docker.

Usa le seguenti istruzioni per assegnare l'immagine Docker del tuo ambiente di runtime personalizzato a un'azione. Dopo aver specificato un'immagine, la CodeCatalyst distribuisce sulla tua piattaforma di elaborazione all'inizio dell'azione.

Note

Le seguenti azioni non supportano immagini Docker in ambiente di runtime personalizzate: Deploy CloudFormation stack, Deploy to ECS e GitHub Actions. Inoltre, le immagini Docker dell'ambiente di runtime personalizzato non supportano il tipo di calcolo Lambda.

Visual

Per assegnare un'immagine Docker all'ambiente di runtime personalizzato utilizzando l'editor visivo

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli Visual.
6. Nel diagramma del flusso di lavoro, scegli l'azione che utilizzerà l'immagine Docker del tuo ambiente di runtime personalizzato.
7. Scegli la scheda Configurazione.
8. Nella parte inferiore, compila i seguenti campi.

Immagine Docker in ambiente di runtime - opzionale

Specificate il registro in cui è archiviata l'immagine. I valori validi includono:

- CODECATALYST(Editor YAML)

L'immagine è memorizzata nel registro. CodeCatalyst

- Docker Hub (editor visivo) o DockerHub (editor YAML)

L'immagine è memorizzata nel registro delle immagini di Docker Hub.

- Altro registro (editor visivo) o Other (editor YAML)

L'immagine viene memorizzata in un registro di immagini personalizzato. È possibile utilizzare qualsiasi registro disponibile pubblicamente.

- Amazon Elastic Container Registry (editor visivo) o ECR (editor YAML)

L'immagine è archiviata in un repository di immagini di Amazon Elastic Container Registry. Per utilizzare un'immagine in un repository Amazon ECR, questa azione richiede l'accesso ad Amazon ECR. Per abilitare questo accesso, devi creare un [ruolo IAM](#) che includa

le seguenti autorizzazioni e una politica di fiducia personalizzata. (Se lo desideri, puoi modificare un ruolo esistente per includere le autorizzazioni e la policy.)

Il ruolo IAM deve includere le seguenti autorizzazioni nella sua politica del ruolo:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

Il ruolo IAM deve includere la seguente politica di fiducia personalizzata:

Per ulteriori informazioni sulla creazione di ruoli IAM, consulta [Creating a role using custom trust policies \(console\)](#) nella IAM User Guide.

Una volta creato il ruolo, è necessario assegnarlo all'azione tramite un ambiente. Per ulteriori informazioni, consulta [Associare un ambiente a un'azione](#).

URL dell'immagine ECR, immagine Docker Hub o URL dell'immagine

Specifica una delle seguenti proprietà:

- Se utilizzi un CODECATALYST registro, imposta l'immagine su una delle seguenti immagini [attive](#):
 - `CodeCatalystLinux_x86_64:2024_03`
 - `CodeCatalystLinux_x86_64:2022_11`
 - `CodeCatalystLinux_Arm64:2024_03`
 - `CodeCatalystLinux_Arm64:2022_11`
 - `CodeCatalystLinuxLambda_x86_64:2024_03`
 - `CodeCatalystLinuxLambda_x86_64:2022_11`
 - `CodeCatalystLinuxLambda_Arm64:2024_03`
 - `CodeCatalystLinuxLambda_Arm64:2022_11`
 - `CodeCatalystWindows_x86_64:2022_11`
- Se utilizzi un registro Docker Hub, imposta l'immagine sul nome dell'immagine Docker Hub e sul tag opzionale.

- Se utilizzi un registro Amazon ECR, imposta l'immagine sull'URI del registro Amazon ECR.

Ad esempio: `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

- Se utilizzi un registro personalizzato, imposta l'immagine sul valore previsto dal registro personalizzato.
9. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
 10. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per assegnare un'immagine Docker all'ambiente di runtime personalizzato utilizzando l'editor YAML

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
3. Scegli Modifica.
4. Scegli YAML.
5. Trova l'azione a cui vuoi assegnare un'immagine Docker in un ambiente di runtime.
6. Nell'azione, aggiungi una `Container` sezione e le proprietà sottostanti `Registry`. `Image`
Per ulteriori informazioni, consultate la `Container` descrizione `Registry` e `Image` le proprietà nella sezione [Azioni](#) dedicata all'azione.
7. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
8. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Esempi

Gli esempi seguenti mostrano come assegnare un'immagine Docker di un ambiente di runtime personalizzato a un'azione nel file di definizione del flusso di lavoro.

Argomenti

- [Esempio: utilizzo di un'immagine Docker in un ambiente di runtime personalizzato per aggiungere il supporto per Node.js 18 con Amazon ECR](#)
- [Esempio: utilizzo di un'immagine Docker in un ambiente di runtime personalizzato per aggiungere il supporto per Node.js 18 con Docker Hub](#)

Esempio: utilizzo di un'immagine Docker in un ambiente di runtime personalizzato per aggiungere il supporto per Node.js 18 con Amazon ECR

L'esempio seguente mostra come utilizzare un'immagine Docker in un ambiente di runtime personalizzato per aggiungere il supporto per Node.js 18 con [Amazon ECR](#).

```
Configuration:
  Container:
    Registry: ECR
    Image: public.ecr.aws/amazonlinux/amazonlinux:2023
```

Esempio: utilizzo di un'immagine Docker in un ambiente di runtime personalizzato per aggiungere il supporto per Node.js 18 con Docker Hub

[L'esempio seguente mostra come utilizzare un'immagine Docker in un ambiente di runtime personalizzato per aggiungere il supporto per Node.js 18 con Docker Hub.](#)

```
Configuration:
  Container:
    Registry: DockerHub
    Image: node:18.18.2
```

Connessione degli archivi di origine ai flussi di lavoro

Una sorgente, chiamata anche sorgente di input, è un archivio di sorgenti a cui si connette [un'azione del flusso](#) di lavoro per ottenere i file necessari per eseguire le sue operazioni. Ad esempio, un'azione del flusso di lavoro potrebbe connettersi a un repository di origine per ottenere i file di origine dell'applicazione per creare un'applicazione.

CodeCatalyst i flussi di lavoro supportano le seguenti fonti:

- CodeCatalyst repository di origine: per ulteriori informazioni, vedere. [Archivia e collabora sul codice con i repository di sorgenti in CodeCatalyst](#)

- GitHub repository, repository Bitbucket e repository di GitLab progetto: per ulteriori informazioni, consulta. [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#)

Argomenti

- [Specificare l'archivio di origine di un file di workflow](#)
- [Specificare l'archivio di origine di un'azione di workflow](#)
- [Riferimento ai file del repository di origine](#)
- [variabili BranchName " e CommitId "](#)

Specificare l'archivio di origine di un file di workflow

Utilizzate le seguenti istruzioni per specificare l'archivio di CodeCatalyst origine in cui archiviare il file di definizione del flusso di lavoro. Se preferisci specificare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto, consulta invece. [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#)

L'archivio di origine in cui si trova il file di definizione del flusso di lavoro è identificato dall'etichetta, WorkflowSource

Note

È necessario specificare l'archivio di origine in cui risiede il file di definizione del flusso di lavoro la prima volta che si esegue il commit del file di definizione del flusso di lavoro. Dopo questo commit, il repository e il file di definizione del workflow vengono collegati tra loro in modo permanente. L'unico modo per modificare il repository dopo il commit iniziale consiste nel ricreare il flusso di lavoro in un altro repository.

Per specificare l'archivio di origine in cui archiviare il file di definizione del flusso di lavoro

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli Crea flusso di lavoro e crea il flusso di lavoro. Per ulteriori informazioni, consulta [Creazione di un flusso di lavoro](#).

Durante il processo di creazione del flusso di lavoro, puoi specificare il CodeCatalyst repository, il ramo e la cartella in cui desideri archiviare il file di definizione del flusso di lavoro.

Specificare l'archivio di origine di un'azione di workflow

Utilizza le seguenti istruzioni per specificare un repository di origine da utilizzare con un'azione del flusso di lavoro. All'avvio, l'azione raggruppa i file presenti nel repository di origine configurato in un artefatto, scarica l'artefatto nell'[immagine Docker dell'ambiente di runtime](#) in cui è in esecuzione l'azione e quindi completa l'elaborazione utilizzando i file scaricati.

Note

Attualmente, all'interno di un'azione del flusso di lavoro, è possibile specificare solo un repository di origine, ovvero l'archivio di origine in cui risiede il file di definizione del flusso di lavoro (nella directory o in una delle relative sottodirectory). `.codecatalyst/workflows/` Questo archivio di origine è rappresentato dall'etichetta. `WorkflowSource`

Visual

Per specificare l'archivio di origine che verrà utilizzato da un'azione (editor visivo)

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione in cui desideri specificare l'origine.
8. Scegliete Ingressi.
9. In Sorgenti, facoltativo, procedi come segue:

Specificate le etichette che rappresentano i repository di origine che saranno necessari all'azione. Attualmente, l'unica etichetta supportata è `WorkflowSource`, che rappresenta l'archivio di origine in cui è archiviato il file di definizione del flusso di lavoro.

Se si omette una fonte, è necessario specificare almeno un elemento di input in. `action-name/Inputs/Artifacts`

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per specificare l'archivio di origine che verrà utilizzato da un'azione (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione, aggiungi un codice simile al seguente:

```
action-name:
  Inputs:
    Sources:
      - WorkflowSource
```

Per ulteriori informazioni, consulta la descrizione della Sources proprietà nella sezione [Definizione YAML del flusso di lavoro](#) dedicata alle tue azioni.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Riferimento ai file del repository di origine

Se disponete di file che si trovano in un repository di origine e dovete fare riferimento a tali file in una delle azioni del flusso di lavoro, completate la procedura seguente.

Note

Consulta anche [Riferimento ai file in un artefatto](#).

Per fare riferimento a un file archiviato in un repository di origine

- Nell'azione in cui desideri fare riferimento a un file, aggiungi un codice simile al seguente:

```
Actions:
  My-action:
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - run: cd my-app && cat file1.jar
```

Nel codice precedente, l'azione cerca il file nella my-app directory principale del repository dei WorkflowSource sorgenti per trovare e visualizzare il file1.jar file.

variabili BranchName " e CommitId "

L' CodeCatalyst origine produce e imposta BranchName e CommitId variabili durante l'esecuzione del flusso di lavoro. Queste sono note come variabili predefinite. Per informazioni su queste variabili, vedere la tabella seguente.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere [Utilizzo di variabili predefinite](#).

Chiave	Valore
CommitId	L'ID di commit che rappresenta lo stato del repository al momento dell'avvio dell'esecuzione del flusso di lavoro. Ad esempio: <code>example3819261db00a3ab59468c8b</code> Consulta anche: Esempio: riferimento alla variabile predefinita "CommitId"
BranchName	Il nome del ramo in base al quale è stato avviato il workflow. Esempi: <code>main</code>, <code>feature/branch</code>, <code>test-LiJuan</code> Consulta anche: Esempio: riferimento alla variabile predefinita BranchName

Connessione degli archivi di pacchetti ai flussi di lavoro

Un pacchetto è un pacchetto che include sia il software che i metadati necessari per installare il software e risolvere eventuali dipendenze. CodeCatalyst supporta il formato del pacchetto npm.

Un pacchetto è composto da:

- Un nome (ad esempio, `webpack` è il nome di un popolare pacchetto npm)
- Uno spazio [dei nomi](#) opzionale (ad esempio, in) `@types @types/node`
- Un insieme di [versioni](#) (ad esempio, `1.0.0, 1.0.1`) `1.0.2`
- Metadati a livello di pacchetto (ad esempio, tag npm dist)

In CodeCatalyst, puoi pubblicare pacchetti e utilizzare pacchetti dagli CodeCatalyst archivi di pacchetti nei tuoi flussi di lavoro. Puoi configurare un'azione di compilazione o test con un repository di CodeCatalyst pacchetti per configurare automaticamente il client npm di un'azione per inviare e estrarre pacchetti dal repository specificato.

Per ulteriori informazioni sui pacchetti, vedere. [Pubblica e condividi pacchetti software in CodeCatalyst](#)

 Note

Attualmente, le azioni di compilazione e test supportano gli archivi di CodeCatalyst pacchetti.

Argomenti

- [Tutorial: Estrai da un archivio di pacchetti](#)
- [Specificazione degli archivi di CodeCatalyst pacchetti nei flussi di lavoro](#)
- [Utilizzo dei token di autorizzazione nelle azioni del flusso di lavoro](#)
- [Esempi: archivi di pacchetti nei flussi di lavoro](#)

Tutorial: Estrai da un archivio di pacchetti

[In questo tutorial, imparerai come creare un flusso di lavoro che esegua un'applicazione le cui dipendenze vengono estratte da un archivio di pacchetti. CodeCatalyst](#) L'applicazione è una semplice app Node.js che stampa un messaggio «Hello World» nei log. CodeCatalyst [L'applicazione ha un'unica dipendenza: il pacchetto lodash npm](#). Il lodash pacchetto viene utilizzato per trasformare una hello-world stringa in. Hello World Utilizzerai la versione 4.17.20 di questo pacchetto.

[Dopo aver configurato l'applicazione e il flusso di lavoro, configurate CodeCatalyst per bloccare l'importazione di lodash versioni aggiuntive nel repository dei CodeCatalyst pacchetti dal registro pubblico esterno \(npmjs.com\).](#) Quindi verificate che le versioni aggiuntive di siano bloccate correttamente. lodash

Alla fine di questo tutorial, dovresti avere una buona comprensione di come un flusso di lavoro interagisce con gli archivi di pacchetti, sia interni che esterni CodeCatalyst, per recuperare i pacchetti. Dovresti anche comprendere le behind-the-scenes interazioni che avvengono tra npm, il tuo repository di pacchetti, il tuo flusso di lavoro e il file dell'applicazione. package . json

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un archivio di sorgenti](#)
- [Fase 2: Creare gli archivi dei pacchetti CodeCatalyst e del gateway](#)

- [Passaggio 3: creare l'applicazione «Hello World»](#)
- [Passaggio 4: Creare un flusso di lavoro che esegua «Hello World»](#)
- [Fase 5: Verifica il flusso di lavoro](#)
- [Passaggio 6: Blocca le importazioni da npmjs.com](#)
- [Passaggio 7: testare la funzionalità di blocco](#)
- [Eliminazione](#)

Prerequisiti

Prima di iniziare:

- Hai bisogno di uno spazio CodeCatalyst . Per ulteriori informazioni, consulta [Creare uno spazio](#).
- Nel tuo CodeCatalyst spazio, hai bisogno di un progetto vuoto chiamato:

```
codecatalyst-package-project
```

Usa l'opzione Inizia da zero per creare questo progetto.

Per ulteriori informazioni, consulta [Creare un progetto vuoto in Amazon CodeCatalyst](#).

Fase 1: Creare un archivio di sorgenti

In questo passaggio, creerai un repository di origine in CodeCatalyst. Questo repository memorizza i file sorgente del tutorial, come i file `index.js` and `package.json`.

Per ulteriori informazioni sui repository dei sorgenti, consulta [Creazione di un repository di sorgenti](#)

Per creare un archivio di sorgenti

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo progetto, `codecatalyst-package-project`.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Crea repository.
5. Nel nome del repository, inserisci:

```
hello-world-app
```

6. Scegli Create (Crea) .

Fase 2: Creare gli archivi dei pacchetti CodeCatalyst e del gateway

In questo passaggio, crei un repository di pacchetti nel tuo CodeCatalyst progetto e lo connetti a un repository gateway, anch'esso presente nel tuo progetto. CodeCatalyst Successivamente importi la dipendenza del tutorial da npmjs.com in lodash entrambi i repository.

Il repository gateway è il «collante» che collega l'archivio dei pacchetti al npmjs.com pubblico. CodeCatalyst

Per ulteriori informazioni sugli archivi dei pacchetti, consulta. [Pubblica e condividi pacchetti software in CodeCatalyst](#)

Note

Questo tutorial utilizza i termini repository di CodeCatalyst pacchetti e repository gateway per fare riferimento ai due repository creati CodeCatalyst nella procedura seguente.

Per creare repository di CodeCatalyst pacchetti e gateway

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Scegli Crea repository di pacchetti.
3. Nel nome del repository, inserisci:

codecatalyst-package-repository

4. Scegli + Seleziona i repository upstream.
5. Scegli i repository Gateway.
6. Nella npm-public-registry-gatewaycasella, scegli Crea.
7. Scegli Seleziona.
8. Scegli Create (Crea) .

CodeCatalyst crea un repository di pacchetti chiamato `codecatalyst-package-repository` che è collegato a un repository gateway. Il repository del gateway è connesso al registro npmjs.com.

Passaggio 3: creare l'applicazione «Hello World»

In questo passaggio, crei un'applicazione Node.js «Hello World» e ne importi la dipendenza (lodash) nel gateway e nei repository di pacchetti. CodeCatalyst

Per creare l'applicazione, è necessario un computer di sviluppo su cui siano installati Node.js e il client associato. npm

Questo tutorial presuppone che utilizzerai un ambiente di CodeCatalyst sviluppo come macchina di sviluppo. Sebbene non sia necessario utilizzare un ambiente di CodeCatalyst sviluppo, è consigliato perché fornisce un ambiente di lavoro pulito, ha Node.js ed è npm preinstallato ed è facile da eliminare una volta terminato il tutorial. Per ulteriori informazioni sugli ambienti di CodeCatalyst sviluppo, vedere [Creazione di un ambiente di sviluppo](#).

Usa le seguenti istruzioni per avviare un ambiente di CodeCatalyst sviluppo e utilizzarlo per creare l'applicazione 'Hello World'.

Per avviare un ambiente di sviluppo CodeCatalyst

1. Nel pannello di navigazione, scegli Codice, quindi scegli Dev Environments.
2. Nella parte superiore scegli Crea ambiente di sviluppo, quindi scegli AWS Cloud9 (nel browser).
3. Assicurati che Repository sia impostato su hello-world-app e Existing branch sia impostato main su. Scegli Create (Crea) .

Il tuo ambiente di sviluppo viene avviato in una nuova scheda del browser e il tuo repository (hello-world-app) viene clonato al suo interno.

4. Lasciate aperte entrambe le schede CodeCatalyst del browser e passate alla procedura successiva.

Per creare l'applicazione Node.js 'Hello World'

1. Vai al tuo ambiente di sviluppo.
2. Al prompt del terminale, passa alla directory principale del repository di hello-world-app origine:

```
cd hello-world-app
```

3. Inizializza un progetto Node.js:

```
npm init -y
```

L'inizializzazione crea un `package.json` file nella directory principale di `hello-world-app`

4. Connetti il client npm nel tuo ambiente di sviluppo al tuo repository di CodeCatalyst pacchetti:
 1. Passa alla console. CodeCatalyst
 2. Nel riquadro di navigazione scegliere Pacchetti.
 3. Scegli `codecatalyst-package-repository`.
 4. Scegli Connect to repository.
 5. Scegli Crea token. Un token di accesso personale (PAT) viene creato per te.
 6. Scegliete Copia per copiare i comandi.
 7. Passa al tuo ambiente di sviluppo.
 8. Assicurati di essere nella `hello-world-app` directory.
 9. Incolla i comandi. Sono simili ai seguenti:

```
npm set registry=https://packages.us-west-2.codecatalyst.aws/npm/ExampleCompany/codecatalyst-package-project/codecatalyst-package-repository/ --location project
npm set //packages.us-west-2.codecatalyst.aws/npm/ExampleCompany/codecatalyst-package-project/hello-world-app/:_authToken=username:token-secret
```

5. `lodash` Versione di importazione 4.17.20:

```
npm install lodash@v4.17.20 --save --save-exact
```

npm cerca la `lodash` versione 4.17.20 nelle seguenti posizioni, nell'ordine seguente:

- Nell'ambiente di sviluppo. Non può trovarlo qui.
- Nel repository dei CodeCatalyst pacchetti. Non può trovarlo qui.
- Nel repository del gateway. Non può trovarlo qui.
- In `npmjs.com`. Lo trova qui.

npm importa `lodash` nel repository gateway, nel repository dei CodeCatalyst pacchetti e nell'ambiente di sviluppo.

 Note

Se non avessi collegato il client npm al tuo repository di CodeCatalyst pacchetti nel passaggio 4, npm avrebbe estratto lodash direttamente da npmjs.com e non avrebbe importato il pacchetto in nessuno dei due repository.

npm aggiorna anche il `package.json` file con la dipendenza e crea una directory contenente tutte le lodash sue dipendenze. `node_modules lodash`

6. Verifica che sia lodash stato importato con successo nel tuo ambiente di sviluppo. Inserisci:

```
npm list
```

Viene visualizzato il seguente messaggio che indica l'avvenuta importazione:

```
`-- lodash@4.17.20
```

7. (Facoltativo) Apri `hello-world-app/package.json` e verifica che le righe inserite **red bold** siano state aggiunte:

```
{
  "name": "hello-world-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "lodash": "4.17.20"
  }
}
```

8. In `hello-world-app`, crea un file chiamato `index.js` con il seguente contenuto:

 Tip

Puoi usare la navigazione laterale nel tuo ambiente di sviluppo per creare questo file.

```
// Importing lodash library
const _ = require('lodash');

// Input string
const inputString = 'hello-world';

// Transforming the string using lodash
const transformedString = _.startCase(inputString.replace('-', ' '));

// Outputting the transformed string to the console
console.log(transformedString);
```

Per verificare che 'lodash' sia stato importato nel gateway e negli archivi dei CodeCatalyst pacchetti

1. Passa alla console. CodeCatalyst
2. Nel riquadro di navigazione scegliere Pacchetti.
3. Scegli npm-public-registry-gateway.
4. Assicurati che lodash sia visualizzato. La colonna Ultima versione indica 4.17.20.
5. Ripetere questa procedura per codecatalyst-package-repository. Potrebbe essere necessario aggiornare la finestra del browser per visualizzare il pacchetto importato.

Per testare 'Hello World' nel tuo ambiente di sviluppo

1. Passa al tuo ambiente di sviluppo.
2. Assicurati di essere ancora nella hello-world-app directory, quindi esegui l'applicazione:

```
node index.js
```

Viene Hello World visualizzato un messaggio. Node.js ha eseguito l'applicazione utilizzando il lodash pacchetto scaricato nell'ambiente di sviluppo in un passaggio precedente.

Ignorare la directory 'node_modules' e confermare 'Hello World'

1. Ignora `node_modules` la cartella. Inserisci:

```
echo "node_modules/" >> .gitignore
```

È consigliabile evitare di salvare questa directory. Inoltre, il salvataggio di questa directory interferirà con i passaggi successivi di questo tutorial.

2. Aggiungi, esegui il commit e invia:

```
git add .
git commit -m "add the Hello World application"
git push
```

I file dell'applicazione e del progetto 'Hello World' vengono aggiunti all'archivio dei sorgenti.

Passaggio 4: Creare un flusso di lavoro che esegua «Hello World»

In questo passaggio, crei un flusso di lavoro che esegue l'applicazione 'Hello World' utilizzando la dipendenza. `lodash` Il flusso di lavoro include una singola azione o attività chiamata.

`RunHelloWorldApp` L'`RunHelloWorldApp` include i seguenti comandi e sezioni importanti:

- **Packages**

Questa sezione indica il nome dell'archivio dei CodeCatalyst pacchetti a cui l'azione deve connettersi durante l'esecuzione. `npm install`

- - **Run: `npm install`**

Questo comando dice a `npm` di installare le dipendenze specificate nel file. `package.json` L'unica dipendenza specificata nel `package.json` file è `lodash`. `npm` cerca `lodash` nelle seguenti posizioni:

- Nell'immagine Docker in esecuzione dell'azione. Non riesce a trovarlo qui.
- Nel repository dei CodeCatalyst pacchetti. Lo trova qui.

Dopo che `npm` lo trova `lodash`, lo importa nell'immagine Docker eseguendo l'azione.

- - **Run: `npm list`**

Questo comando stampa quale versione di `lodash` è stata scaricata nell'immagine Docker che esegue l'azione.

- **- Run: node index.js**

Questo comando esegue l'applicazione 'Hello World' utilizzando la dipendenza specificata nel file `package.json`

Notate che l'`RunHelloWorldApp` azione è un'azione di compilazione, come indicato dall'`aws/build@v1` identificatore nella parte superiore del flusso di lavoro. Per ulteriori informazioni sull'azione di compilazione, consulta [Creazione con flussi di lavoro](#).

Usa le seguenti istruzioni per creare un flusso di lavoro che estragga la `lodash` dipendenza dal tuo repository di CodeCatalyst pacchetti e quindi esegua l'applicazione 'Hello World'.

Creazione di un flusso di lavoro

1. Passa alla console. CodeCatalyst
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli Crea flusso di lavoro.
4. Per Source repository, scegli `hello-world-app`.
5. Per Branch, scegli `main`.

Il file di definizione del flusso di lavoro verrà creato nel repository e nel ramo di origine scelti.

6. Scegli Create (Crea) .
7. Scegli YAML nella parte superiore.
8. Elimina il codice di esempio YAML.
9. Aggiungi il seguente codice YAML:

```
Name: codecatalyst-package-workflow
SchemaVersion: "1.0"

# Required - Define action configurations.
Actions:
  RunHelloWorldApp:
    # Identifies the action. Do not modify this value.
    Identifier: aws/build@v1
```

```
Compute:
  Type: Lambda
Inputs:
  Sources:
    - WorkflowSource # This specifies your source repository.
Configuration:
  Steps:
    - Run: npm install
    - Run: npm list
    - Run: node index.js
  Container: # This specifies the Docker image that runs the action.
    Registry: CODECATALYST
    Image: CodeCatalystLinuxLambda_x86_64:2024_03
Packages:
  NpmConfiguration:
    PackageRegistries:
      - PackagesRepository: codecatalyst-package-repository
```

Nel codice precedente, sostituiscilo *codecatalyst-package-repository* con il nome del repository dei CodeCatalyst pacchetti in cui hai creato. [Fase 2: Creare gli archivi dei pacchetti CodeCatalyst e del gateway](#)

Per informazioni sulle proprietà di questo file, vedere. [Crea e testa azioni YAML](#)

10. (Facoltativo) Scegliete Convalida per assicurarvi che il codice YAML sia valido prima di eseguire il commit.
11. Scegli Applica.
12. Nella finestra di dialogo Conferma flusso di lavoro, inserisci quanto segue:
 - a. Per il nome del file Workflow, mantenete il valore predefinito, `codecatalyst-package-workflow`.
 - b. Per il messaggio di conferma, inserisci:

```
add initial workflow file
```
 - c. Per Repository, scegli hello-world-app.
 - d. Per il nome del ramo, scegli principale.
 - e. Scegli Applica.

Ora hai creato un flusso di lavoro.

Per eseguire il flusso di lavoro

1. Accanto al flusso di lavoro appena creato (`codecatalyst-package-workflow`), scegli Azioni, quindi scegli Esegui.

Viene avviata l'esecuzione di un flusso di lavoro.

2. Nella notifica verde in alto a destra, scegli il link all'esecuzione. Il link è simile a `View Run-1234`.

Viene visualizzato un diagramma del flusso di lavoro che mostra chi ha avviato l'esecuzione e l'`RunHelloWorldAppazione`.

3. Scegli la casella di `RunHelloWorldAppazione` per vedere l'avanzamento dell'azione.
4. Al termine della corsa, vai a [Fase 5: Verifica il flusso di lavoro](#).

Fase 5: Verifica il flusso di lavoro

In questo passaggio, verifichi che il flusso di lavoro abbia eseguito correttamente l'applicazione «Hello World» con la relativa `lodash` dipendenza.

Per verificare che l'applicazione 'Hello World' sia stata eseguita utilizzando la sua dipendenza

1. Nel diagramma del flusso di lavoro, seleziona la casella. `RunHelloWorldApp`

Viene visualizzato un elenco di messaggi di registro.

2. Espandi il messaggio di `node index.js` registro.

Viene visualizzato il seguente messaggio:

```
[Container] 2024/04/24 21:15:41.545650 Running command node index.js
Hello World
```

L'aspetto di `Hello Word` (anziché `dihello-world`) indica che la `lodash` dipendenza è stata utilizzata correttamente.

3. Espandi il `npm list` registro.

Viene visualizzato un messaggio simile al seguente:

```
### lodash@4.17.20
```

Questo messaggio indica che la `lodash` versione 4.17.20 è stata scaricata nell'immagine Docker che esegue l'azione del flusso di lavoro.

Passaggio 6: Blocca le importazioni da npmjs.com

Ora che `lodash` la versione 4.17.20 è presente nel gateway e negli archivi dei CodeCatalyst pacchetti, puoi bloccare le importazioni di altre versioni. Il blocco impedisce l'importazione accidentale di versioni successive (o precedenti) di `lodash`, che potrebbero contenere codice dannoso. Per ulteriori informazioni, consulta [Modifica dei controlli di origine dei pacchetti](#) e [Attacchi di sostituzione delle dipendenze](#).

Utilizza le seguenti istruzioni per bloccare le importazioni `lodash` nel tuo repository del gateway. Quando blocchi i pacchetti sul gateway, questi vengono bloccati anche nelle posizioni a valle.

Per bloccare le importazioni nell'archivio del gateway

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Scegli npm-publish-registry-gateway.
3. Scegli `lodash`.
4. Nella parte superiore, scegli i controlli Origin.
5. In Upstream, scegli Blocca.
6. Seleziona Salva.

Ora hai bloccato le importazioni nel tuo repository gateway (e negli archivi e computer downstream) da npmjs.com.

Passaggio 7: testare la funzionalità di blocco

In questa sezione, verifichi che il blocco che hai [Passaggio 6: Blocca le importazioni da npmjs.com](#) impostato funzioni. Iniziate configurando 'Hello World' per richiedere la versione 4.17.21 di **lodash** anziché quella disponibile nel repository del gateway, che è la 4.17.20. Verificate quindi che l'applicazione non sia in grado di recuperare la versione 4.17.21 da npmjs.com, il che indica che il blocco è avvenuto con successo. Come test finale, sbloccate le importazioni nel vostro repository gateway e verificate che l'applicazione sia in grado di estrarre correttamente la versione 4.17.21 di `lodash`.

Utilizzate il seguente set di procedure per testare la funzionalità di blocco.

Prima di iniziare

1. Passa al tuo ambiente di sviluppo.
2. Estrai il `codecatalyst-package-workflow.yaml` file che hai creato utilizzando la CodeCatalyst console in precedenza:

```
git pull
```

Per configurare 'Hello World' per richiedere la versione 4.17.21 di 'lodash'

1. Aprire `/hello-world-app/package.json`.
2. Cambia la `lodash` versione alla 4.17.21 come mostrato in: ***red bold***

```
{
  "name": "hello-world-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "lodash": "4.17.21"
  }
}
```

Ora c'è una mancata corrispondenza tra la versione nel `package.json` file (4.17.21) e la versione nei repository del gateway e dei CodeCatalyst pacchetti (4.17.20).

3. Aggiungi, esegui il commit e invia:

```
git add .
git commit -m "update package.json to use lodash 4.17.21"
git push
```

Per verificare che 'Hello World' non possa scaricare la versione 4.17.21 di 'lodash'

1. Esegui il flusso di lavoro con la versione non corrispondente:

1. Passa alla CodeCatalyst console.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Quindi **codecatalyst-package-workflow**, scegli Azioni, quindi scegli Esegui.

npm cerca le dipendenze e rileva che la versione 4.17.21 di lodash è richiesta da 'Hello World'. npm cerca la dipendenza nelle seguenti posizioni, nell'ordine seguente: `package.json`

- Nell'immagine Docker in esecuzione dell'azione. Non riesce a trovarlo qui.
- Nel repository dei CodeCatalyst pacchetti. Non può trovarlo qui.
- Nel repository del gateway. Non può trovarlo qui.
- In npmjs.com. Lo trova qui.

Dopo che npm ha trovato la versione 4.17.21 in npmjs.com, tenta di importarla nel repository del gateway, ma poiché hai impostato il gateway per bloccare le importazioni di, l'importazione non avviene. lodash

Poiché l'importazione non avviene, il flusso di lavoro fallisce.

2. Verifica che il flusso di lavoro non sia riuscito:

1. Nella notifica verde in alto a destra, scegli il link alla corsa. Il link è simile a `View Run-2345`.
2. Nel diagramma del flusso di lavoro, selezionate la `RunHelloWorldApp` casella.
3. Espandi il messaggio di `npm install` registro.

Viene visualizzato il seguente messaggio:

```
[Container] 2024/04/25 17:20:34.995591 Running command npm install
npm ERR! code ETARGET
npm ERR! notarget No matching version found for lodash@4.17.21.
npm ERR! notarget In most cases you or one of your dependencies are requesting
npm ERR! notarget a package version that doesn't exist.

npm ERR! A complete log of this run can be found in: /tmp/.npm/
_logs/2024-05-08T22_03_26_493Z-debug-0.log
```

L'errore indica che la versione 4.17.21 non è stata trovata. Questo è previsto perché l'hai bloccato.

Per sbloccare le importazioni da npmjs.com

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Scegli npm-publish-registry-gateway.
3. Scegli lodash.
4. Nella parte superiore, scegli i controlli Origin.
5. In Upstream, scegli Consenti.
6. Seleziona Salva.

Ora hai sbloccato le importazioni di lodash

Il tuo flusso di lavoro può ora importare la versione 4.17.21 di lodash

Per verificare che le importazioni da npmjs.com siano sbloccate

1. Esegui nuovamente il flusso di lavoro. Questa volta il flusso di lavoro dovrebbe avere successo perché l'importazione della versione 4.17.21 dovrebbe ora funzionare. Per eseguire nuovamente il flusso di lavoro:

1. Scegli CI/CD, quindi scegli Flussi di lavoro.
2. Successivamente **codecatalyst-package-workflow**, scegli Azioni e scegli Esegui.
3. Nella notifica verde in alto, a destra, scegli il link alla corsa. Il link è simile a `View Run-3456`.

Viene visualizzato un diagramma del flusso di lavoro che mostra chi ha avviato l'esecuzione e l'RunHelloWorldAppazione.

4. Scegli la casella di RunHelloWorldAppazione per vedere l'avanzamento dell'azione.
5. Espandi il messaggio di `npm list registro` e verifica che venga visualizzato un messaggio simile al seguente:

```
### lodash@4.17.21
```

Questo messaggio indica che è stata lodash scaricata la versione 4.17.21.

2. Verifica che la versione 4.17.21 sia stata importata nei tuoi CodeCatalyst repository e in quelli del gateway:

1. Nel riquadro di navigazione scegliere Pacchetti.
2. Scegli npm-public-registry-gateway.
3. Trova lodash e assicurati che la versione sia. 4.17.21

 Note

Sebbene la versione 4.17.20 non sia elencata in questa pagina, puoi trovarla scegliendo lodash e quindi scegliendo Versioni nella parte superiore.

4. Ripeti questi passaggi per verificare che la versione 4.17.21 sia stata importata in. `codecatalyst-package-repository`

Eliminazione

Pulisci i file e i servizi utilizzati in questo tutorial per evitare che ti vengano addebitati dei costi.

Per ripulire i pacchetti (tutorial)

1. Eliminare `codecatalyst-package-project`:

- a. Nella CodeCatalyst console, accedi al `codecatalyst-package-project` progetto se non ci sei già.
- b. Nel pannello di navigazione, scegli Impostazioni del progetto.
- c. Scegliete Elimina progetto **delete**, immettete e scegliete Elimina progetto.

CodeCatalyst elimina tutte le risorse del progetto, inclusi i repository di origine, gateway e CodeCatalyst pacchetti. Viene eliminato anche l'ambiente di sviluppo.

2. Elimina il token PAT:

- a. Scegli il tuo nome utente sulla destra, quindi scegli Le mie impostazioni.
- b. In Token di accesso personali, scegli il token che hai creato in questo tutorial e scegli Elimina.

In questo tutorial, hai imparato a creare un flusso di lavoro che esegua un'applicazione che estrae le sue dipendenze da un CodeCatalyst archivio di pacchetti. Hai anche imparato a bloccare e sbloccare l'accesso ai pacchetti nel gateway e negli archivi dei pacchetti. CodeCatalyst

Specificazione degli archivi di CodeCatalyst pacchetti nei flussi di lavoro

In CodeCatalyst, puoi aggiungere un archivio di CodeCatalyst pacchetti alle tue azioni di compilazione e test nel tuo flusso di lavoro. Il tuo repository di pacchetti deve essere configurato con un formato di pacchetto, come npm. Puoi anche scegliere di includere una sequenza di ambiti per il repository di pacchetti selezionato.

Utilizzate le seguenti istruzioni per specificare una configurazione del pacchetto da utilizzare con un'azione del flusso di lavoro.

Visual

Per specificare la configurazione del pacchetto che verrà utilizzata da un'azione (editor visivo)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione Build o Test con cui desideri configurare un repository di pacchetti.
8. Scegliete Pacchetti.
9. Dal menu a discesa Aggiungi configurazione, scegli la configurazione del pacchetto che desideri utilizzare con le azioni del flusso di lavoro.
10. Scegli Aggiungi archivio di pacchetti.
11. Nel menu a discesa Package repository, specifica il nome del repository di CodeCatalyst pacchetti che desideri venga utilizzato dall'azione.

Per ulteriori informazioni sugli archivi dei pacchetti, consulta. [Archivi di pacchetti](#)

12. (Facoltativo) In Ambiti: facoltativo, specificate una sequenza di ambiti che desiderate definire nel registro dei pacchetti.

Quando si definiscono gli ambiti, l'archivio dei pacchetti specificato viene configurato come registro per tutti gli ambiti elencati. Se un pacchetto con l'ambito viene richiesto tramite il client npm, utilizzerà quel repository anziché quello predefinito. Ogni nome di ambito deve avere il prefisso «@».

Se Scopes viene omissso, l'archivio dei pacchetti specificato viene configurato come registro predefinito per tutti i pacchetti utilizzati dall'azione.

[Per ulteriori informazioni sugli ambiti, vedere i pacchetti and Namespace dei pacchetti Scoped.](#)

13. Scegliere Aggiungi.
14. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
15. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per specificare la configurazione del pacchetto che verrà utilizzata da un'azione (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione Build o Test, aggiungi un codice simile al seguente:

```
action-name:
  Configuration:
    Packages:
      NpmConfiguration:
        PackageRegistries:
```

```
- PackagesRepository: package-repository
  Scopes:
    - "@scope"
```

Per ulteriori informazioni, consulta la descrizione della Packages proprietà nella sezione [Crea e testa azioni YAML](#) dedicata alle tue azioni.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Utilizzo dei token di autorizzazione nelle azioni del flusso di lavoro

È possibile utilizzare un token fornito dall'azione del flusso di lavoro per configurare manualmente un gestore di pacchetti per l'autenticazione con gli archivi di CodeCatalyst pacchetti. CodeCatalyst rende questo token disponibile come variabile di ambiente a cui fare riferimento nelle azioni.

Variabile di ambiente	Valore
CATALYST_MACHINE_RESOURCE_NAME	L'identità utente del token di autorizzazione.
CATALYST_PACKAGES_AUTHORIZATION_TOKEN	Il valore del token di autorizzazione.

Note

Tieni presente che queste variabili di ambiente verranno compilate solo se hai configurato l'azione per esportare il token di autorizzazione.

Utilizza le seguenti istruzioni per utilizzare un token di autorizzazione con un'azione del flusso di lavoro.

Visual

Per utilizzare un token di autorizzazione esportato con un'azione (editor visivo)

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.

2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione Build o Test con cui desideri configurare un repository di pacchetti.
8. Scegliete Pacchetti.
9. Attiva il token di autorizzazione all'esportazione.

YAML

Per utilizzare un token di autorizzazione esportato con un'azione (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione Build o Test, aggiungi un codice simile al seguente:

```
Actions:
  action-name:
    Packages:
      ExportAuthorizationToken: true
```

Puoi fare riferimento alle variabili `$CATALYST_MACHINE_RESOURCE_NAME` e di `$CATALYST_PACKAGES_AUTHORIZATION_TOKEN` ambiente nella Steps sezione del tuo YAML. Per ulteriori informazioni, vedi [Esempio: configurazione pip manuale per l'autenticazione con CodeCatalyst](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Esempi: archivi di pacchetti nei flussi di lavoro

Gli esempi seguenti mostrano come fare riferimento ai pacchetti nel file di definizione del flusso di lavoro.

Argomenti

- [Esempio: definizione di pacchetti con NpmConfiguration](#)
- [Esempio: sovrascrivere il registro predefinito](#)
- [Esempio: sovrascrivere gli ambiti nel registro dei pacchetti](#)
- [Esempio: configurazione pip manuale per l'autenticazione con CodeCatalyst](#)

Esempio: definizione di pacchetti con **NpmConfiguration**

L'esempio seguente mostra come definire un pacchetto NpmConfiguration nel file di definizione del flusso di lavoro.

```
Actions:
  Build:
    Identifier: aws/build-beta@v1
    Configuration:
      Packages:
        NpmConfiguration:
          PackageRegistries:
            - PackagesRepository: main-repo
            - PackagesRepository: scoped-repo
          Scopes:
            - "@scope1"
```

Questo esempio configura il client npm come tale:

```
default: main-repo
@scope1: scoped-repo
```

In questo esempio, sono definiti due repository. Il registro predefinito viene impostato `main-repo` così com'è definito senza un ambito. `@scope1` L'ambito è configurato `PackageRegistries` `inscoped-repo`.

Esempio: sovrascrivere il registro predefinito

L'esempio seguente mostra come sovrascrivere il registro predefinito.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-repo-1
    - PackagesRepository: my-repo-2
    - PackagesRepository: my-repo-3
```

Questo esempio configura il client npm come tale:

```
default: my-repo-3
```

Se specifichi più repository predefiniti, l'ultimo repository avrà la priorità. In questo esempio, l'ultimo repository elencato è `my-repo-3`, il che significa a cui npm si conatterà. `my-repo-3` Questo sovrascrive i repository e. `my-repo-1` `my-repo-2`

Esempio: sovrascrivere gli ambiti nel registro dei pacchetti

L'esempio seguente mostra come sovrascrivere un ambito nel registro dei pacchetti.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-default-repo
    - PackagesRepository: my-repo-1
  Scopes:
    - "@scope1"
    - "@scope2"
  - PackagesRepository: my-repo-2
  Scopes:
    - "@scope2"
```

Questo esempio configura il client npm come tale:

```
default: my-default-repo
@scope1: my-repo-1
```

```
@scope2: my-repo-2
```

Se includi gli ambiti di override, l'ultimo repository avrà la priorità. In questo esempio, l'ultima volta che l'ambito `@scope2` è configurato in `PackageRegistries` è `for. my-repo-2`. Ciò sostituisce l'ambito `@scope2` configurato per `my-repo-1`.

Esempio: configurazione **pip** manuale per l'autenticazione con CodeCatalyst

L'esempio seguente mostra come fare riferimento alle variabili dell'ambiente di CodeCatalyst autorizzazione in un'azione di compilazione.

```
Actions:
  Build:
    Identifier: aws/build@v1.0.0
    Configuration:
      Steps:
        - Run: pip config set global.index-url https://$CATALYST_MACHINE_RESOURCE_NAME:
$CATALYST_PACKAGES_AUTHORIZATION_TOKEN@codecatalyst.aws/pypi/my-space/my-project/my-
repo/simple/
      Packages:
        ExportAuthorizationToken: true
```

Richiamo di una funzione Lambda utilizzando un flusso di lavoro

Questa sezione descrive come richiamare una funzione utilizzando un flusso di lavoro. AWS Lambda CodeCatalyst A tale scopo, è necessario aggiungere l'azione di AWS Lambda richiamo al flusso di lavoro. L'azione AWS Lambda invoke richiama la funzione Lambda specificata.

[Oltre a richiamare la funzione, l'azione AWS Lambda invoke converte anche ogni chiave di primo livello nel payload di risposta ricevuto dalla funzione Lambda in una variabile di output del flusso di lavoro.](#) È quindi possibile fare riferimento a queste variabili nelle azioni successive del flusso di lavoro. Se non desideri che tutte le chiavi di primo livello vengano convertite in variabili, puoi utilizzare i filtri per specificare quelle esatte. Per ulteriori informazioni, vedere la descrizione [ResponseFilters](#) della proprietà in [AWS Lambda azione 'invoca' YAML](#).

Argomenti

- [Quando utilizzare questa azione](#)
- [Immagine di runtime utilizzata dall'azione 'invoke'AWS Lambda](#)
- [Esempio: richiamare una funzione Lambda](#)

- [Aggiungere l'AWS Lambda azione «invoca»](#)
- [variabili 'AWS Lambda invoke'](#)
- [AWS Lambda azione 'invoca' YAML](#)

Quando utilizzare questa azione

Utilizza questa azione se desideri aggiungere funzionalità al flusso di lavoro incapsulate ed eseguite da una funzione Lambda.

Ad esempio, potresti volere che il tuo flusso di lavoro invii una `Build started` notifica a un canale Slack prima di iniziare una build dell'applicazione. In questo caso, il flusso di lavoro includerebbe un'azione di AWS Lambda `invoca` per richiamare una Lambda per inviare la notifica Slack e un'azione di [compilazione per creare l'applicazione](#).

Come altro esempio, potresti volere che il tuo flusso di lavoro esegua una scansione delle vulnerabilità dell'applicazione prima che venga distribuita. In questo caso, dovresti utilizzare un'azione di compilazione per creare l'applicazione, un'azione di richiamo per AWS Lambda richiamare una Lambda per la scansione delle vulnerabilità e un'azione di distribuzione per distribuire l'applicazione scansionata.

Immagine di runtime utilizzata dall'azione 'invoke'AWS Lambda

L'azione AWS Lambda `invoke` viene eseguita su un'immagine di [novembre 2022](#). Per ulteriori informazioni, consulta [Immagini attive](#).

Esempio: richiamare una funzione Lambda

Il seguente flusso di lavoro di esempio include l'azione AWS Lambda `invoke`, insieme a un'azione di distribuzione. Il flusso di lavoro invia una notifica Slack che indica l'avvio di una distribuzione, quindi distribuisce un'applicazione in AWS modo che utilizzi un modello. CloudFormation Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione di AWS Lambda richiamo (`LambdaNotify`): all'attivazione, questa azione richiama la funzione `Notify-Start Lambda` nell'AWS account e nella regione specificati (e). `my-aws-`

account us-west-2 Alla chiamata, la funzione Lambda invia una notifica Slack che indica l'avvio di una distribuzione.

- Un'azione Deploy CloudFormation stack (Deploy): al termine dell'azione di AWS Lambda richiamo, l'azione Deploy CloudFormation stack esegue il template () per distribuire lo stack di applicazioni. `cfn-template.yml` Per ulteriori informazioni sull'azione Deploy stack, vedere. CloudFormation [Implementazione di uno stack CloudFormation](#)

Note

Il seguente esempio di flusso di lavoro è a scopo illustrativo e non funzionerà senza una configurazione aggiuntiva.

Note

Nel codice YAML che segue, puoi omettere le `Connections:` sezioni se lo desideri. Se ometti queste sezioni, devi assicurarti che il ruolo specificato nel campo `Default IAM role` nel tuo ambiente includa le autorizzazioni e le policy di fiducia richieste dalle azioni dello stack AWS Lambda invoke e Deploy. CloudFormation Per ulteriori informazioni sulla configurazione di un ambiente con un ruolo IAM predefinito, consulta. [Creazione di un ambiente](#) Per ulteriori informazioni sulle autorizzazioni e sulle politiche di fiducia richieste dalle azioni dello CloudFormation stack AWS Lambda Invoke e Deploy, consulta la descrizione della `Role` proprietà nella e. [AWS Lambda azione 'invoca' YAML Azione 'Implementa CloudFormation stack' YAML](#)

Name: codecatalyst-lambda-invoke-workflow

SchemaVersion: 1.0

Triggers:

- Type: PUSH

Branches:

- main

Actions:

LambdaNotify:

Identifier: aws/lambda-invoke@v1

Environment:

Name: my-production-environment

```
Connections:
  - Name: my-aws-account
    Role: codecatalyst-lambda-invoke-role
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  Function: Notify-Start
  AWSRegion: us-west-2

Deploy:
  Identifier: aws/cfn-deploy@v1
  Environment:
    Name: my-production-environment
    Connections:
      - Name: my-aws-account
        Role: codecatalyst-deploy-role
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      name: my-application-stack
      region: us-west-2
      role-arn: arn:aws:iam::111122223333:role/StackRole
      template: ./cfn-template.yml
      capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
```

Aggiungere l'AWS Lambda azione «invoca»

Utilizza le seguenti istruzioni per aggiungere l'azione AWS Lambda invoke al tuo flusso di lavoro.

Prerequisito

Prima di iniziare, assicurati che la AWS Lambda funzione e il ruolo di esecuzione Lambda associato siano pronti e disponibili in. AWS Per ulteriori informazioni, consulta l'argomento relativo al [ruolo di esecuzione Lambda](#) nella Guida per gli AWS Lambda sviluppatori.

Visual

Per aggiungere l'AWS Lambda azione 'invoke' utilizzando l'editor visuale

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/.](https://codecatalyst.aws/)
2. Scegliere il progetto.

3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di AWS Lambda richiamo ed esegui una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli AWS Lambda invoke. Viene visualizzata la finestra di dialogo dei dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
- 10. Nelle schede Ingressi, Configurazione e Uscite, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, consulta la. [AWS Lambda azione 'invoca' YAML](#). Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.
- 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
- 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere l'AWS Lambda azione 'invoke' utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/) CodeCatalyst .
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di AWS Lambda richiamo ed esegui una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli AWS Lambda invoke. Viene visualizzata la finestra di dialogo dei dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
- 10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [AWS Lambda azione 'invoca' YAML](#)
- 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
- 12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

variabili 'AWS Lambda invoke'

Per impostazione predefinita, l'azione AWS Lambda invoke produce una variabile per chiave di primo livello nel payload di risposta Lambda.

Ad esempio, se il payload di risposta è simile al seguente:

```
responsePayload = {  
  "name": "Saanvi",  
  "location": "Seattle",  
  "department": {  
    "company": "Amazon",
```

```
"team": "AWS"
}
```

... allora l'azione genererebbe le seguenti variabili.

Chiave	Valore
nome	Saanvi
location	Seattle
department	{"azienda": «Amazon», «team»: "AWS"}

Note

È possibile modificare le variabili generate utilizzando la proprietà `ResponseFilters` YAML. Per ulteriori informazioni, consultare [ResponseFilters](#) nella [AWS Lambda azione 'invoca' YAML](#).

Le variabili prodotte e impostate dall'azione 'AWS Lambda invoca' in fase di esecuzione sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

AWS Lambda azione 'invoca' YAML

Di seguito è riportata la definizione YAML dell'azione `invoke.AWS Lambda`. Per informazioni su come utilizzare questa azione, consulta. [Richiamo di una funzione Lambda utilizzando un flusso di lavoro](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

 Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
LambdaInvoke\_nn:
  Identifier: aws/lambda-invoke@v1
  DependsOn:
    - dependent-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
    Artifacts:
      - request-payload
    Variables:
      - Name: variable-name-1
        Value: variable-value-1
      - Name: variable-name-2
        Value: variable-value-2
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
  Configuration:
    Function: my-function/function-arn
    AWSRegion: us-west-2
```

```
# Specify RequestPayload or RequestPayloadFile, but not both.
RequestPayload: '{"firstname": "Li", lastname: "Jean", "company": "ExampleCo",
"team": "Development"}'
RequestPayloadFile: my/request-payload.json
ContinueOnError: true/false
LogType: Tail/None
ResponseFilters: '{"name": ".name", "company": ".department.company"}'
Outputs:
  Artifacts:
    - Name: lambda_artifacts
      Files:
        - "lambda-response.json"
```

LambdaInvoke

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: Lambda_Invoke_Action_Workflow_nn.

Interfaccia utente corrispondente: scheda di configurazione/nome dell'azione

Identifier

(*LambdaInvoke*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: aws/lambda-invoke@v1.

Interfaccia utente corrispondente: Workflow diagram/ LambdaInvoke _nn/ aws/lambda-invoke @v1 label

DependsOn

(*LambdaInvoke*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente per eseguire questa azione.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Ingressi/Dipende da - opzionale

Compute

(*LambdaInvoke*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*LambdaInvoke*/Compute/Type)

(Richiesto [Compute](#) se incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: scheda di configurazione/tipo di calcolo

Fleet

(*LambdaInvoke*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand: `Linux.x86-64.Large` `Linux.x86-64.XLarge` Per ulteriori informazioni sulle flotte on-demand, vedere [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omissa, l'impostazione predefinita è `Linux.x86-64.Large`

Interfaccia utente corrispondente: scheda di configurazione/parco di calcolo

Timeout

(*LambdaInvoke*/Timeout)

(Obbligatorio)

Specificate il periodo di tempo, in minuti (editor YAML) o ore e minuti (editor visivo), che l'azione può essere eseguita prima che termini l'azione. Il minimo è 5 minuti e il massimo è descritto in [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Inputs

(*LambdaInvoke*/Inputs)

(Obbligatorio)

La `Inputs` sezione definisce i dati necessari all'azione di AWS Lambda invocazione durante l'esecuzione di un flusso di lavoro.

Note

È consentito un solo input (fonte o artefatto) per AWS Lambda azione di invoca. Le variabili non vengono conteggiate per questo totale.

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*LambdaInvoke*/Inputs/Sources)

(Obbligatorio se [RequestPayloadFile](#) fornito)

Se desideri passare un file JSON del payload della richiesta all'azione AWS Lambda invoke e questo file di payload è archiviato in un repository di origine, specifica l'etichetta di tale repository di origine. Attualmente, l'unica etichetta supportata è. WorkflowSource

Se il file di payload della richiesta non è contenuto in un repository di origine, deve risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sul file di payload, vedere. [RequestPayloadFile](#)

Note

Invece di specificare un file di payload, è possibile aggiungere il codice JSON del payload direttamente all'azione utilizzando la proprietà. RequestPayload Per ulteriori informazioni, consulta [RequestPayload](#).

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*LambdaInvoke*/Inputs/Artifacts)

(Obbligatorio se fornito [RequestPayloadFile](#))

Se desideri passare un file JSON del payload della richiesta all'azione di AWS Lambda invoke e questo file di payload è contenuto in un elemento di [output di un'azione precedente, specifica l'artefatto](#) qui.

Per ulteriori informazioni sul file di payload, consulta. [RequestPayloadFile](#)

Note

Invece di specificare un file di payload, è possibile aggiungere il codice JSON del payload direttamente all'azione utilizzando la proprietà. RequestPayload Per ulteriori informazioni, consulta [RequestPayload](#).

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda di configurazione/artefatti - opzionale

Variables - input

(*LambdaInvoke*/Inputs/Variables)

(Facoltativo)

Specificate una sequenza di name/value coppie che definisca le variabili di input che desiderate rendere disponibili all'azione. I nomi delle variabili sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle variabili.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Variabili - opzionale

Environment

(*LambdaInvoke*/Environment)

(Obbligatorio)

Specificare l' CodeCatalyst ambiente da utilizzare con l'azione. L'azione si connette Account AWS all'Amazon VPC opzionale specificato nell'ambiente scelto. L'azione utilizza il ruolo IAM predefinito

specificato nell'ambiente per connettersi a e utilizza il ruolo IAM specificato nella [connessione Amazon VPC](#) per connettersi ad Amazon VPC. Account AWS

Note

Se il ruolo IAM predefinito non dispone delle autorizzazioni richieste dall'azione, puoi configurare l'azione per utilizzare un ruolo diverso. Per ulteriori informazioni, consulta [Modifica del ruolo IAM di un'azione](#).

Per ulteriori informazioni sugli ambienti, consulta [Implementazione in e Account AWS VPCs](#) e [Creazione di un ambiente](#).

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Name

(*LambdaInvoke*/Environment/Name)

(Richiesto se [Environment](#) incluso)

Specificate il nome di un ambiente esistente che desiderate associare all'azione.

Interfaccia utente corrispondente: scheda di configurazione/ambiente

Connections

(*LambdaInvoke*/Environment/Connections)

(Facoltativo nelle versioni più recenti dell'azione; richiesto nelle versioni precedenti)

Specificare la connessione dell'account da associare all'azione. È possibile specificare al massimo una connessione a un account in `Environment`.

Se non si specifica una connessione all'account:

- L'azione utilizza la Account AWS connessione e il ruolo IAM predefinito specificati nell'ambiente della CodeCatalyst console. Per informazioni sull'aggiunta di una connessione all'account e del ruolo IAM predefinito all'ambiente, consulta [Creazione di un ambiente](#).
- Il ruolo IAM predefinito deve includere le policy e le autorizzazioni richieste dall'azione. Per determinare quali sono tali politiche e autorizzazioni, consulta la descrizione della proprietà Role nella documentazione sulla definizione YAML dell'azione.

Per ulteriori informazioni sulle connessioni degli account, vedere [Consentire l'accesso alle AWS risorse con connessione Account AWS](#). Per informazioni sull'aggiunta di una connessione tramite account a un ambiente, vedere [Creazione di un ambiente](#).

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / connessione all'account Environment/account/role AWS

Name

(*LambdaInvoke*/Environment/Connections/Name)

(Richiesto se è incluso [Connections](#))

Specificare il nome della connessione all'account.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione dell'azione:

- (Versioni più recenti) tab/Environment/What La configurazione è attiva? *my-environment* /three dot menu/ Cambia ruolo
- (Versioni precedenti) Scheda di configurazione/' ' / connessione all'account Environment/account/role AWS

Role

(*LambdaInvoke*/Environment/Connections/Role)

(Richiesto se è incluso [Connections](#))

Specificate il nome del ruolo IAM utilizzato dall'azione AWS Lambda invoke per accedere AWS e richiamare la funzione Lambda. Assicurati di aver [aggiunto il ruolo al tuo CodeCatalyst spazio](#) e che il ruolo includa le seguenti politiche.

Se non specifichi un ruolo IAM, l'azione utilizza il ruolo IAM predefinito elencato nell'[ambiente](#) nella CodeCatalyst console. Se utilizzi il ruolo predefinito nell'ambiente, assicurati che abbia le seguenti politiche.

- La seguente politica di autorizzazione:

Warning

Limita le autorizzazioni a quelle mostrate nella seguente politica. L'utilizzo di un ruolo con autorizzazioni più ampie potrebbe rappresentare un rischio per la sicurezza.

- La seguente politica di fiducia personalizzata:

Note

Puoi usare il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo con questa azione, se lo desideri. Per ulteriori informazioni su questo ruolo, consulta [Creazione del CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo per il tuo account e il tuo spazio](#). Tieni presente che il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo dispone di autorizzazioni di accesso complete, il che può rappresentare un rischio per la sicurezza. Ti consigliamo di utilizzare questo ruolo solo in tutorial e scenari in cui la sicurezza è meno preoccupante.

Interfaccia utente corrispondente: una delle seguenti a seconda della versione di azione:

- (Versioni più recenti) `tab/Environment/What La configurazione è attiva? my-environment /three dot menu/ Cambia ruolo`
- (Versioni precedenti) `Scheda di configurazione/' '/ Ruolo Environment/account/role`

Configuration

(*LambdaInvoke*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

Function

(*LambdaInvoke*/Configuration/Function)

(Obbligatorio)

Specificate la AWS Lambda funzione che verrà richiamata da questa azione. Puoi specificare il nome della funzione o il relativo Amazon Resource Name (ARN). Puoi trovare il nome o l'ARN nella console Lambda.

Note

L' AWS account in cui risiede la funzione Lambda può essere diverso dall'account specificato in. Connections:

Interfaccia utente corrispondente: scheda di configurazione/funzione

AWSRegion

(*LambdaInvoke*/Configuration/AWSRegion)

(Obbligatorio)

Specificate la AWS regione in cui risiede la AWS Lambda funzione. Per un elenco dei codici regionali, consulta [Endpoint regionali](#) in. Riferimenti generali di AWS

Interfaccia utente corrispondente: scheda di configurazione/bucket di destinazione - opzionale

RequestPayload

(*LambdaInvoke*/Configuration/RequestPayload)

(Facoltativo)

Se vuoi passare un payload della richiesta all'azione AWS Lambda invoke, specifica qui il payload della richiesta, in formato JSON.

Esempio di payload della richiesta:

```
'{ "key": "value" }'
```

Se non vuoi passare un payload di richiesta alla tua funzione Lambda, ometti questa proprietà.

 Note

Puoi specificare `RequestPayload` o `RequestPayloadFile`, ma non entrambi.

Per ulteriori informazioni sul payload della richiesta, consulta l'argomento [Invoke nell'API Reference](#).AWS Lambda

Interfaccia utente corrispondente: scheda di configurazione/payload della richiesta - opzionale

RequestPayloadFile

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Facoltativo)

Se desideri passare un payload della richiesta all'azione AWS Lambda invoke, specifica qui il percorso di questo file di payload della richiesta. Il file deve essere in formato JSON.

Il file di payload della richiesta può risiedere in un repository di origine o in un elemento di un'azione precedente. Il percorso del file è relativo al repository di origine o alla radice dell'artefatto.

Se non vuoi passare un payload di richiesta alla tua funzione Lambda, ometti questa proprietà.

 Note

Puoi specificare `RequestPayload` o `RequestPayloadFile`, ma non entrambi.

Per ulteriori informazioni sul file di payload della richiesta, consultate l'argomento [Invoke nell'API Reference](#).AWS Lambda

Interfaccia utente corrispondente: scheda di configurazione/file di payload della richiesta - opzionale

ContinueOnError

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Facoltativo)

Specificate se desiderate contrassegnare l'azione di AWS Lambda richiamo come riuscita anche se la funzione richiamata fallisce. AWS Lambda Valuta la possibilità di impostare questa proprietà `true` per consentire l'avvio delle azioni successive nel flusso di lavoro nonostante l'errore Lambda.

L'impostazione predefinita prevede di fallire l'azione se la funzione Lambda fallisce («off» nell'editor visuale o `false` nell'editor YAML).

Interfaccia utente corrispondente: scheda di configurazione/ Continua in caso di errore

LogType

(*LambdaInvoke*/Configuration/LogType)

(Facoltativo)

Specificate se desiderate includere i log degli errori nella risposta della funzione Lambda dopo che è stata richiamata. Puoi visualizzare questi registri nella scheda Registri dell'azione Lambda invoke nella console. CodeCatalyst I valori possibili sono:

- `Tail`— log di ritorno
- `None`— non restituire i log

L'impostazione predefinita è `Tail`.

Per ulteriori informazioni sul tipo di registro, consulta l'argomento [Invoke](#) nell'AWS Lambda API Reference.

Per ulteriori informazioni sulla visualizzazione dei log, consultare [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

Interfaccia utente corrispondente: scheda di configurazione/tipo di registro

ResponseFilters

(*LambdaInvoke*/Configuration/ResponseFilters)

(Facoltativo)

Specificate quali chiavi del payload di risposta Lambda desiderate convertire in variabili di output. È quindi possibile fare riferimento alle variabili di output nelle azioni successive del flusso di lavoro. Per ulteriori informazioni sulle variabili in CodeCatalyst, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Ad esempio, se il payload di risposta è simile al seguente:

```
responsePayload = {  
  "name": "Saanvi",  
  "location": "Seattle",  
  "department": {  
    "company": "Amazon",  
    "team": "AWS"  
  }  
}
```

... e i tuoi filtri di risposta hanno il seguente aspetto:

```
Configuration:  
...  
ResponseFilters: '{"name": ".name", "company": ".department.company"}'
```

... quindi l'azione genera le seguenti variabili di output:

Chiave	Valore
nome	Saanvi
company	Amazon

È quindi possibile fare riferimento alle company variabili name and nelle azioni successive.

Se non si specifica alcuna chiave in `ResponseFilters`, l'azione converte ogni chiave di primo livello nella risposta Lambda in una variabile di output. Per ulteriori informazioni, consulta [variabili 'AWS Lambda invoke'](#).

Prendi in considerazione l'utilizzo di filtri di risposta per limitare le variabili di output generate solo a quelle che desideri effettivamente utilizzare.

Interfaccia utente corrispondente: scheda di configurazione/filtri di risposta - opzionale

Outputs

(*LambdaInvoke*/Outputs)

(Facoltativo)

Definisce i dati emessi dall'azione durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Output

Artifacts

(*LambdaInvoke*/Outputs/Artifacts)

(Facoltativo)

Specificate gli artefatti generati dall'azione. È possibile fare riferimento a questi artefatti come input in altre azioni.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: Outputs tab/Artifacts/ Build artefact name

Name

(*LambdaInvoke*/Outputs/Artifacts/Name)

(Facoltativo)

Specificate il nome dell'elemento che conterrà il payload di risposta Lambda restituito dalla funzione Lambda. Il valore predefinito è `lambda_artifacts`. Se non specifichi un artefatto, il payload della risposta Lambda può essere visualizzato nei registri dell'azione, disponibili nella scheda Registri dell'azione nella console. CodeCatalyst Per ulteriori informazioni sulla visualizzazione dei log, consultare [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

Interfaccia utente corrispondente: Outputs tab/Artifacts/ Build artifact name

Files

(*LambdaInvoke*/Outputs/Artifacts/Files)

(Facoltativo)

Specificate i file da includere nell'artefatto. È necessario specificare `lambda-response.json` in modo che venga incluso il file del payload della risposta Lambda.

Interfaccia utente corrispondente: Outputs tab/Artifacts/ Files prodotti da build

Modifica della definizione di un'attività Amazon ECS

Questa sezione descrive come aggiornare il `image` campo in un file di [definizione delle attività di Amazon Elastic Container Service \(Amazon ECS\) utilizzando un flusso di lavoro](#). CodeCatalyst A tale scopo, è necessario aggiungere l'azione di definizione delle attività Render Amazon ECS al flusso di lavoro. Questa azione aggiorna il campo dell'immagine nel file di definizione dell'attività con un nome di immagine Docker fornito dal flusso di lavoro in fase di esecuzione.

Note

È inoltre possibile utilizzare questa azione per aggiornare il `environment` campo della definizione dell'attività con variabili di ambiente.

Argomenti

- [Quando usare questa azione](#)
- [Come funziona l'azione «Render Amazon ECS task definition»](#)
- [Immagine di runtime utilizzata dall'azione «Render Amazon ECS task definition»](#)
- [Esempio: modificare un taskdef di Amazon ECS](#)
- [Aggiungere l'azione «Render Amazon ECS task definition»](#)
- [Visualizzazione del file di definizione delle attività aggiornato](#)
- [Variabili «Render Amazon ECS task definition»](#)
- [Azione «Render Amazon ECS task definition» YAML](#)

Quando usare questa azione

Usala se disponi di un flusso di lavoro che crea e contrassegna un'immagine Docker con contenuti dinamici, come un ID di commit o un timestamp.

Non utilizzare questa azione se il file di definizione dell'attività contiene un valore di immagine che rimane sempre lo stesso. In questo caso, puoi inserire manualmente il nome dell'immagine nel file di definizione dell'attività.

Come funziona l'azione «Render Amazon ECS task definition»

È necessario utilizzare l'azione di definizione delle attività Render Amazon ECS con le azioni Build e Deploy to Amazon ECS nel flusso di lavoro. Insieme, queste azioni funzionano come segue:

1. L'azione di compilazione crea l'immagine Docker e la contrassegna con un nome, un ID di commit, un timestamp o altri contenuti dinamici. Ad esempio, l'azione di compilazione potrebbe essere simile alla seguente:

```
MyECSWorkflow
  Actions:
    BuildAction:
      Identifier: aws/build@v1
      ...
    Configuration:
      Steps:
        # Build, tag, and push the Docker image...
        - Run: docker build -t MyDockerImage:${WorkflowSource.CommitId} .
        ...
```

Nel codice precedente, la `docker build -t` direttiva indica di creare l'immagine Docker e di etichettarla con l'ID di commit in fase di esecuzione dell'azione. Il nome dell'immagine generata potrebbe essere simile al seguente:

`MyDockerImage:a37bd7e`

2. L'azione Render Amazon ECS di definizione dell'attività aggiunge il nome dell'immagine generato dinamicamente al file di definizione dell'attività, in questo modo: `MyDockerImage:a37bd7e`

```
{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": MyDockerImage:a37bd7e,
      "essential": true,
      ...
      "portMappings": [
        {
          "hostPort": 80,
          "protocol": "tcp",
```

```

        "containerPort": 80
      }
    ]
  },
  ...
}

```

Facoltativamente, puoi anche fare in modo che l'azione Render Amazon ECS di definizione dell'attività aggiunga variabili di ambiente alla definizione dell'attività, in questo modo:

```

{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-execution-
role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": MyDockerImage:a37bd7e,
      ...
      "environment": [
        {
          "name": "ECS_LOGLEVEL",
          "value": "info"
        }
      ]
    }
  ],
  ...
}

```

Per ulteriori informazioni sulle variabili di ambiente, consulta [Specificare le variabili di ambiente](#) nella Amazon Elastic Container Service Developer Guide.

3. L'azione Deploy to Amazon ECS registra il file di definizione delle attività aggiornato con Amazon ECS. La registrazione del file di definizione delle attività aggiornato distribuisce la nuova immagine in MyDockerImage:a37bd7e Amazon ECS.

Immagine di runtime utilizzata dall'azione «Render Amazon ECS task definition»

L'azione di definizione delle attività Render Amazon ECS viene eseguita su un'[immagine di novembre 2022](#). Per ulteriori informazioni, consulta [Immagini attive](#).

Esempio: modificare un taskdef di Amazon ECS

Di seguito è riportato un esempio di flusso di lavoro completo che include l'azione di definizione delle attività di Render Amazon ECS, insieme alle azioni di compilazione e distribuzione. Lo scopo del flusso di lavoro è creare e distribuire un'immagine Docker nel tuo cluster Amazon ECS. Il flusso di lavoro è costituito dai seguenti elementi costitutivi eseguiti in sequenza:

- Un trigger: questo trigger avvia l'esecuzione automatica del flusso di lavoro quando invii una modifica al tuo repository di origine. Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).
- Un'azione di compilazione (BuildDocker): all'attivazione, l'azione crea l'immagine Docker utilizzando il Dockerfile, la contrassegna con un ID di commit e invia l'immagine ad Amazon ECR. Per ulteriori informazioni sull'azione di compilazione, consulta [Creazione con flussi di lavoro](#)
- Un'azione di definizione dell'attività di Render Amazon ECS (RenderTaskDef): al termine dell'azione di compilazione, questa azione aggiorna un'azione esistente taskdef.json situata nella radice del repository di origine con un valore di image campo che include l'ID di commit corretto. Salva il file aggiornato con un nuovo nome di file (task-definition-random-string.json) e quindi crea un elemento di output che contiene questo file. L'azione di rendering genera anche una variabile chiamata task-definition e la imposta sul nome del nuovo file di definizione dell'attività. L'artefatto e la variabile verranno utilizzati nell'azione di distribuzione, che è la successiva.
- Un'azione Deploy to Amazon ECS (DeployToECS): al termine dell'azione di definizione dell'attività Render Amazon ECS, l'azione Deploy to Amazon ECS cerca l'artefatto di output generato dall'azione di rendering TaskDefArtifact (), trova il file task-definition-random-string.json al suo interno e lo registra con il servizio Amazon ECS. Il servizio Amazon ECS segue quindi le istruzioni contenute nel task-definition-random-string.json file per eseguire le attività Amazon ECS e i contenitori di immagini Docker associati all'interno del cluster Amazon ECS.

Name: codecatalyst-ecs-workflow

```

SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildDocker:
    Identifier: aws/build@v1
    Environment:
      Name: codecatalyst-ecs-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-ecs-build-role
    Inputs:
      Variables:
        - Name: REPOSITORY_URI
          Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
        - Name: IMAGE_TAG
          Value: ${WorkflowSource.CommitId}
    Configuration:
      Steps:
        #pre_build:
          - Run: echo Logging in to Amazon ECR...
          - Run: aws --version
          - Run: aws ecr get-login-password --region us-east-2 | docker login --username
AWS --password-stdin 111122223333.dkr.ecr.us-east-2.amazonaws.com
        #build:
          - Run: echo Build started on `date`
          - Run: echo Building the Docker image...
          - Run: docker build -t $REPOSITORY_URI:latest .
          - Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
        #post_build:
          - Run: echo Build completed on `date`
          - Run: echo Pushing the Docker images...
          - Run: docker push $REPOSITORY_URI:latest
          - Run: docker push $REPOSITORY_URI:$IMAGE_TAG

RenderTaskDef:
  DependsOn:
    - BuildDocker
  Identifier: aws/ecs-render-task-definition@v1
  Inputs:

```



```

Variables:
  - Name: REPOSITORY_URI
    Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
  - Name: IMAGE_TAG
    Value: ${WorkflowSource.CommitId}
Configuration:
  task-definition: taskdef.json
  container-definition-name: codecatalyst-ecs-container
  image: $REPOSITORY_URI:$IMAGE_TAG
# The output artifact contains the updated task definition file.
# The new file is prefixed with 'task-definition'.
# The output variable is set to the name of the updated task definition file.
Outputs:
  Artifacts:
    - Name: TaskDefArtifact
      Files:
        - "task-definition*"
  Variables:
    - task-definition

DeployToECS:
  Identifier: aws/ecs-deploy@v1
  Environment:
    Name: codecatalyst-ecs-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-ecs-deploy-role
#Input artifact contains the updated task definition file.
Inputs:
  Sources: []
  Artifacts:
    - TaskDefArtifact
Configuration:
  region: us-east-2
  cluster: codecatalyst-ecs-cluster
  service: codecatalyst-ecs-service
  task-definition: ${RenderTaskDef.task-definition}

```

Aggiungere l'azione «Render Amazon ECS task definition»

Utilizza le seguenti istruzioni per aggiungere l'azione di definizione delle attività di Render Amazon ECS al tuo flusso di lavoro.

Prerequisito

Prima di iniziare, assicurati di disporre di un flusso di lavoro che includa un'azione di compilazione che generi dinamicamente un'immagine Docker. Per i dettagli, consulta il [flusso di lavoro di esempio](#) precedente.

Visual

Per aggiungere l'azione «Render Amazon ECS task definition» utilizzando l'editor visuale

1. [Apri la console all'indirizzo https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
9. Cerca l'azione di definizione dell'attività Render Amazon ECS ed esegui una delle seguenti operazioni:

- Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.

Or

- Scegli la definizione delle attività di Render Amazon ECS. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Nelle schede Ingressi e Configurazione, completa i campi in base alle tue esigenze. Per una descrizione di ogni campo, vedi. [Azione «Render Amazon ECS task definition» YAML](#) Questo riferimento fornisce informazioni dettagliate su ogni campo (e il valore della proprietà YAML corrispondente) così come appaiono sia nell'editor YAML che in quello visivo.

11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per aggiungere l'azione «Render Amazon ECS task definition» utilizzando l'editor YAML

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
 2. Scegliere il progetto.
 3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
 5. Scegli Modifica.
 6. Scegli YAML.
 7. In alto a sinistra, scegli + Azioni per aprire il catalogo delle azioni.
 8. Dall'elenco a discesa, scegli Amazon. CodeCatalyst
 9. Cerca l'azione di definizione dell'attività Render Amazon ECS ed esegui una delle seguenti operazioni:
 - Scegli il segno più (+) per aggiungere l'azione al diagramma del flusso di lavoro e apri il relativo riquadro di configurazione.
- Or
- Scegli la definizione delle attività di Render Amazon ECS. Viene visualizzata la finestra di dialogo con i dettagli dell'azione. In questa finestra di dialogo:
 - (Facoltativo) Scegliete Visualizza sorgente per [visualizzare il codice sorgente dell'azione](#).
 - Scegli Aggiungi al flusso di lavoro per aggiungere l'azione al diagramma del flusso di lavoro e aprire il relativo riquadro di configurazione.
10. Modifica le proprietà del codice YAML in base alle tue esigenze. Una spiegazione di ogni proprietà disponibile è fornita in. [Azione «Render Amazon ECS task definition» YAML](#)
 11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.

12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Fasi successive

Dopo aver aggiunto l'azione di rendering, aggiungi l'azione Deploy to Amazon ECS al tuo flusso di lavoro seguendo le istruzioni riportate in [Distribuzione su Amazon ECS con un flusso di lavoro](#). Mentre aggiungi l'azione di distribuzione, procedi come segue:

1. Nella scheda Input dell'azione di distribuzione, in Artefatti - opzionale, selezionate l'artefatto generato dall'azione di rendering. Contiene il file di definizione dell'attività aggiornato.

Per ulteriori informazioni sugli artefatti, vedi [Condivisione di artefatti e file tra azioni](#).

2. Nella scheda Configurazione dell'azione di distribuzione, nel campo Definizione del task, specifica la seguente variabile di azione: `${action-name.task-definition}` *action-name* dov'è il nome dell'azione di rendering, ad esempio, `RenderTaskDef`. L'azione di rendering imposta questa variabile sul nuovo nome del file di definizione dell'attività.

Per ulteriori informazioni sulle variabili, consultate [Utilizzo delle variabili nei flussi di lavoro](#).

Per ulteriori informazioni su come configurare l'azione di distribuzione, vedere il flusso di lavoro di [esempio](#) precedente.

Visualizzazione del file di definizione delle attività aggiornato

Puoi visualizzare il nome e il contenuto del file di definizione delle attività aggiornato.

Per visualizzare il nome del file di definizione dell'attività aggiornato, dopo l'elaborazione dell'azione di definizione dell'attività di Render Amazon ECS.

1. Trova l'esecuzione che include un'azione di rendering completata:
 - a. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
 - b. Scegliere il progetto.
 - c. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - d. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
 - e. Scegliete un'esecuzione che includa l'azione di rendering completata.

2. Nel diagramma del flusso di lavoro, scegliete l'azione di rendering.
3. Scegliete Output.
4. Scegliete Variabili.
5. Viene visualizzato il nome del file di definizione dell'attività. Sembra simile `task-definition--259-0a2r7gx1TF5X-.json`.

Per visualizzare il contenuto del file di definizione delle attività aggiornato

1. Trova l'esecuzione che include un'azione di rendering completata:
 - a. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
 - b. Scegliere il progetto.
 - c. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - d. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
 - e. Scegliete un'esecuzione che includa l'azione di rendering completata.
2. Nell'esecuzione del flusso di lavoro, in alto, accanto a Visual e YAML, scegli Output del flusso di lavoro.
3. Nella sezione Artefatti, scegli Scarica accanto all'elemento che contiene il file di definizione dell'attività aggiornato. Questo artefatto avrà una colonna Prodotto da impostata sul nome dell'azione di rendering.
4. Apri il file.zip per visualizzare il file.json di definizione dell'attività.

Variabili «Render Amazon ECS task definition»

L'azione di definizione dell'attività Render Amazon ECS produce e imposta le seguenti variabili in fase di esecuzione. Queste sono note come variabili predefinite.

Per informazioni su come fare riferimento a queste variabili in un flusso di lavoro, vedere. [Utilizzo di variabili predefinite](#)

Chiave	Valore
definizione dell'attività	Il nome assegnato al file di definizione dell'attività che è stato aggiornato dall'azione di definizione dell'attività Render Amazon ECS. Il nome segue il formato <code>task-definition-<i>random-string</i>.json</code>. Ad esempio: <code>task-definition--259-0a2r7gx1TF5Xr.json</code>

Azione «Render Amazon ECS task definition» YAML

Di seguito è riportata la definizione YAML dell'azione di definizione dell'attività Render Amazon ECS. Per informazioni su come utilizzare questa azione, consulta [Modifica della definizione di un'attività Amazon ECS](#)

Questa definizione di azione esiste come sezione all'interno di un file di definizione del flusso di lavoro più ampio. Per ulteriori informazioni su questo file, consulta [Definizione YAML del flusso di lavoro](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visivo. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

```
# The workflow definition starts here.
# See Proprietà di primo livello for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSTaskDefinition\_nn:
  Identifier: aws/ecs-render-task-definition@v1
```

DependsOn:

- *build-action*

Compute:

Type: *EC2 | Lambda*

Fleet: *fleet-name*

Timeout: *timeout-minutes*

Inputs:

Specify a source or an artifact, but not both.

Sources:

- *source-name-1*

Artifacts:

- *task-definition-artifact*

Variables:

- Name: *variable-name-1*
Value: *variable-value-1*
- Name: *variable-name-2*
Value: *variable-value-2*

Configuration

task-definition: *task-definition-path*

container-definition-name: *container-definition-name*

image: *docker-image-name*

environment-variables:

- *variable-name-1=variable-value-1*
- *variable-name-2=variable-value-2*

Outputs:Artifacts:

- Name: *TaskDefArtifact*
- Files: "task-definition*"

Variables:

- *task-definition*

ECSRenderTaskDefinition

(Obbligatorio)

Specificare il nome dell'azione. Tutti i nomi delle azioni devono essere univoci all'interno del flusso di lavoro. I nomi delle azioni sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle azioni.

Default: ECSRenderTaskDefinition_nn.

Interfaccia utente corrispondente: scheda di configurazione/nome dell'azione

Identifier

(*ECSRenderTaskDefinition*/Identifier)

(Obbligatorio)

Identifica l'azione. Non modificate questa proprietà a meno che non vogliate cambiare la versione. Per ulteriori informazioni, consulta [Specificare la versione dell'azione da utilizzare](#).

Default: aws/ecs-render-task-definition@v1.

Interfaccia utente corrispondente: Workflow diagram/ ECSRenderTaskDefinition _nn/ aws/ @v1 label ecs-render-task-definition

DependsOn

(*ECSRenderTaskDefinition*/DependsOn)

(Facoltativo)

Specificate un'azione, un gruppo di azioni o un gate che deve essere eseguito correttamente per eseguire questa azione.

Per ulteriori informazioni sulla funzionalità «dipende da», vedere. [Azioni di sequenziamento](#)

Interfaccia utente corrispondente: scheda Ingressi/Dipende da - opzionale

Compute

(*ECSRenderTaskDefinition*/Compute)

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Interfaccia utente corrispondente: nessuna

Type

(*ECSRenderTaskDefinition*/Compute/Type)

(Obbligatorio se [Compute](#) incluso)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: scheda di configurazione/tipo di calcolo

Fleet

(*ECSRenderTaskDefinition*/Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand:,. Linux.x86-64.Large Linux.x86-64.XLarge Per ulteriori informazioni sulle flotte on-demand, vedere. [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere. [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omesso, l'impostazione predefinita è. Linux.x86-64.Large

Interfaccia utente corrispondente: scheda di configurazione/parco di calcolo

Timeout

(*ECSRenderTaskDefinition*/Timeout)

(Facoltativo)

Specificate il periodo di tempo, in minuti (editor YAML) o ore e minuti (editor visivo), che l'azione può essere eseguita prima CodeCatalyst che termini l'azione. Il minimo è 5 minuti e il massimo è descritto in. [Quote per i flussi di lavoro in CodeCatalyst](#) Il timeout predefinito è lo stesso del timeout massimo.

Interfaccia utente corrispondente: scheda di configurazione/timeout - opzionale

Inputs

(*ECSRenderTaskDefinition*/Inputs)

(Facoltativo)

La Inputs sezione definisce i dati ECSRenderTaskDefinition necessari durante l'esecuzione di un flusso di lavoro.

Note

È consentito un solo input (sorgente o artefatto) per azione di definizione delle attività di Render Amazon ECS. Le variabili non vengono conteggiate per questo totale.

Interfaccia utente corrispondente: scheda Ingressi

Sources

(*ECSRenderTaskDefinition*/Inputs/Sources)

(Richiesto se il file di definizione delle attività è archiviato in un repository di origine)

Se il file di definizione delle attività è archiviato in un repository di origine, specifica l'etichetta di tale repository di origine. Attualmente, l'unica etichetta supportata è. `WorkflowSource`

Se il file di definizione dell'attività non è contenuto in un repository di origine, deve risiedere in un artefatto generato da un'altra azione.

Per ulteriori informazioni sulle origini, consulta [Connessione degli archivi di origine ai flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Sources - opzionale

Artifacts - input

(*ECSRenderTaskDefinition*/Inputs/Artifacts)

(Richiesto se il file di definizione dell'attività è memorizzato in un elemento [di output di un'azione precedente](#))

Se il file di definizione dell'attività che desideri distribuire è contenuto in un artefatto generato da un'azione precedente, specifica l'artefatto qui. Se il file di definizione dell'attività non è contenuto in un artefatto, deve risiedere nel repository di origine.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda di configurazione/artefatti - opzionale

Variables - input

(*ECSRenderTaskDefinition*/Inputs/Variables)

(Obbligatorio)

Specificate una sequenza di name/value coppie che definisca le variabili di input che desiderate rendere disponibili all'azione. I nomi delle variabili sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle variabili.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda Input/Variabili - opzionale

Configuration

(*ECSRenderTaskDefinition*/Configuration)

(Obbligatorio)

Una sezione in cui è possibile definire le proprietà di configurazione dell'azione.

Interfaccia utente corrispondente: scheda Configurazione

task-definition

(*ECSRenderTaskDefinition*/Configuration/task-definition)

(Obbligatorio)

Specificare il percorso di un file di definizione delle attività esistente. Se il file si trova nel repository di origine, il percorso è relativo alla cartella principale del repository di origine. Se il file si trova in un elemento di un'azione precedente del flusso di lavoro, il percorso è relativo alla cartella principale dell'elemento. Per ulteriori informazioni sui file di definizione delle attività, consulta [Definizioni delle attività](#) nella Amazon Elastic Container Service Developer Guide.

Interfaccia utente corrispondente: scheda di configurazione/definizione dell'attività

container-definition-name

(*ECSRenderTaskDefinition*/Configuration/container-definition-name)

(Obbligatorio)

Specificate il nome del contenitore in cui verrà eseguita l'immagine Docker. Puoi trovare questo nome nel name campo `containerDefinitions`, del file di definizione dell'attività. Per ulteriori informazioni, consulta [Name](#) nella Amazon Elastic Container Service Developer Guide.

Interfaccia utente corrispondente: scheda di configurazione/nome del contenitore

image

(*ECSRenderTaskDefinition*/Configuration/image)

(Obbligatorio)

Specificate il nome dell'immagine Docker che desiderate che l'azione di definizione delle attività di Render Amazon ECS aggiunga al file di definizione dell'attività. L'azione aggiunge questo nome al `image` campo `containerDefinitions`, nel file di definizione dell'attività. Se nel `image` campo esiste già un valore, l'azione lo sovrascrive. È possibile includere variabili nel nome dell'immagine.

Esempi:

Se si specifica `MyDockerImage:${WorkflowSource.CommitId}`, l'azione viene aggiunta `MyDockerImage: commit-id` al file di definizione dell'attività, dove *commit-id* è presente un ID di commit generato in fase di esecuzione dal flusso di lavoro.

Se lo specifichiamo `my-ecr-repo/image-repo:${date +%m-%d-%y-%H-%m-%s}`, l'azione aggiunge `my-ecr-repo /image-repo: date +%m-%d-%y-%H-%m-%s` al file di definizione dell'attività, dove *my-ecr-repo* è l'URI di un Amazon Elastic Container Registry (ECR) ed *date +%m-%d-%y-%H-*

`%m-%s` è un timestamp nel formato month-day-year-hour-minute-second generato in fase di esecuzione dal flusso di lavoro.

Per ulteriori informazioni sul `image` campo, consulta [Image](#) nella Amazon Elastic Container Service Developer Guide. Per ulteriori informazioni sulle variabili, consulta [Utilizzo delle variabili nei flussi di lavoro](#).

Interfaccia utente corrispondente: scheda di configurazione/nome immagine

`environment-variables`

(ECSRenderTaskDefinition/Configuration/environment-variables)

(Obbligatorio)

Specificate le variabili di ambiente che desiderate che l'azione di definizione delle attività di Render Amazon ECS aggiunga al file di definizione dell'attività. L'azione aggiunge le variabili al `environment` campo `containerDefinitions`, nel file di definizione delle attività. Se nel file esistono già delle variabili, l'azione sovrascrive i valori delle variabili esistenti e aggiunge nuove variabili. Per ulteriori informazioni sulle variabili di ambiente di Amazon ECS, consulta [Specificare le variabili di ambiente](#) nella Amazon Elastic Container Service Developer Guide.

Interfaccia utente corrispondente: scheda di configurazione/variabili di ambiente - opzionale

Outputs

(ECSRenderTaskDefinition/Outputs)

(Obbligatorio)

Definisce i dati emessi dall'azione durante l'esecuzione di un flusso di lavoro.

Interfaccia utente corrispondente: scheda Output

Artifacts

(ECSRenderTaskDefinition/Outputs/Artifacts)

(Obbligatorio)

Specificate gli artefatti generati dall'azione. È possibile fare riferimento a questi artefatti come input in altre azioni.

Per ulteriori informazioni sugli artefatti, inclusi esempi, vedere. [Condivisione di artefatti e file tra azioni](#)

Interfaccia utente corrispondente: scheda Outputs/Artifacts

Name

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Name)

(Obbligatorio)

Specificare il nome dell'elemento che conterrà il file di definizione dell'attività aggiornato. Il valore predefinito è `MyTaskDefinitionArtifact`. È quindi necessario specificare questo artefatto come input nell'azione Deploy to Amazon ECS. Per capire come aggiungere questo artefatto come input all'azione Deploy to Amazon ECS, consulta. [Esempio: modificare un taskdef di Amazon ECS](#)

Interfaccia utente corrispondente: Outputs tab/Artifacts/ Name

Files

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Files)

(Obbligatorio)

Specificate i file da includere nell'artefatto. È necessario specificare `task-definition-*` in modo che venga incluso il file di definizione delle attività aggiornatotask-definition-, che inizia con.

Interfaccia utente corrispondente: scheda Output/Artifacts/ Files

Variables

(*ECSRenderTaskDefinition*/Outputs/Variables)

(Obbligatorio)

Specificate il nome di una variabile che deve essere impostata dall'azione di rendering. L'azione di rendering imposterà il valore di questa variabile sul nome del file di definizione dell'attività aggiornato (ad esempio,task-definition-random-string.json). È quindi necessario specificare questa variabile nella proprietà Task definition (visual editor) o task-definition (yaml editor) dell'azione Deploy to Amazon ECS. Per informazioni su come aggiungere questa variabile all'azione Deploy to Amazon ECS, consulta. [Esempio: modificare un taskdef di Amazon ECS](#)

Impostazione predefinita: task-definition

Interfaccia utente corrispondente: scheda Outputs/Variables/ Campo Nome

Utilizzo delle variabili nei flussi di lavoro

Una variabile è una coppia chiave-valore che contiene informazioni a cui puoi fare riferimento nel tuo flusso di lavoro Amazon CodeCatalyst . La parte relativa al valore della variabile viene sostituita con un valore effettivo durante l'esecuzione del flusso di lavoro.

Esistono due tipi di variabili che è possibile utilizzare in un flusso di lavoro:

- Variabili definite dall'utente: si tratta di coppie chiave-valore definite dall'utente.
- Variabili predefinite: si tratta di coppie chiave-valore emesse automaticamente da un flusso di lavoro. Non è necessario che tu le definisca.

Per ulteriori informazioni sui flussi di lavoro, consulta [Crea, testa e distribuisce con i flussi di lavoro](#).

Note

CodeCatalyst supporta anche i [parametri GitHub di output](#), che si comportano come variabili e possono essere referenziati in altre azioni. Per ulteriori informazioni, consulta [Esportazione dei parametri GitHub di output](#) e [Parametri di GitHub output di riferimento](#).

Argomenti

- [Utilizzo di variabili definite dall'utente](#)
- [Utilizzo di variabili predefinite](#)

Utilizzo di variabili definite dall'utente

Le variabili definite dall'utente sono coppie chiave-valore definite dall'utente. Esistono due tipi di elenchi:

- Variabili in testo semplice o semplicemente variabili: si tratta di coppie chiave-valore che vengono definite in testo semplice all'interno del file di definizione del flusso di lavoro.
- Segreti: si tratta di coppie chiave-valore che definisci in una pagina Secrets separata della console Amazon CodeCatalyst . La chiave (nome) è un'etichetta pubblica e il valore contiene le

informazioni che desideri mantenere private. La chiave viene specificata solo nel file di definizione del flusso di lavoro. Utilizzate i segreti al posto delle password e di altre informazioni sensibili nel file di definizione del flusso di lavoro.

Note

Per brevità, questa guida utilizza il termine *variabile* per indicare una variabile in testo non crittografato.

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#)

Argomenti

- [Esempi di variabili](#)
- [Definizione di una variabile](#)
- [Definire un segreto](#)
- [Esportazione di una variabile in modo che altre azioni possano utilizzarla](#)
- [Fare riferimento a una variabile nell'azione che la definisce](#)
- [Fare riferimento a un output variabile mediante un'altra azione](#)
- [Fare riferimento a un segreto](#)

Esempi di variabili

Gli esempi seguenti mostrano come definire e fare riferimento alle variabili nel file di definizione del flusso di lavoro.

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Esempi

- [Esempio: definizione di una variabile utilizzando la proprietà Inputs](#)
- [Esempio: definizione di una variabile utilizzando la proprietà Steps](#)
- [Esempio: esportazione di una variabile utilizzando la proprietà Outputs](#)
- [Esempio: riferimento a una variabile definita nella stessa azione](#)
- [Esempio: riferimento a una variabile definita in un'altra azione](#)

- [Esempio: fare riferimento a un segreto](#)

Esempio: definizione di una variabile utilizzando la proprietà Inputs

L'esempio seguente mostra come definire due variabili VAR1 e VAR2, in una Inputs sezione.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: "My variable 1"
        - Name: VAR2
          Value: "My variable 2"
```

Esempio: definizione di una variabile utilizzando la proprietà Steps

L'esempio seguente mostra come definire una DATE variabile nella Steps sezione in modo esplicito.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: DATE=$(date +%m-%d-%y)
```

Esempio: esportazione di una variabile utilizzando la proprietà Outputs

L'esempio seguente mostra come definire due variabili REPOSITORY-URI e TIMESTAMP come esportarle utilizzando la Outputs sezione.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: REPOSITORY-URI
          Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-repo
    Configuration:
```

```
Steps:
  - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
Outputs:
Variables:
  - REPOSITORY-URI
  - TIMESTAMP
```

Esempio: riferimento a una variabile definita nella stessa azione

L'esempio seguente mostra come specificare una VAR1 variabile inMyBuildAction, e quindi farvi riferimento nella stessa azione utilizzando\$VAR1.

```
Actions:
  MyBuildAction:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: my-value
    Configuration:
      Steps:
        - Run: $VAR1
```

Esempio: riferimento a una variabile definita in un'altra azione

L'esempio seguente mostra come specificare una TIMESTAMP variabile inBuildActionA, esportarla utilizzando la Outputs proprietà e quindi farvi riferimento in BuildActionB using\${BuildActionA.TIMESTAMP}.

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
    Outputs:
      Variables:
        - TIMESTAMP
  BuildActionB:
    Identifier: aws/build@v1
    Configuration:
      Steps:
```

```
- Run: docker build -t my-ecr-repo/image-repo:latest .
- Run: docker tag my-ecr-repo/image-repo:${BuildActionA.TIMESTAMP}

# Specifying just '$TIMESTAMP' here will not work
# because TIMESTAMP is not a variable
# in the BuildActionB action.
```

Esempio: fare riferimento a un segreto

L'esempio seguente mostra come fare riferimento a un my-password segreto. my-passwordQuesta è la chiave del segreto. La chiave di questo segreto e il valore della password corrispondente devono essere specificati nella pagina Segreti della CodeCatalyst console prima di essere utilizzati nel file di definizione del flusso di lavoro. Per ulteriori informazioni, consulta [Mascherare i dati utilizzando segreti](#).

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: curl -u LiJuan:${Secrets.my-password} https://example.com
```

Definizione di una variabile

È possibile definire le variabili in due modi:

- Nella Inputs sezione di un'azione del flusso di lavoro, consulta [Per definire una variabile nella sezione «Ingressi»](#)
- Nella Steps sezione di un'azione del flusso di lavoro, consulta [Per definire una variabile nella sezione «Passaggi»](#)

Note

Il Steps metodo funziona solo con le azioni CodeCatalyst build, test e GitHub Actions, poiché queste sono le uniche azioni che includono una Steps sezione.

Per alcuni esempi, consulta [Esempi di variabili](#).

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Visual

Per definire una variabile nella sezione 'Input' (editor visuale)

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione in cui desideri impostare la variabile.
8. Scegliete Ingressi.
9. In Variabili, facoltativo, scegliete Aggiungi variabile, quindi effettuate le seguenti operazioni:

Specificate una sequenza di name/value coppie che definisca le variabili di input che desiderate rendere disponibili per l'azione. I nomi delle variabili sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi delle variabili.

Per ulteriori informazioni sulle variabili, inclusi esempi, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per definire una variabile nella sezione 'Input' (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.

4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione del flusso di lavoro, aggiungi un codice simile al seguente:

```
action-name:
  Inputs:
    Variables:
      - Name: variable-name
        Value: variable-value
```

Per ulteriori esempi, consulta [Esempi di variabili](#). Per ulteriori informazioni, consulta la sezione [Definizione YAML del flusso di lavoro](#) dedicata alla tua azione.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Visual

Per definire una variabile nella sezione «Steps» (editor visivo)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione in cui desideri impostare la variabile.
8. Scegliere Configuration (Configurazione).
9. Nei comandi Shell o nelle GitHubazioni YAML, a seconda di quale sia disponibile, definisci una variabile nelle azioni, in modo esplicito o Steps implicito.

- Per definire la variabile in modo esplicito, includila in un comando bash direttamente nella sezione. Steps
- Per definire una variabile in modo implicito, specificala in un file a cui si fa riferimento nella sezione dell'azione. Steps

Per alcuni esempi, consulta [Esempi di variabili](#). Per ulteriori informazioni, consulta la sezione dedicata [Definizione YAML del flusso di lavoro](#) all'azione.

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per definire una variabile nella sezione 'Steps' (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un'azione del flusso di lavoro, definisci una variabile nella Steps sezione dell'azione, in modo esplicito o implicito.
 - Per definire la variabile in modo esplicito, includila in un comando bash direttamente nella sezione. Steps
 - Per definire una variabile in modo implicito, specificala in un file a cui si fa riferimento nella sezione dell'azione. Steps

Per alcuni esempi, consulta [Esempi di variabili](#). Per ulteriori informazioni, consulta la sezione dedicata [Definizione YAML del flusso di lavoro](#) all'azione.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.

9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Definire un segreto

È possibile definire un segreto nella pagina Segreti della CodeCatalyst console. Per ulteriori informazioni, consulta [Mascherare i dati utilizzando segreti](#).

Ad esempio, potresti definire un segreto simile al seguente:

- Nome (chiave): **my-password**
- Valore: **^*H3#!b9**

Dopo aver definito il segreto, è possibile specificare la chiave del segreto (**my-password**) nel file di definizione del flusso di lavoro. Per un esempio su come Eseguire questa operazione, consulta [Esempio: fare riferimento a un segreto](#).

Esportazione di una variabile in modo che altre azioni possano utilizzarla

Utilizza le seguenti istruzioni per esportare una variabile da un'azione in modo da potervi fare riferimento in altre azioni.

Prima di esportare una variabile, tenete presente quanto segue:

- Se devi solo fare riferimento alla variabile all'interno dell'azione in cui è definita, non devi esportarla.
- Non tutte le azioni supportano l'esportazione di variabili. Per determinare se la tua azione supporta questa funzionalità, esegui le istruzioni dell'editor visivo che seguono e verifica se l'azione include un pulsante Variabili nella scheda Output. In caso affermativo, è supportata l'esportazione di variabili.
- Per esportare una variabile da un' GitHub azione, vedere [Esportazione dei parametri GitHub di output](#).

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Prerequisito

Assicurati di aver definito la variabile che desideri esportare. Per ulteriori informazioni, consulta [Definizione di una variabile](#).

Visual

Per esportare una variabile (editor visuale)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione da cui desideri esportare la variabile.
8. Scegliete Output.
9. In Variabili, facoltativo, scegliete Aggiungi variabile, quindi effettuate le seguenti operazioni:

Specificate il nome di una variabile che desiderate che l'azione esporti. Questa variabile deve essere già definita nella Steps sezione Inputs o della stessa azione.

10. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
11. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

YAML

Per esportare una variabile (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.

7. Nell'azione da cui desideri esportare la variabile, aggiungi un codice simile al seguente:

```
action-name:  
  Outputs:  
    Variables:  
      - Name: variable-name
```

Per ulteriori esempi, consulta [Esempi di variabili](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Fare riferimento a una variabile nell'azione che la definisce

Utilizzate le seguenti istruzioni per fare riferimento a una variabile nell'azione che la definisce.

Note

Per fare riferimento a una variabile generata da un' GitHub azione, vedere [Parametri di GitHub output di riferimento](#).

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Prerequisito

Assicurati di aver definito la variabile a cui vuoi fare riferimento. Per ulteriori informazioni, consulta [Definizione di una variabile](#).

Visual

Non disponibile Scegli YAML per visualizzare le istruzioni YAML.

YAML

Per fare riferimento a una variabile nell'azione che la definisce

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.

3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell' CodeCatalyst azione che definisce la variabile a cui vuoi fare riferimento, aggiungi la variabile usando la seguente sintassi bash:

```
$variable-name
```

Esempio:

```
MyAction:
  Configuration:
    Steps:
      - Run: $variable-name
```

Per ulteriori esempi, consulta [Esempi di variabili](#). Per ulteriori informazioni, consultate le informazioni di riferimento per la vostra azione in [Definizione YAML del flusso di lavoro](#)

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Fare riferimento a un output variabile mediante un'altra azione

Utilizzate le seguenti istruzioni per fare riferimento alle variabili generate da altre azioni.

Note

Per fare riferimento a una variabile in uscita da un' GitHub azione, vedere [Parametri di GitHub output di riferimento](#).

Per ulteriori informazioni sulle variabili, vedere [Utilizzo delle variabili nei flussi di lavoro](#).

Prerequisito

Assicurati di aver esportato la variabile a cui desideri fare riferimento. Per ulteriori informazioni, consulta [Esportazione di una variabile in modo che altre azioni possano utilizzarla](#).

Visual

Non disponibile. Scegli YAML per visualizzare le istruzioni YAML.

YAML

Per fare riferimento a un output variabile mediante un'altra azione (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell' CodeCatalyst azione, aggiungi un riferimento alla variabile utilizzando la seguente sintassi:

```
${action-group-name.action-name.variable-name}
```

Sostituire:

- *action-group-name* con il nome del gruppo di azioni che contiene l'azione che genera la variabile.

Note

È possibile omettere *action-group-name* se non esiste un gruppo di azioni o se la variabile è prodotta da un'azione nello stesso gruppo di azioni.

- *action-name* con il nome dell'azione che genera la variabile.
- *variable-name* con il nome della variabile.

Esempio:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: ${MyFirstAction.TIMESTAMP}
```

Per ulteriori esempi, consulta [Esempi di variabili](#). Per ulteriori informazioni, consulta la sezione [Definizione YAML del flusso di lavoro](#) dedicata alle tue azioni.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Fare riferimento a un segreto

Per istruzioni su come fare riferimento a un segreto nel file di definizione del flusso di lavoro, vedere. [Usare un segreto](#)

Per un esempio, consulta [Esempio: fare riferimento a un segreto](#).

Utilizzo di variabili predefinite

Le variabili predefinite sono coppie chiave-valore emesse automaticamente da un flusso di lavoro e rese disponibili per l'uso nelle azioni del flusso di lavoro.

Per ulteriori informazioni sulle variabili, vedere. [Utilizzo delle variabili nei flussi di lavoro](#)

Argomenti

- [Esempi di riferimento a variabili predefinite](#)
- [Riferimento a una variabile predefinita](#)
- [Determinazione delle variabili predefinite emesse dal flusso di lavoro](#)
- [Elenco di variabili predefinite](#)

Esempi di riferimento a variabili predefinite

Gli esempi seguenti mostrano come fare riferimento a variabili predefinite nel file di definizione del flusso di lavoro.

Per ulteriori informazioni sulle variabili predefinite, vedere. [Utilizzo di variabili predefinite](#)

Esempi

- [Esempio: riferimento alla variabile predefinita "CommitId"](#)
- [Esempio: riferimento alla variabile predefinita BranchName ""](#)

Esempio: riferimento alla variabile predefinita "CommitId"

L'esempio seguente mostra come fare riferimento alla variabile `CommitId` predefinita nell'azione. `MyBuildAction` La `CommitId` variabile viene emessa automaticamente da CodeCatalyst. Per ulteriori informazioni, consulta [Elenco di variabili predefinite](#).

Sebbene l'esempio mostri la variabile utilizzata nell'azione di creazione, è possibile utilizzarla `CommitId` in qualsiasi azione.

```
MyBuildAction:
  Identifier: aws/build@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    Steps:
      #Build Docker image and tag it with a commit ID
      - Run: docker build -t image-repo/my-docker-image:latest .
      - Run: docker tag image-repo/my-docker-image:${WorkflowSource.CommitId}
```

Esempio: riferimento alla variabile predefinita BranchName ""

L'esempio seguente mostra come fare riferimento alla variabile `BranchName` predefinita nell'azione. `CDKDeploy` La `BranchName` variabile viene emessa automaticamente da CodeCatalyst. Per ulteriori informazioni, consulta [Elenco di variabili predefinite](#).

Sebbene l'esempio mostri la variabile utilizzata nell'azione di AWS CDK distribuzione, è possibile utilizzarla `BranchName` in qualsiasi azione.

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v2
  Inputs:
    Sources:
      - WorkflowSource
```

Configuration:`StackName: app-stack-${WorkflowSource.BranchName}`

Riferimento a una variabile predefinita

Puoi fare riferimento a variabili predefinite in qualsiasi azione all'interno di un CodeCatalyst flusso di lavoro Amazon.

Utilizza le seguenti istruzioni per fare riferimento a una variabile predefinita in un flusso di lavoro.

Per ulteriori informazioni sulle variabili predefinite, vedere. [Utilizzo di variabili predefinite](#)

Prerequisito

Determina il nome della variabile predefinita a cui desideri fare riferimento, ad esempio. `CommitId`
Per ulteriori informazioni, consulta [Determinazione delle variabili predefinite emesse dal flusso di lavoro](#).

Visual

Non disponibile. Scegli YAML per visualizzare le istruzioni YAML.

YAML

Per fare riferimento a una variabile predefinita (editor YAML)

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. In un' CodeCatalyst azione, aggiungete il riferimento alla variabile predefinito utilizzando la seguente sintassi:

```
${action-group-name.action-name-or-WorkflowSource.variable-name}
```

Sostituisci:

- *action-group-name* con il nome del gruppo di azione.

 Note

È possibile omettere *action-group-name* se non esiste un gruppo di azioni o se la variabile è prodotta da un'azione nello stesso gruppo di azioni.

- *action-name-or-WorkflowSource* con:

Il nome dell'azione che genera la variabile.

oppure

WorkflowSource, se la variabile è la CommitId variabile BranchName or.

- *variable-name* con il nome della variabile.

Per esempio:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${MyFirstECSAction.cluster}
```

Un altro esempio:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${WorkflowSource.CommitId}
```

Per ulteriori esempi, consulta [Esempi di riferimento a variabili predefinite](#). Per ulteriori informazioni, consulta la sezione [Definizione YAML del flusso di lavoro](#) dedicata alle tue azioni.

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Determinazione delle variabili predefinite emesse dal flusso di lavoro

Utilizzate la procedura seguente per determinare quali variabili predefinite emette un flusso di lavoro durante l'esecuzione. È quindi possibile fare riferimento a queste variabili all'interno dello stesso flusso di lavoro.

Per ulteriori informazioni sulle variabili predefinite, vedere [Utilizzo di variabili predefinite](#).

Per determinare le variabili predefinite emesse dal flusso di lavoro

- Esegui una di queste operazioni:
 - Esegui il flusso di lavoro una volta. Al termine dell'esecuzione, le variabili emesse dal flusso di lavoro vengono visualizzate nella scheda Variabili della pagina dei dettagli dell'esecuzione. Per ulteriori informazioni, consulta [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).
 - Consulta il [Elenco di variabili predefinite](#). Questo riferimento elenca il nome della variabile (chiave) e il valore per ogni variabile predefinita.

Note

La dimensione totale massima delle variabili di un flusso di lavoro è elencata in [Quote per i flussi di lavoro in CodeCatalyst](#). Se la dimensione totale supera il massimo, l'azione che si verifica dopo il raggiungimento del massimo potrebbe non riuscire.

Elenco di variabili predefinite

Consultate le seguenti sezioni per visualizzare le variabili predefinite prodotte automaticamente dalle CodeCatalyst azioni come parte dell'esecuzione di un flusso di lavoro.

Per ulteriori informazioni sulle variabili predefinite, vedere [Utilizzo di variabili predefinite](#)

Note

[Questo elenco include solo le variabili predefinite emesse dall'origine e dalle azioni. CodeCatalyst](#) Se utilizzi altri tipi di azioni, come le azioni GitHub Actions o

CodeCatalyst Labs, consulta invece. [Determinazione delle variabili predefinite emesse dal flusso di lavoro](#)

Elenco

Note

Non tutte le CodeCatalyst azioni producono variabili predefinite. Se l'azione non è presente nell'elenco, non produce variabili.

- [variabili BranchName " e CommitId "](#)
- [Variabili 'Deploy CloudFormation stack'](#)
- [Variabili «Distribuisci su Amazon ECS»](#)
- [Variabili «Distribuisci su cluster Kubernetes»](#)
- [variabili 'AWS CDK deploy'](#)
- [variabili 'AWS CDK bootstrap'](#)
- [variabili 'AWS Lambda invoke'](#)
- [Variabili «Render Amazon ECS task definition»](#)

Mascherare i dati utilizzando segreti

In alcuni momenti può essere necessario utilizzare dati sensibili, come le credenziali di autenticazione, nei flussi di lavoro. La memorizzazione di questi valori in testo semplice in qualsiasi punto del repository dovrebbe essere evitata perché chiunque abbia accesso all'archivio che contiene il segreto può vederli. Analogamente, questi valori non devono essere utilizzati direttamente in nessuna definizione del flusso di lavoro perché saranno visibili come file nel repository. Con CodeCatalyst, puoi proteggere questi valori aggiungendo un segreto al tuo progetto e quindi facendo riferimento al segreto nel file di definizione del flusso di lavoro. Tieni presente che puoi avere un massimo di cinque segreti per azione.

 Note

I segreti possono essere utilizzati solo per sostituire password e informazioni sensibili nel file di definizione del flusso di lavoro.

Argomenti

- [Creare un segreto](#)
- [Modificare un segreto](#)
- [Usare un segreto](#)
- [Eliminazione di un segreto](#)

Creare un segreto

Utilizzare la procedura seguente per creare un segreto. Il segreto contiene le informazioni riservate che desiderate nascondere alla vista.

 Note

I segreti sono visibili alle azioni e non vengono mascherati quando vengono scritti su un file.

Per creare un segreto

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Secrets.
3. Scegli Crea segreto.
4. Immetti le seguenti informazioni:

Nome

Inserisci un nome per il tuo segreto.

Valore

Inserisci il valore del segreto. Si tratta delle informazioni riservate che desideri nascondere alla vista. Per impostazione predefinita, il valore non viene visualizzato. Per visualizzare il valore, scegli Mostra valore.

Descrizione

(Facoltativo) Inserisci una descrizione per il tuo segreto.

5. Scegli Create (Crea).

Modificare un segreto

Utilizzate la seguente procedura per modificare un segreto.

Per modificare un segreto

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Secrets.
3. Nell'elenco dei segreti, scegli il segreto che desideri modificare.
4. Scegli Modifica.
5. Modificate le seguenti proprietà:

Valore

Immettete il valore del segreto. Questo è il valore che vuoi nascondere alla vista. Per impostazione predefinita, il valore non viene visualizzato.

Descrizione

(Facoltativo) Inserisci una descrizione per il tuo segreto.

6. Scegli Save (Salva).

Usare un segreto

Per utilizzare un segreto in un'azione del flusso di lavoro, è necessario ottenere l'identificatore di riferimento del segreto e utilizzarlo nell'azione del flusso di lavoro.

Argomenti

- [Ottenere l'identificatore di un segreto](#)
- [Fare riferimento a un segreto in un flusso di lavoro](#)

Ottenere l'identificatore di un segreto

Utilizzare la procedura seguente per ottenere l'identificatore di riferimento del segreto. Aggiungerai questo identificatore al tuo flusso di lavoro.

Per ottenere l'identificatore di riferimento del segreto

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Secrets.
3. Nell'elenco dei segreti, trova il segreto che desideri utilizzare.
4. Nella colonna Reference ID, copia l'identificatore del segreto. Di seguito è riportata la sintassi per l'ID di riferimento:

```
${Secrets.<name>}
```

Fare riferimento a un segreto in un flusso di lavoro

Utilizzate la procedura seguente per fare riferimento a un segreto in un flusso di lavoro.

Per fare riferimento a un segreto

1. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
2. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
3. Scegli Modifica.
4. Scegli YAML.
5. Modifica lo YAML per utilizzare l'identificatore del segreto. Ad esempio, per utilizzare un nome utente e una password archiviati come segreti con il `curl` comando, è necessario utilizzare un Run comando simile al seguente:

```
- Run: curl -u <username-secret-identifier>:<password-secret-identifier> https://example.com
```

6. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
7. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Eliminazione di un segreto

Utilizzare la procedura seguente per eliminare un segreto e l'identificatore di riferimento segreto.

Note

Prima di eliminare un segreto, si consiglia di rimuovere l'identificatore di riferimento del segreto da tutte le azioni del flusso di lavoro. Se elimini il segreto senza eliminare l'identificatore di riferimento, l'azione avrà esito negativo alla successiva esecuzione.

Per eliminare l'identificatore di riferimento di un segreto da un flusso di lavoro

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
3. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
4. Scegli Modifica.
5. Scegli YAML.
6. Cerca nel flusso di lavoro la seguente stringa:

```
${Secrets.
```

In questo modo vengono trovati tutti gli identificatori di riferimento di tutti i segreti.

7. Elimina l'identificatore di riferimento del segreto scelto o sostituiscilo con un valore in chiaro.
8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

Eliminare il segreto

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Secrets.
3. Nell'elenco dei segreti, scegli il segreto che desideri eliminare.

4. Scegli Elimina.
5. Inserisci **delete** per confermare l'eliminazione.
6. Scegli Delete (Elimina).

Visualizzazione dello stato di un flusso di lavoro

Potresti voler visualizzare lo stato di un flusso di lavoro per vedere se ci sono problemi di configurazione del flusso di lavoro che devi risolvere o per risolvere le esecuzioni che non si avviano. CodeCatalyst valuta lo stato del flusso di lavoro ogni volta che si crea o si aggiorna il file di definizione del flusso di [lavoro](#) sottostante.

Note

È inoltre possibile visualizzare lo stato di esecuzione del flusso di lavoro, che è diverso dallo stato del flusso di lavoro. Per ulteriori informazioni, consulta [Visualizzazione dello stato e dei dettagli dell'esecuzione del flusso di lavoro](#).

Per un elenco dei possibili stati del flusso di lavoro, consulta [Stati del flusso di lavoro in CodeCatalyst](#).

Per visualizzare lo stato di un flusso di lavoro

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.

Lo stato viene visualizzato con il flusso di lavoro nell'elenco.

5. (Facoltativo) Scegliete il nome del flusso di lavoro e individuate il campo Definizione del flusso di lavoro. Mostra lo stato del flusso di lavoro.

Quote per i flussi di lavoro in CodeCatalyst

La tabella seguente descrive le quote e i limiti per i flussi di lavoro in Amazon. CodeCatalyst

Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Numero massimo di flussi di lavoro per spazio	800
Dimensione massima del file di definizione del flusso di lavoro	256 KB
Numero massimo di file di workflow elaborati in un unico evento di origine	50
Numero massimo di file elaborati in un singolo evento di origine	4.000
Numero massimo di flotte attive per spazio	10
Numero massimo di istanze di elaborazione attive per flotta	20
Numero massimo di artefatti di input per azione	10
Numero massimo di artefatti di output per azione	10
Dimensione totale massima delle variabili di output di una singola azione	120 KB
Lunghezza massima di un valore variabile di uscita	500 caratteri o più, a seconda dell'azione che emette il valore.  Note I valori possono essere troncati se superano il limite dell'azione.
Numero massimo di giorni per conservare gli artefatti generati durante l'esecuzione di un flusso di lavoro	30
Numero massimo di report per azione	50

Numero massimo di casi di test per rapporto di test	20.000
Numero massimo di file per rapporto di copertura del codice	20.000
Numero massimo di risultati dell'analisi della composizione del software per rapporto	20.000
Numero massimo di file per rapporto di analisi statica	20.000
Numero massimo di flussi di lavoro simultanei per spazio	100
Numero massimo di azioni per workflow	50
Numero massimo di azioni eseguite contemporaneamente per flusso di lavoro	50
Numero massimo di azioni eseguite contemporaneamente per spazio	200
Tempo massimo di esecuzione di un'azione	Per le azioni di compilazione e test, il timeout è di 8 ore. Per tutte le altre azioni, il timeout è di 1 ora.
Numero massimo di ambienti associati a un ambiente Account AWS per spazio	5.000
Numero massimo di segreti per azione	5
Numero massimo di segreti per spazio	500.000

Stati di esecuzione del flusso di lavoro

L'esecuzione di un workflow può trovarsi in uno dei seguenti stati:

- Riuscita: il workflow eseguito è stato elaborato correttamente.
- Non riuscita: l'esecuzione di una o più azioni nel flusso di lavoro non è riuscita.
- In corso: l'esecuzione del flusso di lavoro è attualmente in fase di elaborazione.
- Interrotto: una persona ha interrotto l'esecuzione del flusso di lavoro mentre era in corso.
- Interruzione: l'esecuzione del flusso di lavoro è attualmente interrotta.
- Annullato: l'esecuzione del flusso di lavoro è stata annullata CodeCatalyst perché il flusso di lavoro associato è stato eliminato o aggiornato mentre l'esecuzione era in corso.
- Sostituita: [si verifica solo se è stata configurata la modalità di esecuzione sostituita](#). L'esecuzione del flusso di lavoro è stata annullata CodeCatalyst perché è stata sostituita da un'esecuzione successiva del flusso di lavoro.

Stati del flusso di lavoro in CodeCatalyst

Un flusso di lavoro può avere uno dei seguenti stati:

- Valido: [il flusso di lavoro è eseguibile e può essere attivato tramite trigger](#).

Affinché un flusso di lavoro sia contrassegnato come valido, devono essere soddisfatte entrambe le seguenti condizioni:

- Il file di definizione del flusso di lavoro deve essere valido.
- Il flusso di lavoro non deve avere alcun trigger, nessun trigger push o un trigger push che viene eseguito utilizzando i file sul ramo corrente. Per ulteriori informazioni, consulta [Linee guida per l'utilizzo di trigger e filiali](#).
- Non valido: il file di definizione del flusso di lavoro non è valido. Il flusso di lavoro non può essere eseguito manualmente o automaticamente tramite i trigger. I flussi di lavoro non validi vengono visualizzati con una definizione di workflow con un messaggio di **n** errore (o simile) nella CodeCatalyst console.

Affinché un flusso di lavoro venga contrassegnato come non valido, deve essere soddisfatta la seguente condizione:

- Il file di definizione del flusso di lavoro deve essere configurato in modo errato.

Per correggere un file di definizione del flusso di lavoro configurato in modo errato, vedere. [Come posso correggere gli errori «La definizione del flusso di lavoro presenta n errori»?](#)

- Inattivo: la definizione del flusso di lavoro è valida ma non può essere eseguita manualmente o automaticamente tramite i trigger.

Affinché un flusso di lavoro sia contrassegnato come inattivo, devono essere soddisfatte entrambe le seguenti condizioni:

- Il file di definizione del flusso di lavoro deve essere valido.
- Il file di definizione del flusso di lavoro deve includere un trigger push che specifichi un ramo diverso da quello in cui si trova attualmente il file di definizione del flusso di lavoro. Per ulteriori informazioni, consulta [Linee guida per l'utilizzo di trigger e filiali](#).

Per cambiare un flusso di lavoro da Inattivo ad Attivo, consulta. [Come posso correggere i messaggi «Workflow is inactive»?](#)

Note

Se ci sono molti flussi di lavoro nello stato Inattivo, puoi filtrarli dalla visualizzazione. Per filtrare i flussi di lavoro inattivi, scegli il campo Filtra flussi di lavoro nella parte superiore della pagina Flussi di lavoro, scegli Stato, scegli Stato! = Non è uguale e scegli INATTIVO.

Note

Se il flusso di lavoro specifica una risorsa che verrà successivamente rimossa (ad esempio, un archivio di pacchetti), CodeCatalyst non rileverà questa modifica e continuerà a contrassegnare il flusso di lavoro come valido. Questi tipi di problemi verranno rilevati durante l'esecuzione del flusso di lavoro.

Definizione YAML del flusso di lavoro

Di seguito è riportata la documentazione di riferimento per il file di definizione del flusso di lavoro.

Un file di definizione del flusso di lavoro è un file YAML che descrive il flusso di lavoro. Per impostazione predefinita, il file viene archiviato in una `~/.codecatalyst/workflows/` cartella nella radice del repository di [origine](#). Il file può avere un'estensione `.yaml` o `.yml` e l'estensione deve essere minuscola.

Per creare e modificare il file di definizione del flusso di lavoro, puoi usare un editor come vim oppure puoi usare l'editor visivo o l'editor YAML della CodeCatalyst console. Per ulteriori informazioni, consulta [Utilizzo degli editor visivi e YAML della CodeCatalyst console](#).

Note

La maggior parte delle proprietà YAML che seguono hanno elementi dell'interfaccia utente corrispondenti nell'editor visuale. Per cercare un elemento dell'interfaccia utente, usa Ctrl+F. L'elemento verrà elencato con la proprietà YAML associata.

Argomenti

- [Esempio di file di definizione del flusso di lavoro](#)
- [Linee guida e convenzioni sulla sintassi](#)
- [Proprietà di primo livello](#)

Esempio di file di definizione del flusso di lavoro

Di seguito è riportato un esempio di un semplice file di definizione del flusso di lavoro. Include alcune proprietà di primo livello, una Triggers sezione e una Actions sezione con due azioni: Build e Test. Per ulteriori informazioni, consulta [Informazioni sul file di definizione del flusso di lavoro](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: docker build -t MyApp:latest .
  Test:
```

```
Identifier: aws/managed-test@v1
DependsOn:
  - Build
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  Steps:
    - Run: npm install
    - Run: npm run test
```

Linee guida e convenzioni sulla sintassi

Questa sezione descrive le regole di sintassi per il file di definizione del flusso di lavoro, nonché le convenzioni di denominazione utilizzate in questa documentazione di riferimento.

Linee guida sulla sintassi YAML

Il file di definizione del flusso di lavoro è scritto in YAML e segue la specifica [YAML 1.1, quindi tutto ciò che è consentito in quella specifica](#) è consentito anche nel flusso di lavoro YAML. Se non conosci YAML, ecco alcune linee guida rapide per assicurarti di fornire codice YAML valido.

- Distinzione tra maiuscole e minuscole: il file di definizione del flusso di lavoro fa distinzione tra maiuscole e minuscole, quindi assicurati di utilizzare la distinzione tra maiuscole e minuscole mostrata in questa documentazione.
- Caratteri speciali: si consiglia di utilizzare virgolette o virgolette doppie attorno ai valori delle proprietà che includono uno dei seguenti caratteri speciali: `{},[,],,,,*,#,<?|,>-,,,,,,= e ! % @ : ` ,`

Se non includi le virgolette, i caratteri speciali elencati in precedenza potrebbero essere interpretati in modo inaspettato.

- Nomi di proprietà: i nomi delle proprietà (al contrario dei valori delle proprietà) sono limitati ai caratteri alfanumerici (a-z, A-Z, 0-9), ai trattini (-) e ai caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare virgolette o virgolette doppie per abilitare caratteri e spazi speciali nei nomi delle proprietà.

Non consentito:

```
'My#Build@action'
```

My#Build@action

My Build Action

Consentito:

My-Build-Action_1

- Codici di escape: se il valore della proprietà include codici di escape (ad esempio, `\n o\t`), segui queste linee guida:
 - Usa virgolette singole per restituire il codice di escape come stringa. Ad esempio `'my string \n my string'`, restituisce la stringa `my string \n my string`.
 - Usa le virgolette doppie per analizzare il codice di escape. Ad esempio `"my string \n my new line"`, restituisce:

```
my string
my new line
```

- Commenti: prefigura i commenti con `#`.

Esempio:

```
Name: MyWorkflow
# This is a comment.
SchemaVersion: 1.0
```

- Triple dash (`---`): non utilizzare `---` nel codice YAML. CodeCatalyst ignora tutto ciò che segue.
`---`

Convenzioni di denominazione

In questa guida, utilizziamo i termini proprietà e sezione per fare riferimento agli elementi principali di un file di definizione del flusso di lavoro.

- Una proprietà è qualsiasi elemento che include i due punti (`:`). Ad esempio, nel seguente frammento di codice, tutte le seguenti sono proprietà: `Name`, `SchemaVersion`, `RunMode`, `TriggersType`, e `Branches`
- Una sezione è qualsiasi proprietà che presenta proprietà secondarie. Nel seguente frammento di codice, c'è una sezione. `Triggers`

 Note

In questa guida, le «sezioni» sono talvolta chiamate «proprietà» e viceversa, a seconda del contesto.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

Proprietà di primo livello

Di seguito è riportata la documentazione di riferimento per le proprietà di primo livello nel file di definizione del flusso di lavoro.

```
# Name
Name: workflow-name

# Schema version
SchemaVersion: 1.0

# Run mode
RunMode: QUEUED/SUPERSEDED/PARALLEL

# Compute
Compute:
...

# Triggers
Triggers:
...

# Actions
Actions:
...
```

Name

(Obbligatorio)

Nome del flusso di lavoro. Il nome del flusso di lavoro viene visualizzato nell'elenco dei flussi di lavoro e menzionato nelle notifiche e nei registri. Il nome del flusso di lavoro e il nome del file di definizione del flusso di lavoro possono corrispondere oppure è possibile denominarli in modo diverso. I nomi dei flussi di lavoro non devono essere necessariamente univoci. I nomi dei flussi di lavoro sono limitati a caratteri alfanumerici (a-z, A-Z, 0-9), trattini (-) e caratteri di sottolineatura (_). Gli spazi non sono consentiti. Non è possibile utilizzare le virgolette per abilitare caratteri e spazi speciali nei nomi dei flussi di lavoro.

Interfaccia utente corrispondente: editor/Workflow proprietà visive/nome del flusso di lavoro

SchemaVersion

(Obbligatorio)

La versione dello schema della definizione del flusso di lavoro. Attualmente, l'unico valore valido è 1.0.

Interfaccia utente corrispondente: nessuna

RunMode

(Facoltativo)

Come CodeCatalyst gestisce più esecuzioni. È possibile utilizzare uno dei seguenti valori:

- **QUEUED**— Le esecuzioni multiple vengono messe in coda ed eseguite una dopo l'altra. È possibile avere fino a 50 esecuzioni in coda.
- **SUPERSEDED**— Le esecuzioni multiple vengono messe in coda ed eseguite una dopo l'altra. Una coda può avere una sola esecuzione, quindi se due esecuzioni finiscono insieme nella stessa coda, l'esecuzione successiva sostituisce (sostituisce) l'esecuzione precedente e l'esecuzione precedente viene annullata.
- **PARALLEL**— Vengono eseguite più esecuzioni contemporaneamente.

Se questa proprietà viene omessa, l'impostazione predefinita è **QUEUED**.

Per ulteriori informazioni, consulta [Configurazione del comportamento di accodamento delle esecuzioni](#).

Interfaccia utente corrispondente: editor/Workflow proprietà visive/modalità Avanzata/Esegui

Compute

(Facoltativo)

Il motore di calcolo utilizzato per eseguire le azioni del flusso di lavoro. È possibile specificare l'elaborazione a livello di flusso di lavoro o a livello di azione, ma non entrambi. Se specificata a livello di flusso di lavoro, la configurazione di calcolo si applica a tutte le azioni definite nel flusso di lavoro. A livello di flusso di lavoro, puoi anche eseguire più azioni sulla stessa istanza. Per ulteriori informazioni, consulta [Condivisione dell'elaborazione tra le azioni](#).

Per ulteriori informazioni sull'elaborazione, consulta [Configurazione delle immagini di calcolo e di runtime](#).

Interfaccia utente corrispondente: nessuna

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
  SharedInstance: true | false
```

Type

(Compute/Type)

(Obbligatorio se Compute è impostato)

Il tipo di motore di calcolo. È possibile utilizzare uno dei seguenti valori:

- EC2(editor visivo) o EC2 (editor YAML)

Ottimizzato per la flessibilità durante le operazioni.

- Lambda (editor visivo) o Lambda (editor YAML)

Velocità di avvio delle azioni ottimizzate.

Per ulteriori informazioni sui tipi di calcolo, consulta [Tipi di calcolo](#).

Interfaccia utente corrispondente: editor/Workflow proprietà visive/avanzate/tipo di calcolo

Fleet

(Compute/Fleet)

(Facoltativo)

Specificate la macchina o il parco macchine che eseguiranno il flusso di lavoro o le azioni del flusso di lavoro. Con le flotte on-demand, all'avvio di un'azione, il flusso di lavoro fornisce le risorse necessarie e le macchine vengono distrutte al termine dell'azione. Esempi di flotte on-demand: `Linux.x86-64.Large` `Linux.x86-64.XLarge` Per ulteriori informazioni sulle flotte on-demand, vedere [Proprietà del parco veicoli su richiesta](#)

Con le flotte assegnate, puoi configurare una serie di macchine dedicate per eseguire le azioni del flusso di lavoro. Queste macchine rimangono inattive, pronte a elaborare immediatamente le azioni. Per ulteriori informazioni sulle flotte rifornite, vedere [Proprietà del parco veicoli assegnate](#)

Se Fleet viene omesso, l'impostazione predefinita è `Linux.x86-64.Large`

Per ulteriori informazioni sulle flotte di elaborazione, consulta [Flotte di calcolo](#)

Interfaccia utente corrispondente: editor/Workflow proprietà visive/Advanced/Compute fleet

SharedInstance

(Compute/SharedInstance)

(Facoltativo)

Specificate la capacità di condivisione del calcolo per le vostre azioni. Con la condivisione del calcolo, le azioni in un flusso di lavoro vengono eseguite sulla stessa istanza (immagine dell'ambiente di runtime). È possibile utilizzare uno dei seguenti valori:

- `TRUE` significa che l'immagine dell'ambiente di runtime è condivisa tra le azioni del flusso di lavoro.
- `FALSE` significa che viene avviata e utilizzata un'immagine di ambiente di runtime separata per ogni azione in un flusso di lavoro, quindi non è possibile condividere risorse come artefatti e variabili senza una configurazione aggiuntiva.

Per ulteriori informazioni sulla condivisione del calcolo, consulta [Condivisione dell'elaborazione tra le azioni](#)

Interfaccia utente corrispondente: nessuna

Triggers

(Facoltativo)

Una sequenza di uno o più trigger per questo flusso di lavoro. Se non viene specificato un trigger, è necessario avviare manualmente il flusso di lavoro.

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Interfaccia utente corrispondente: editor/workflow diagramma visivo/Trigger

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Triggers:
- Type: PUSH
  Branches:
    - branch-name
  FilesChanged:
    - folder1/file
    - folder2/

- Type: PULLREQUEST
  Events:
    - OPEN
    - CLOSED
    - REVISION
  Branches:
    - branch-name
  FilesChanged:
    - file1.txt

- Type: SCHEDULE
  # Run the workflow at 10:15 am (UTC+0) every Saturday
  Expression: "15 10 ? * 7 *"
  Branches:
    - branch-name
```

Type

(Triggers/Type)

(Richiesto se impostato `Triggers`)

Specificare il tipo di trigger. È possibile utilizzare uno dei seguenti valori:

- Push (editor visivo) o PUSH (editor YAML)

Un push trigger avvia l'esecuzione di un flusso di lavoro quando una modifica viene inviata all'archivio di origine. L'esecuzione del flusso di lavoro utilizzerà i file nel ramo in cui state inviando il push (ovvero il ramo di destinazione).

- Pull request (editor visuale) o PULLREQUEST (editor YAML)

Un trigger di pull request avvia un flusso di lavoro quando una pull request viene aperta, aggiornata o chiusa nel repository di origine. L'esecuzione del flusso di lavoro utilizzerà i file nel ramo da cui state estraendo (ovvero il ramo di origine).

- Pianificazione (editor visivo) o SCHEDULE (editor YAML)

Un trigger di pianificazione avvia l'esecuzione del flusso di lavoro in base a una pianificazione definita da un'espressione cron specificata dall'utente. Verrà avviato un flusso di lavoro separato per ogni ramo del repository di origine utilizzando i file del ramo. (Per limitare i rami su cui si attiva il trigger, usa il campo `Branches` (editor visivo) o la `Branches` proprietà (editor YAML).)

Quando configuri un trigger di pianificazione, segui queste linee guida:

- Utilizza solo un trigger di pianificazione per flusso di lavoro.
- Se hai definito più flussi di lavoro nel tuo CodeCatalyst spazio, ti consigliamo di programmarne non più di 10 per avviarli contemporaneamente.
- Assicurati di configurare l'espressione cron del trigger con un tempo adeguato tra le esecuzioni. Per ulteriori informazioni, consulta [Expression](#).

Per alcuni esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

Interfaccia utente corrispondente: editor/workflow diagramma visivo/Triggers/ Tipo di trigger

Events

(Triggers/Events)

(Richiesto se il trigger è impostato su) Type PULLREQUEST

Specificate il tipo di eventi di pull request che avvieranno l'esecuzione di un flusso di lavoro. I seguenti sono i valori validi:

- Viene creata una pull request (editor visivo) o OPEN (editor YAML)

L'esecuzione del flusso di lavoro viene avviata quando viene creata una richiesta pull.

- La pull request è chiusa (editor visivo) o CLOSED (editor YAML)

L'esecuzione del flusso di lavoro viene avviata quando viene chiusa una pull request. Il comportamento dell'CLOSEDevento è complicato e può essere compreso meglio attraverso un esempio. Per ulteriori informazioni, consulta [Esempio: un trigger con pull, branch e un evento «CLOSED»](#).

- Viene effettuata una nuova revisione per pull request (editor visivo) o REVISION (editor YAML)

L'esecuzione del workflow viene avviata quando viene creata una revisione di una pull request. La prima revisione viene creata quando viene creata la pull request. Dopodiché, viene creata una nuova revisione ogni volta che qualcuno invia un nuovo commit al ramo di origine specificato nella pull request. Se includi l'REVISIONevento nel trigger della pull request, puoi omettere l'OPENevento, poiché REVISION è un superset di OPEN.

È possibile specificare più eventi nello stesso trigger di pull request.

Per alcuni esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

Interfaccia utente corrispondente: editor/workflow diagramma visivo/trigger /eventi per la richiesta pull

Branches

(Triggers/Branches)

(Facoltativo)

Specificate i rami nel vostro repository di origine monitorati dal trigger per sapere quando avviare l'esecuzione di un flusso di lavoro. È possibile utilizzare modelli regex per definire i nomi delle filiali. Ad esempio, utilizzare `main.*` abbinare tutti i rami che iniziano con `main`.

I rami da specificare sono diversi a seconda del tipo di trigger:

- Per un trigger push, specifica i rami verso cui stai eseguendo il push, ovvero i rami di destinazione. Verrà avviata un'esecuzione del flusso di lavoro per ramo corrispondente, utilizzando i file nel ramo corrispondente.

Esempi: `main.*`, `mainline`

- Per un trigger di pull request, specifica i rami verso cui stai inviando il push, ovvero i rami di destinazione. Verrà avviata un'esecuzione del flusso di lavoro per ramo corrispondente, utilizzando il file di definizione del flusso di lavoro e i file di origine nel ramo di origine (non nel ramo corrispondente).

Esempi: `main.*`, `mainline`, `v1\-.*` (corrisponde ai rami che iniziano con) `v1-`

- Per un trigger di pianificazione, specifica i rami che contengono i file che desideri vengano utilizzati dall'esecuzione pianificata. Verrà avviata un'esecuzione del flusso di lavoro per ramo corrispondente, utilizzando il file di definizione del flusso di lavoro e i file sorgente nel ramo corrispondente.

Esempi: `main.*`, `version\-1\.`

Note

Se non specifichi i rami, il trigger monitora tutti i rami nel tuo repository di origine e avvierà un flusso di lavoro eseguito utilizzando il file di definizione del flusso di lavoro e i file sorgente in:

- Il ramo verso cui stai eseguendo il push (per i trigger push). Per ulteriori informazioni, consulta [Esempio: un semplice pulsante di attivazione tramite codice](#).
- Il ramo da cui stai prelevando (per i trigger di pull request). Per ulteriori informazioni, consulta [Esempio: un semplice trigger di pull request](#).
- Tutte le filiali (per i trigger di pianificazione). Verrà avviata un'unica esecuzione del flusso di lavoro per ramo nel repository di origine. Per ulteriori informazioni, consulta [Esempio: un semplice trigger di pianificazione](#).

Per ulteriori informazioni su rami e trigger, consulta [Linee guida per l'utilizzo di trigger e filiali](#)

Per ulteriori esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

Interfaccia utente corrispondente: `visualeditor/workflow diagram/Triggers/Branches`

FilesChanged

(Triggers/FilesChanged)

(Facoltativo se il trigger Type è impostato suPUSH, oPULLREQUEST. Non supportato se il trigger Type è impostato suSCHEDULE.)

Specificate i file o le cartelle nell'archivio di origine monitorati dal trigger per sapere quando avviare l'esecuzione di un flusso di lavoro. È possibile utilizzare espressioni regolari per abbinare i nomi o i percorsi dei file.

Per alcuni esempi, consulta [Esempi: trigger nei flussi di lavoro](#).

Interfaccia utente corrispondente: editor/workflow diagramma visivo/Triggers/ File modificati

Expression

(Triggers/Expression)

(Richiesto se il trigger è impostato su) Type SCHEDULE

Specificate l'espressione cron che descrive quando desiderate che avvenga l'esecuzione del flusso di lavoro pianificato.

Le espressioni Cron CodeCatalyst utilizzano la seguente sintassi a sei campi, in cui ogni campo è separato da uno spazio:

minutes hours days-of-month month days-of-week year

Esempi di espressioni cron

Minuti	Ore	Giorni del mese	Mese	Giorni della settimana	Anno	Significato
0	0	?	*	LUN-VEN	*	Esegue un flusso di lavoro a mezzanotte (UTC+0) dal lunedì al venerdì.

Minuti	Ore	Giorni del mese	Mese	Giorni della settimana	Anno	Significato
0	2	*	*	?	*	Esegue un flusso di lavoro ogni giorno alle 2:00 (UTC +0).
15	22	*	*	?	*	Esegue un flusso di lavoro alle 22:15 (UTC +0) ogni giorno.
0/30	22-2	?	*	SAT-SUN	*	Esegue un flusso di lavoro ogni 30 minuti dal sabato alla domenica tra le 22:00 del giorno iniziale e le 2:00 del giorno successivo (UTC+0).

Minuti	Ore	Giorni del mese	Mese	Giorni della settimana	Anno	Significato
45	13	L	*	?	2023-2027	Esegue un flusso di lavoro alle 13:45 (UTC+0) dell'ultimo giorno del mese tra gli anni 2023 e 2027 inclusi.

Quando specifichi le espressioni cron in CodeCatalyst, assicurati di seguire queste linee guida:

- Specificate una singola espressione cron per trigger. SCHEDULE
- Racchiudi l'espressione cron tra virgolette doppie (") nell'editor YAML.
- Specificate l'ora in UTC (Coordinated Universal Time). Altri fusi orari non sono supportati.
- Configura almeno 30 minuti tra un'esecuzione e l'altra. Una cadenza più veloce non è supportata.
- Specificate il *days-of-week* campo *days-of-month* o, ma non entrambi. Se si specifica un valore o un asterisco (*) in uno dei campi, è necessario utilizzare un punto interrogativo (?) nell'altro. L'asterisco significa «tutti» e il punto interrogativo significa «qualsiasi».

Per altri esempi di espressioni cron e informazioni sui caratteri jolly come, e ? *L, consulta il [riferimento alle espressioni Cron nella Amazon EventBridge User Guide](#). Le espressioni Cron CodeCatalyst funzionano esattamente EventBridge allo stesso modo.

Per esempi di trigger di pianificazione, vedi. [Esempi: trigger nei flussi di lavoro](#)

Interfaccia utente corrispondente: `visualeditor/workflow diagram/Triggers/Schedule`

Azioni

Una sequenza di una o più azioni per questo flusso di lavoro. CodeCatalyst supporta diversi tipi di azioni, come azioni di compilazione e test, che offrono diversi tipi di funzionalità. Ogni tipo di azione ha:

- una `Identifier` proprietà che indica l'ID univoco e codificato dell'azione. Ad esempio, `aws/build@v1` identifica l'azione di compilazione.
- una `Configuration` sezione che contiene proprietà specifiche dell'azione.

Per ulteriori informazioni su ciascun tipo di azione, vedere [Tipi di operazione](#). L'[Tipi di operazione](#) argomento contiene collegamenti alla documentazione relativa a ciascuna azione.

Di seguito è riportato il riferimento YAML per le azioni e i gruppi di azioni nel file di definizione del flusso di lavoro.

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Actions:
  action-or-gate-name:
    Identifier: identifier
    Configuration:
      ...
  #Action groups
  action-group-name:
    Actions:
      ...
```

`action-or-gate-name`

(Actions/*action-or-gate-name*)

(Obbligatorio)

Sostituiscilo *action-name* con il nome che desideri assegnare all'azione. I nomi delle azioni devono essere univoci all'interno del flusso di lavoro e devono includere solo caratteri alfanumerici, trattini e caratteri di sottolineatura. Per ulteriori informazioni sulle regole di sintassi, vedere. [Linee guida sulla sintassi YAML](#)

Per ulteriori informazioni sulle procedure di denominazione delle azioni, comprese le restrizioni, vedere. [action-or-gate-name](#)

Interfaccia utente corrispondente: editor visivo/ **action-name** /Configuration tab/ Nome dell'azione o Nome visualizzato dell'azione

action-group-name

(Actions/**action-group-name**)

(Facoltativo)

Un gruppo di azioni contiene una o più azioni. Il raggruppamento delle azioni in gruppi di azioni consente di mantenere organizzato il flusso di lavoro e consente inoltre di configurare le dipendenze tra diversi gruppi.

action-group-name Sostituiscilo con un nome che desideri assegnare al gruppo di azioni. I nomi dei gruppi di azioni devono essere univoci all'interno del flusso di lavoro e devono includere solo caratteri alfanumerici, trattini e caratteri di sottolineatura. Per ulteriori informazioni sulle regole di sintassi, vedere. [Linee guida sulla sintassi YAML](#)

Per ulteriori informazioni sui gruppi di azioni, vedere [Raggruppamento delle azioni in gruppi di azione](#).

Interfaccia utente corrispondente: nessuna

Tieni traccia e organizza il lavoro con problemi in CodeCatalyst

In CodeCatalyst, puoi monitorare funzionalità, bug e qualsiasi altro lavoro coinvolto nel tuo progetto. Ogni opera è conservata in un registro distinto chiamato issue. È possibile suddividere un problema in obiettivi più piccoli aggiungendovi un elenco di attività. Ogni problema può avere una descrizione, un assegnatario, uno stato e altre proprietà che puoi cercare, raggruppare e filtrare. Puoi visualizzare i problemi utilizzando le visualizzazioni predefinite oppure puoi creare visualizzazioni personalizzate con filtri, ordinamenti o raggruppamenti personalizzati. Per ulteriori informazioni sui concetti relativi ai problemi, vedere [Concetti relativi ai problemi](#). Per informazioni su come creare il primo numero, consulta [Creazione di un problema in CodeCatalyst](#).

Ecco un possibile flusso di lavoro per un team che utilizza i problemi:

Jorge Souza è uno sviluppatore che lavora a un progetto. Lui e i suoi colleghi membri del progetto Li Juan, Mateo Jackson e Wang Xiulan collaborano per determinare quale lavoro deve essere svolto. Ogni giorno, lui e i suoi colleghi sviluppatori tengono una riunione di sincronizzazione, guidata da Wang Xiulan. Visualizzano la lavagna accedendo alla visualizzazione della bacheca da parte di uno dei loro team. Creando visualizzazioni, utenti e team possono salvare filtri, raggruppamenti e ordinamento dei problemi per visualizzare facilmente i problemi che soddisfano i criteri specificati. La loro visualizzazione contiene i problemi raggruppati per Assegnatario e ordinati per priorità per mostrare i problemi più importanti e lo stato dei problemi per ogni sviluppatore. Poiché a Jorge vengono assegnati compiti da completare, pianifica il suo lavoro creando un problema per ogni attività. Durante la creazione dei problemi, Jorge può scegliere lo stato, la priorità e lo sforzo di stima del lavoro appropriati. Per problemi più grandi, Jorge aggiunge delle attività al problema, per suddividere il lavoro in obiettivi più piccoli. Jorge crea i suoi problemi con una bozza di status, ad esempio un arretrato, poiché non ha intenzione di iniziarli immediatamente. I problemi con stato di bozza vengono visualizzati nella visualizzazione Bozze, dove è necessario pianificarli e assegnare loro la priorità. Quando Jorge è pronto per iniziare il lavoro, sposta il problema corrispondente nella bacheca aggiornandone lo stato a uno stato in un'altra categoria (Non iniziato, Iniziato o Completato). Man mano che ogni attività viene completata, il team può filtrare per titolo, stato, assegnatario, etichetta, priorità e stima per trovare un problema specifico o problemi simili che corrispondono al parametro specificato. Utilizzando la lavagna, Jorge e il suo team possono vedere il numero di attività completate per ogni problema e monitorare i day-to-day progressi trascinando ogni problema da uno stato all'altro fino al completamento dell'attività. Man mano che il progetto avanza, i problemi terminati si accumulano nello stato Completato. Wang Xiulan decide di rimuoverli dalla visualizzazione

archiviandoli utilizzando il pulsante di archiviazione rapida, in modo che gli sviluppatori possano concentrarsi sui problemi relativi ai lavori attuali e futuri.

Quando pianificano il loro lavoro, gli sviluppatori che lavorano al progetto scelgono Ordina per e Raggruppa per per trovare i problemi che desiderano spostare dal backlog alla lavagna. Potrebbero scegliere di aggiungere problemi alla bacheca in base alle richieste dei clienti con la massima priorità, quindi raggruppano la bacheca in base all'etichetta Customer request e ordinano per Priorità. Potrebbero anche ordinare in base a stime per assicurarsi che si stiano occupando del volume di lavoro che riescono a realizzare. Il project manager, Saanvi Sarkar, esamina e riordina regolarmente l'arretrato per garantire che la priorità rifletta accuratamente l'importanza di ogni questione per il successo del progetto.

Argomenti

- [Concetti relativi ai problemi](#)
- [Monitoraggio del lavoro con problemi](#)
- [Organizzazione del lavoro con arretrati, etichette e bacheche](#)
- [Quote per le emissioni in CodeCatalyst](#)

Concetti relativi ai problemi

La creazione di un problema è un modo rapido ed efficiente per tenere traccia del lavoro svolto all'interno di un progetto. Puoi utilizzare i problemi per discutere del lavoro nelle riunioni quotidiane di sincronizzazione, dare priorità al lavoro e altro ancora.

Questa pagina include un elenco di concetti che ti aiuteranno a utilizzare efficacemente i problemi in CodeCatalyst

Problemi attivi

I problemi attivi sono tutti i problemi che non sono in stato Bozza o archiviati. In altre parole, i problemi attivi sono problemi con uno stato in una delle seguenti categorie di stato: Non avviato, Avviato e Completato. Per ulteriori informazioni sugli stati e le categorie di stato, vedere [Stato e categorie di stato](#).

È possibile visualizzare tutti i problemi attivi del progetto dalla visualizzazione predefinita Problemi attivi.

Problemi archiviati

Un problema archiviato è un problema che non è più rilevante per il progetto. Ad esempio, puoi [archiviare un problema](#) se è stato completato e non hai più bisogno di vederlo nella colonna Fine o se è stato creato per errore. I problemi archiviati possono essere recuperati se necessario.

Assegnatario

L'assegnatario è la persona a cui è assegnato il problema. Se la persona non appare nell'elenco quando la cerchi, significa che non è stata aggiunta al tuo progetto. Per aggiungerli, vedi [Invitare un utente a un progetto](#). Per abilitare più assegnatari a un problema, vedi [Abilitazione o disabilitazione di più assegnatari](#). I problemi con più assegnatari appariranno sulla tua bacheca con avatar di colori diversi, ognuno dei quali rappresenta uno degli assegnatari.

Campi personalizzati

I campi personalizzati consentono di personalizzare diversi attributi di un problema in base alle proprie esigenze di monitoraggio e gestione dei problemi all'interno di un progetto. Ad esempio, puoi aggiungere un campo per la roadmap, una data di scadenza specifica o un campo richiedente.

Stima

Nello sviluppo agile, la stima è nota come punti della storia. È possibile utilizzare la stima di un problema per rappresentare la quantità di lavoro richiesta, oltre all'ambiguità e alla complessità del problema. Prendi in considerazione l'utilizzo di stime più elevate per i problemi con maggiori rischi, difficoltà e incognite.

Per ulteriori informazioni sui tipi di stima e su come configurarli, consulta [Configurazione della stima dello sforzo da risolvere](#).

Problema

Un problema è un record che tiene traccia del lavoro relativo al progetto. Puoi creare un problema per una funzionalità, un'attività, un bug o qualsiasi altro documento relativo al tuo progetto. Se utilizzi lo sviluppo agile, un problema può anche descrivere un'epopea o una storia utente.

Etichetta

L'etichetta viene utilizzata per raggruppare, ordinare e filtrare i problemi. Puoi inserire un nuovo nome per l'etichetta o scegliere una delle etichette dall'elenco popolato. Questo elenco comprende

le etichette utilizzate di recente nel progetto. Un problema può avere più etichette e un'etichetta può essere rimossa da un numero. Per personalizzare le etichette, consulta [Categorizzazione del lavoro con le etichette](#).

Priorità

La priorità si riferisce al livello di importanza del problema. Sono disponibili quattro opzioni: Bassa, Media, Alta e Nessuna priorità.

Stato e categorie di stato

Lo stato è lo stato attuale del problema e viene utilizzato per verificare rapidamente lo stato di avanzamento di un problema durante il suo ciclo di vita, dall'inizio al completamento. Tutti i problemi devono avere uno stato e ogni stato appartiene a una categoria di stato. Le categorie di stato vengono utilizzate per organizzare gli stati e compilare le visualizzazioni predefinite dei problemi.

Esistono cinque stati predefiniti e quattro categorie di stato. CodeCatalyst È possibile creare altri stati, ma non è possibile creare altre categorie di stato. L'elenco seguente contiene gli stati predefiniti e le relative categorie di stato tra parentesi: Backlog (Bozza), Da fare (Non avviato), In corso (Avviato), In revisione (Avviato) e Fatto (Completato).

Per ulteriori informazioni sull'utilizzo degli stati, vedere. [Il monitoraggio funziona con stati personalizzati](#)

Processi

È possibile aggiungere attività ai problemi per suddividerli ulteriormente e organizzarne il lavoro. Puoi aggiungere attività a un problema al momento della creazione o aggiungere attività a un problema esistente. Quando visualizzi un problema, puoi riordinare, rimuovere o contrassegnare le attività come completate.

Visualizzazioni

I problemi del CodeCatalyst progetto vengono visualizzati nelle visualizzazioni. Le visualizzazioni possono essere viste a griglia che mostrano i problemi in formato elenco o visualizzazioni di bacheche che mostrano i problemi sotto forma di riquadri in colonne organizzate in base allo stato del problema. Sono disponibili quattro visualizzazioni predefinite ed è possibile [creare visualizzazioni personalizzate con raggruppamento, filtro e ordinamento](#) personalizzati. L'elenco seguente contiene dettagli sulle quattro visualizzazioni predefinite.

- La vista Bozze è una visualizzazione a griglia che mostra i problemi su cui non si sta attualmente lavorando. Qualsiasi problema creato con uno stato nella categoria Stato Bozza viene visualizzato in questa visualizzazione. Questa visualizzazione può essere utilizzata dai team per vedere quali problemi sono ancora in fase di definizione o sono in attesa di essere assegnati e risolti.
- La visualizzazione Problemi attivi è una panoramica di tutti i problemi su cui si sta attualmente lavorando. Qualsiasi problema con uno stato nelle categorie Non iniziato, Iniziato o Completato verrà visualizzato in questa visualizzazione.
- La visualizzazione Tutti i problemi è una visualizzazione a griglia che mostra tutti i problemi del progetto, sia le bozze che i problemi attivi.
- La visualizzazione Archiviata mostra tutti i problemi archiviati.

Monitoraggio del lavoro con problemi

Puoi pianificare e tenere traccia del tuo lavoro su un progetto utilizzando i problemi. Ogni numero è un'opera conservata in un registro distinto. I problemi possono essere suddivisi in attività per organizzare e monitorare ulteriormente il lavoro svolto. È inoltre possibile creare collegamenti tra i problemi per tenere traccia del lavoro correlato, aggiungere etichette per organizzare e classificare il lavoro, raggruppare i problemi, assegnare priorità al lavoro e indicare se il lavoro è bloccato.

Quando sei pronto a lavorare su un problema o su una serie di problemi, puoi stimare il lavoro, assegnarlo agli utenti e aggiungere commenti per aiutare gli altri a comprendere il lavoro e i suoi progressi. Puoi anche esportare i problemi per contribuire a trasferire le informazioni in essi contenute in altri formati.

Creazione di un problema in CodeCatalyst

I team di sviluppo creano problemi per monitorare e gestire il proprio lavoro. Puoi creare problemi all'interno di un progetto in base alle tue esigenze. Ad esempio, potresti creare un problema per tenere traccia dell'aggiornamento di una variabile nel codice. Puoi assegnare problemi ad altri utenti del progetto, utilizzare etichette per monitorare il tuo lavoro e altro ancora.

Segui queste istruzioni per creare un problema in CodeCatalyst.

Per creare un problema

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al progetto in cui desideri creare un problema.

3. Nella home page del progetto, scegli Crea problema. In alternativa, nel riquadro di navigazione, scegli Problemi.
4. Scegli Crea problema.

 Note

Puoi anche aggiungere problemi in linea quando usi una visualizzazione a griglia.

5. Inserisci un titolo per il problema.
6. (Facoltativo) Inserisci una descrizione. Puoi usare Markdown per aggiungere la formattazione.
7. (Facoltativo) Scegli uno stato, una priorità e una stima per il problema.

 Note

Se l'impostazione di stima del progetto è impostata su Nascondi stime, non sarà presente un campo Stima.

8. (Facoltativo) Aggiungi attività al problema. Le attività possono essere utilizzate per suddividere il lavoro di un problema in obiettivi più piccoli. Per aggiungere un'attività, scegli + Aggiungi attività. Quindi, inserisci il nome dell'attività nel campo di testo e premi invio. Dopo aver aggiunto le attività, puoi contrassegnarle come complete selezionando la casella di controllo o riordinarle scegliendo e trascinando l'attività dal lato sinistro della casella di controllo.
9. (Facoltativo) Aggiungi un'etichetta esistente o crea una nuova etichetta e aggiungila scegliendo + Aggiungi etichetta.
 - a. Per aggiungere un'etichetta esistente, scegli l'etichetta dall'elenco. Puoi inserire un termine di ricerca nel campo per cercare tutte le etichette che contengono quel termine nel progetto.
 - b. Per creare una nuova etichetta e aggiungerla, inserisci il nome dell'etichetta che desideri creare nel campo di ricerca e premi invio.
10. (Facoltativo) Aggiungi un assegnatario selezionando + Aggiungi un assegnatario. Puoi aggiungerti rapidamente come assegnatario selezionando + Aggiungimi.

 Tip

Puoi scegliere di assegnare un problema ad Amazon Q per fare in modo che Amazon Q provi a risolverlo. Per ulteriori informazioni, consulta [Tutorial: Utilizzo delle funzionalità](#)

[di intelligenza artificiale CodeCatalyst generativa per velocizzare il lavoro di sviluppo.](#)

Questa funzionalità è disponibile solo nella regione Stati Uniti occidentali (Oregon).

Questa funzionalità richiede che le funzionalità di intelligenza artificiale generativa siano abilitate per lo spazio. Per ulteriori informazioni, consulta [Gestione delle funzionalità di intelligenza artificiale generativa](#).

11. (Facoltativo) Aggiungi un campo personalizzato esistente o crea un nuovo campo personalizzato. I problemi possono avere più campi personalizzati.
 - a. Per aggiungere un campo personalizzato esistente, scegli il campo personalizzato dall'elenco. Puoi inserire un termine di ricerca nel campo per cercare tutti i campi personalizzati contenenti quel termine nel progetto.
 - b. Per creare un nuovo campo personalizzato e aggiungerlo, inserisci il nome del campo personalizzato che desideri creare nel campo di ricerca e premi invio. Quindi scegli il tipo di campo personalizzato che desideri creare e imposta un valore.
12. Scegli Crea problema. Nell'angolo in basso a destra viene visualizzata una notifica: se il problema è stato creato correttamente, viene visualizzato un messaggio di conferma che indica che il problema è stato creato correttamente. Se il problema non è stato creato correttamente, viene visualizzato un messaggio di errore con il motivo dell'errore. Puoi quindi scegliere Riprova per modificare e riprova a creare il problema oppure scegliere Ignora per eliminare il problema. Entrambe le opzioni annulleranno la notifica.

Note

Non è possibile collegare una pull request a un problema al momento della creazione. Tuttavia, puoi [modificarla](#) dopo averla creata per aggiungere link alle richieste pull.

Procedure consigliate per la creazione e l'utilizzo di problemi assegnati ad Amazon Q

Quando crei problemi, a volte alcuni persistono. Le cause di ciò possono essere complesse e variabili. A volte è perché non è chiaro chi debba lavorarci. Altre volte il problema richiede ricerche o competenze su una parte particolare del codice base e i migliori candidati per il lavoro sono impegnati con altre questioni. Spesso ci sono altri lavori urgenti da sbrigare prima. Alcune o tutte queste cause possono causare problemi irrisolti. CodeCatalyst include l'integrazione con un assistente AI generativo chiamato Amazon Q in grado di analizzare un problema in base al titolo e alla descrizione. Se assegni il problema ad Amazon Q, quest'ultima tenterà di creare una bozza di soluzione da

valutare. Questo può aiutare te e il tuo team a concentrarti e ottimizzare il lavoro su questioni che richiedono la tua attenzione, mentre Amazon Q lavora a una soluzione per i problemi che non hai risorse per risolvere immediatamente.

Note

Note

Realizzato da Amazon Bedrock: AWS implementa il rilevamento [automatico degli abusi](#). Poiché le funzionalità *Scrivi una descrizione per me*, *Crea un riepilogo dei commenti*, *Attività consigliate*, *Usa Amazon Q per creare o aggiungere funzionalità a un progetto* e *Assegna i problemi ad Amazon Q con Amazon Q Developer Agent* per lo sviluppo software sono supportate da Amazon Bedrock, gli utenti possono utilizzare al meglio i controlli implementati in Amazon Bedrock per rafforzare la protezione, la sicurezza e l'uso responsabile dell'intelligenza artificiale (IA).

Amazon Q offre le migliori prestazioni su problemi semplici e diretti. Per ottenere i migliori risultati, usa un linguaggio semplice per spiegare chiaramente cosa vuoi fare. Di seguito sono riportate alcune best practice per aiutarti a creare problemi ottimizzati per Amazon Q su cui lavorare.

Important

Le funzionalità di intelligenza artificiale generativa sono disponibili solo nella regione Stati Uniti occidentali (Oregon).

- Mantieni la semplicità. Amazon Q fa meglio con semplici modifiche e correzioni al codice, che possono essere spiegate nel titolo e nella descrizione del problema. Non assegnare problemi con titoli vaghi o descrizioni eccessivamente esagerate o contraddittorie.
- Sii specifico. Più informazioni puoi fornire sulle modifiche esatte necessarie per risolvere il problema, più è probabile che Amazon Q sia in grado di creare una soluzione che risolva il problema. Se possibile, includi dettagli specifici come il nome che APIs desideri modificare, i metodi che desideri aggiornare, i test che richiedono modifiche e qualsiasi altro dettaglio che ti viene in mente.

- Assicurati di avere tutti i dettagli inclusi nel titolo e nella descrizione del problema prima di assegnarlo ad Amazon Q. Non puoi modificare il titolo o la descrizione di un problema dopo averlo assegnato ad Amazon Q, quindi assicurati di disporre di tutte le informazioni necessarie per un problema prima di assegnarlo ad Amazon Q.
- Assegna solo i problemi che richiedono modifiche al codice in un unico repository di origine. Amazon Q può funzionare solo sul codice in un unico repository di origine in CodeCatalyst. I repository collegati non sono supportati. Assicurati che il problema richieda solo modifiche in un unico repository di origine prima di assegnarlo ad Amazon Q.
- Utilizza l'impostazione predefinita suggerita da Amazon Q per l'approvazione di ogni passaggio. Per impostazione predefinita, Amazon Q richiederà la tua approvazione per ogni passaggio necessario. Ciò ti consente di interagire con Amazon Q nei commenti non solo sul problema, ma anche su qualsiasi pull request creata. Ciò offre un'esperienza più interattiva con Amazon Q che ti aiuta a modificare il suo approccio e perfezionare il codice che crea per risolvere il problema.

Note

Amazon Q non risponde ai singoli commenti nei problemi o nelle pull request, ma li esaminerà quando gli verrà chiesto di riconsiderare il suo approccio o di creare una revisione.

- Esamina sempre attentamente l'approccio suggerito da Amazon Q. Una volta approvato l'approccio, Amazon Q inizierà a lavorare sulla generazione di codice basato su tale approccio. Assicurati che l'approccio sembri corretto e includa tutti i dettagli previsti prima di dire ad Amazon Q di procedere.
- Assicurati di consentire ad Amazon Q di lavorare sui flussi di lavoro solo se non disponi di flussi di lavoro esistenti che potrebbero distribuirli prima che vengano esaminati. Il tuo progetto potrebbe avere flussi di lavoro configurati per avviare esecuzioni in base a eventi di pull request. In tal caso, qualsiasi richiesta pull creata da Amazon Q che includa la creazione o l'aggiornamento del flusso di lavoro YAML potrebbe avviare un'esecuzione dei flussi di lavoro inclusi nella richiesta pull. Come best practice, non scegliere di consentire ad Amazon Q di lavorare su file di workflow a meno che tu non sia sicuro che nel tuo progetto non ci siano flussi di lavoro che eseguano automaticamente tali flussi di lavoro prima di rivedere e approvare la pull request creata.

Per ulteriori informazioni, consulta [Gestione delle funzionalità di Tutorial: Utilizzo delle funzionalità di intelligenza artificiale CodeCatalyst generativa per velocizzare il lavoro di sviluppo intelligenza artificiale generativa](#).

Stima di un problema

Nello sviluppo agile, la stima è nota come story points. È possibile utilizzare la stima di un problema per rappresentare la quantità di lavoro richiesta, oltre all'ambiguità e alla complessità del problema. Prendi in considerazione l'utilizzo di stime più elevate per i problemi con maggiori rischi, difficoltà e incognite.

Prima di iniziare a stimare i problemi, è necessario innanzitutto scegliere il tipo di stime da utilizzare per il progetto. Per impostazione predefinita, sono disponibili due tipi tra cui scegliere. Per utilizzare in modo efficace la taglia delle magliette o il sequenziamento di Fibonacci, la tua squadra deve allinearsi su ciò che rappresenta ciascuna taglia. Decidete insieme cosa rappresenta per voi ogni stima, quindi iniziate ad applicarla a ciascun numero. Valuta la possibilità di riveder

Segui questi passaggi per configurare l'impostazione per le stime degli sforzi relativi ai problemi in CodeCatalyst.

Per configurare la stima dello sforzo per i problemi

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. Nella sezione Stima, nella sezione Impostazioni di base, scegli come verranno visualizzati i valori di stima. I tipi di stime disponibili sono il dimensionamento delle magliette, il sequenziamento di Fibonacci o le stime Hide. Se l'impostazione di stima del progetto è impostata su Nascondi stime, non ci sarà un campo Stima nei numeri del progetto.

Quando il tipo di stima viene aggiornato, nessun dato andrà perso e il valore di stima di tutti i problemi verrà convertito automaticamente. La mappatura delle conversioni è mostrata nella tabella seguente.

Taglia della maglietta	Sequenza di Fibonacci
XS	1
XS	2
S	3
M	5

Taglia della maglietta	Sequenza di Fibonacci
L	8
XL	13

Per aggiungere o modificare una stima per un problema, puoi [modificare il problema](#).

Redazione e collaborazione su problemi in CodeCatalyst

Indice

- [Modificare un problema](#)
- [Utilizzare gli allegati](#)
 - [Visualizzazione e gestione degli allegati](#)
- [Gestione delle attività relative ai problemi](#)
- [Collegare un problema a un altro](#)
- [Contrassegnare un problema come bloccato o sbloccato](#)
- [Aggiungere, modificare o eliminare commenti](#)
 - [Usare le menzioni in un commento](#)

Modificare un problema

Segui questi passaggi per modificare il titolo, la descrizione, lo stato, l'assegnatario, la priorità, la stima o le etichette di un problema.

Per modificare un problema

1. Scegli il problema che desideri modificare per visualizzarne i dettagli. Per assistenza su come individuare il problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. Per modificare il titolo del problema, scegli il titolo, inserisci un nuovo titolo e premi invio.
3. Per modificare la descrizione, scegli la descrizione, inserisci una nuova descrizione e premi invio. Puoi usare Markdown per aggiungere la formattazione.
4. In Attività, puoi visualizzare e gestire le attività relative al problema. Se non ci sono attività, puoi fare in modo che Amazon Q analizzi il problema e consigli attività che possono suddividere il

lavoro relativo al problema in elementi separati, ognuno dei quali può essere assegnato a un utente. Per ulteriori informazioni, consulta [Gestione delle attività relative ai problemi](#).

5. Per modificare lo stato, la stima o la priorità, scegli un'opzione dai rispettivi menu a discesa.
6. In Etichette, puoi aggiungere un'etichetta esistente, creare una nuova etichetta o rimuovere un'etichetta.
 - a. Per aggiungere un'etichetta esistente, scegli + Aggiungi etichetta e scegli l'etichetta dall'elenco. Puoi inserire un termine di ricerca nel campo per cercare tutte le etichette che contengono quel termine nel progetto.
 - b. Per creare una nuova etichetta e aggiungerla, scegli + Aggiungi etichetta, inserisci il nome dell'etichetta che desideri creare nel campo di ricerca e premi invio.
 - c. Per rimuovere un'etichetta, scegli l'icona X accanto all'etichetta che desideri rimuovere. Se rimuovi un'etichetta da tutti i numeri, l'etichetta verrà visualizzata nella sezione Etichette non utilizzate della sezione Etichette delle impostazioni del numero. Le etichette non utilizzate vengono visualizzate alla fine dell'elenco delle etichette quando si utilizzano filtri o si aggiungono etichette a un numero. Puoi trovare una panoramica di tutte le etichette (usate e non utilizzate) e dei problemi che le presentano nelle impostazioni del problema.
7. Per assegnare un problema, scegli + Aggiungi un assegnatario nella sezione Assegnatario, quindi cerca e scegli l'assegnatario dall'elenco. Puoi scegliere + Aggiungi me per aggiungerti rapidamente come assegnatario.
8. In Allegati, puoi aggiungere, scaricare o rimuovere allegati. Per ulteriori informazioni, consulta [Utilizzare gli allegati](#).
9. Per collegare una pull request, scegli Collega pull request, quindi scegli una pull request dall'elenco o inserisci il relativo URL o ID. Per scollegare una pull request, scegli l'icona di scollegamento.

 Tip

Dopo aver aggiunto un link a una pull request a un problema, puoi accedervi rapidamente selezionandone l'ID nell'elenco delle pull request collegate. Puoi utilizzare l'URL di una pull request per collegare le pull request che si trovano in progetti diversi dalla issue board, ma solo gli utenti che sono membri di quel progetto potranno visualizzare o navigare verso quella pull request.

10. (Facoltativo) Aggiungi e imposta un campo personalizzato esistente, crea un nuovo campo personalizzato o rimuovi un campo personalizzato. I problemi possono avere più campi personalizzati.
 - a. Per aggiungere un campo personalizzato esistente, scegli il campo personalizzato dall'elenco. Puoi inserire un termine di ricerca nel campo per cercare tutti i campi personalizzati contenenti quel termine nel progetto.
 - b. Per creare un nuovo campo personalizzato e aggiungerlo, inserisci il nome del campo personalizzato che desideri creare nel campo di ricerca e premi invio. Quindi scegli il tipo di campo personalizzato che desideri creare e imposta un valore.
 - c. Per rimuovere un campo personalizzato, scegli l'icona X accanto al campo personalizzato che desideri rimuovere. Se rimuovi un campo personalizzato da tutti i numeri, il campo personalizzato verrà eliminato e non lo vedrai più durante il filtraggio.

Utilizzare gli allegati

Puoi aggiungere allegati ai problemi in per CodeCatalyst rendere facilmente accessibili i file correlati. Utilizza la procedura seguente per gestire gli allegati relativi a un problema.

La dimensione degli allegati aggiunti ai problemi viene conteggiata ai fini delle quote di archiviazione del tuo spazio. Per informazioni sulla visualizzazione e la gestione degli allegati del progetto, consulta [Visualizzazione e gestione degli allegati](#)

Important

Gli allegati ai problemi non vengono scansati o analizzati da Amazon. CodeCatalyst
Qualsiasi utente può aggiungere un allegato a un problema che potrebbe contenere codice o contenuti dannosi. Assicurati che gli utenti conoscano le migliori pratiche per la gestione degli allegati e la protezione da codice, contenuti o virus dannosi.

Per aggiungere, scaricare o rimuovere allegati

1. Scegli il problema per il quale desideri gestire gli allegati. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. Per aggiungere un allegato, scegli Carica file. Vai al file nell'esploratore di file del tuo sistema operativo e selezionalo. Scegli Apri per aggiungerlo come allegato. Per informazioni sulla

quota, ad esempio la dimensione massima degli allegati, consulta [Quote per le emissioni in CodeCatalyst](#).

Tieni presente le seguenti restrizioni relative ai nomi dei file e ai tipi di contenuto degli allegati:

- I seguenti caratteri non sono consentiti nei nomi di file:
 - Caratteri di controllo: 0x00–0x1f e 0x80–0x9f
 - Caratteri riservati: /?,<,>,\,;,*,|, e "
 - Nomi di file riservati Unix: e . . .
 - Periodi e spazi finali
 - Nomi di file riservati per Windows: CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9
- Il tipo di contenuto dell'allegato deve rispettare il seguente schema di tipo di supporto:

```
media-type = type "/" [tree "."] subtype ["+" suffix]* [";" parameter];
```

Ad esempio, text/html; charset=UTF-8.

3. Per scaricare un allegato, scegliete il menu con i puntini di sospensione accanto all'allegato che desiderate scaricare e scegliete Scarica.
4. Per copiare l'URL di un allegato, scegli il menu con i puntini di sospensione accanto all'allegato di cui desideri copiare l'URL e scegli Copia URL.
5. Per rimuovere un allegato, scegli il menu con i puntini di sospensione accanto all'allegato che desideri rimuovere e scegli Elimina.

Visualizzazione e gestione degli allegati

Puoi visualizzare una tabella con tutti gli allegati aggiunti ai problemi del tuo progetto nelle impostazioni dei problemi. Questa tabella include i dettagli di ogni allegato, incluse informazioni come il tipo di contenuto, la data di aggiunta, il problema a cui è stato aggiunto, il relativo stato e la dimensione del file.

Questa tabella può essere utilizzata per identificare facilmente gli allegati di grandi dimensioni relativi a problemi completati o archiviati e rimuoverli per liberare spazio di archiviazione.

Important

Gli allegati ai problemi non vengono scansionati o analizzati da Amazon. CodeCatalyst
Qualsiasi utente può aggiungere un allegato a un problema che potrebbe contenere codice o contenuti dannosi. Assicurati che gli utenti conoscano le migliori pratiche per la gestione degli allegati e la protezione da codice, contenuti o virus dannosi.

Per visualizzare e gestire tutti gli allegati dei problemi in un progetto

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli l'icona con i puntini di sospensione e scegli Impostazioni.
3. Scegli la scheda Allegati.

Gestione delle attività relative ai problemi

È possibile aggiungere attività ai problemi per suddividerli, organizzarli e monitorarne ulteriormente il funzionamento. Puoi creare attività da solo oppure puoi utilizzare Amazon Q per consigliare attività in base all'analisi del problema e alla sua complessità.

Amazon Q Developer è un assistente conversazionale generativo basato sull'intelligenza artificiale che può aiutarti a comprendere, creare, estendere e utilizzare le applicazioni. AWS Per accelerare la tua crescita AWS, il modello alla base di Amazon Q è arricchito con AWS contenuti di alta qualità per produrre risposte più complete, utilizzabili e referenziate. Per ulteriori informazioni, consulta [Cos'è Amazon Q Developer?](#) nella Guida per l'utente di Amazon Q Developer.

Per gestire le attività relative a un problema

1. Scegli il problema per il quale desideri gestire le attività. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. In Attività, puoi visualizzare e gestire le attività relative al problema.
 1. Per aggiungere un'attività, inserisci il nome dell'attività nel campo di testo e premi invio.
 2. Se non ci sono attività per il problema, puoi scegliere di fare in modo che Amazon Q analizzi il problema e crei attività in base al titolo del problema, alla descrizione e alla relativa analisi della complessità del problema e del codice del repository, scegliendo Consiglia attività. Dovrai specificare l'archivio di origine che contiene il codice del problema. Scegli Inizia a consigliare le attività per iniziare l'analisi dei consigli sulle attività. Quella finestra di

dialogo si chiuderà. Una volta completata la raccomandazione, scegli Visualizza le attività consigliate per rivedere le attività e intraprendere le azioni necessarie, ad esempio eliminare o aggiungere attività all'elenco o riordinare le attività consigliate, prima di scegliere Crea attività.

Dopo aver creato le attività automaticamente, puoi assegnarle agli utenti e utilizzarle nello stesso modo in cui lavori con le attività create manualmente.

3. Per contrassegnare un'attività come completata, scegli la casella di controllo dell'attività.
4. Per visualizzare o aggiornare i dettagli di un'attività, selezionala dall'elenco.
5. Per riordinare le attività, scegli e trascina l'attività dal lato sinistro della casella di controllo.
6. Per rimuovere un'attività, scegli il menu con i puntini di sospensione dell'attività e scegli Rimuovi.

Collegare un problema a un altro

Se un problema è correlato al lavoro su un altro numero, puoi collegarlo. Questo è un modo rapido per aiutarti a comprendere e tenere traccia delle dipendenze tra gli elementi di lavoro.

Quando si collega un problema, è possibile scegliere tra quattro stati:

- Correlato a: utilizza questo stato per indicare un lavoro correlato al problema collegato.
- Bloccato da: utilizza questo stato per indicare che il problema è bloccato dal problema collegato.
- Blocco a: utilizza questo stato per indicare che il problema sta bloccando l'avanzamento del problema collegato.
- Duplica: utilizza questo stato per indicare che il problema duplica il lavoro acquisito in un altro numero.

È possibile modificare lo stato di un collegamento dopo averlo creato. I collegamenti e il relativo stato dei collegamenti vengono visualizzati in Problemi collegati sia nel problema in cui è stato creato il collegamento sia nel problema collegato. Puoi anche rimuovere i link dopo averli creati.

Per collegare un problema a un altro problema o attività

1. Apri il problema che desideri collegare a un altro problema. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. Scegli Problemi relativi ai collegamenti.
3. In Segna come, scegli lo stato del link.

In Numero numero, inserisci il numero del problema o selezionalo dall'elenco a discesa.

4. Per aggiungere un altro link, scegli Aggiungi problema collegato.
5. Al termine, scegli Aggiorna per aggiornare il problema e tutti i problemi collegati alla relazione collegata.

Contrassegnare un problema come bloccato o sbloccato

Se qualcosa ti impedisce di risolvere un problema, potresti volerlo contrassegnare come bloccato. Ad esempio, il problema potrebbe essere bloccato se si basa su una modifica a un'altra parte della base di codice che non è stata ancora unita.

Quando contrassegni un problema come bloccato, CodeCatalyst aggiunge un'etichetta rossa Bloccato al problema, rendendolo ben visibile nel backlog o nell'archivio o nella bacheca.

Puoi sbloccare il problema quando le circostanze esterne vengono risolte.

Per contrassegnare un problema come bloccato

1. Apri il problema che desideri contrassegnare come bloccato. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. Scegli Azioni, quindi scegli Contrassegna come bloccato.

Per sbloccare un problema

1. Apri il problema che desideri sbloccare. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. Scegli Azioni, quindi scegli Contrassegna come sbloccato.

Aggiungere, modificare o eliminare commenti

Puoi lasciare un commento su un problema. Nei commenti puoi taggare altri membri dello spazio, altri progetti nello spazio, problemi correlati e codice.

Per aggiungere un commento a un problema

1. Vai al tuo progetto.
2. Nella barra di navigazione scegli Problemi.

3. Scegli il problema a cui desideri aggiungere il commento. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
4. Inserisci il commento nel campo Commenti. Puoi usare Markdown per aggiungere la formattazione.
5. Scegli Invia.

Per modificare un commento

Puoi modificare i commenti che fai sui problemi. Puoi modificare solo i commenti di cui sei l'autore.

1. Vai al tuo progetto.
2. Nella barra di navigazione scegli Problemi.
3. Scegli il problema in cui desideri modificare un commento. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
4. Per modificare un commento, trova il commento che desideri modificare.

Tip

Puoi ordinare prima i commenti in base al più vecchio o al più recente. I commenti vengono caricati 10 alla volta.

5. Scegli l'icona con i puntini di sospensione, quindi scegli Modifica.
6. Modifica il commento. Puoi usare Markdown per aggiungere la formattazione.
7. Scegli Save (Salva). Il commento è ora aggiornato.

Per eliminare un commento

Puoi eliminare i commenti che fai sui problemi. Puoi eliminare solo i commenti di cui sei l'autore. Quando un commento viene eliminato, viene visualizzato il nome utente, ma con le parole Questo commento è stato eliminato al posto del testo originale del commento.

1. Vai al tuo progetto.
2. Nella barra di navigazione scegli Problemi.
3. Scegli il problema in cui desideri eliminare un commento. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
4. Scegli l'icona con i puntini di sospensione, scegli Elimina, quindi scegli Conferma.

Usare le menzioni in un commento

Puoi menzionare i membri dello spazio, altri progetti nello spazio, i problemi correlati e il codice nei commenti. In questo modo viene creato un collegamento rapido all'utente o alla risorsa menzionata.

A @mention in un commento

1. Vai al tuo progetto.
2. Nella barra di navigazione scegli Problemi.
3. Scegli il problema che desideri modificare per visualizzarne i dettagli. Per assistenza su come individuare il problema, consulta [Individuazione e visualizzazione dei problemi](#).
4. Scegli la casella di testo Aggiungi un commento.
5. Digita `@user_name` per menzionare un altro utente.
6. Digita `@project_name` per menzionare un progetto.
7. Digita `@issue_name` o `@issue_number` menziona un altro problema.
8. Digita `@file_name` per menzionare file o codice specifici in un repository di sorgenti.

Note

Un elenco dei 5 elementi principali (utenti, repository di sorgenti, progetti, ecc.) contenenti i termini che corrispondono ai tuoi @mention verrà compilato durante la digitazione.

9. Scegli l'elemento desiderato che desideri menzionare. Il percorso che mostra dove si trova l'elemento verrà inserito nella casella di testo del commento.
10. Completa il commento e scegli Invia.

Individuazione e visualizzazione dei problemi

Le sezioni seguenti descrivono come cercare e visualizzare efficacemente i problemi all'interno di un CodeCatalyst progetto.

Ricerca di un problema

Puoi trovare un problema cercando parametri specifici. Per ulteriori informazioni sul perfezionamento della ricerca, consulta [Cerca codice, problemi, progetti e utenti in CodeCatalyst](#).

Per cercare un problema

1. Vai al tuo progetto.
2. Utilizza la barra di ricerca per cercare problemi o informazioni relative ai problemi. È possibile utilizzare i parametri di interrogazione per affinare la ricerca. Per ulteriori informazioni, consulta [Cerca codice, problemi, progetti e utenti in CodeCatalyst](#).

Problemi di ordinamento

Per impostazione predefinita, i problemi in CodeCatalyst sono ordinati in base all'ordine manuale. L'ordine manuale mostra i problemi nell'ordine in cui vengono spostati dagli utenti. Puoi trascinare e rilasciare i problemi se ordinati in Manuale per modificarne l'ordine. Questa opzione di ordinamento è utile per gestire l'arretrato dei problemi e assegnare priorità ai problemi.

La tabella seguente mostra come è possibile ordinare i problemi sia nella visualizzazione a griglia che in quella a bacheca.

Opzioni di ordinamento della visualizzazione a griglia	Opzioni di ordinamento della visualizzazione della bacheca
Ordine manuale	Ordine manuale
Ultimo aggiornamento	Ultimo aggiornamento
Priorità	Priorità
Stima	Stima
Titolo	Titolo
ID	
Status	
Bloccato	
Campi personalizzati	

Utilizzare la procedura seguente per modificare la modalità di ordinamento dei problemi.

Per ordinare i problemi

1. Vai al tuo progetto.
2. Nel riquadro di navigazione, scegli Problemi. La visualizzazione predefinita è la bacheca.
3. (Facoltativo) Scegli Problemi attivi per aprire il menu a discesa del commutatore di visualizzazione dei problemi e passare a una visualizzazione diversa dei problemi.
4. Per ordinare una visualizzazione a griglia, sono disponibili due opzioni:
 - a. Scegli l'intestazione del campo in base al quale desideri ordinare. La scelta dell'intestazione passerà dall'ordine crescente a quello decrescente.
 - b. Scegli il menu a discesa Ordina per e scegli un parametro in base al quale ordinare. I problemi verranno ordinati in ordine crescente.
5. Per ordinare una visualizzazione della bacheca, scegli il menu a discesa Ordina per e scegli un parametro in base al quale ordinare. I problemi verranno ordinati in ordine crescente.

Problemi di raggruppamento

Il raggruppamento viene utilizzato per organizzare i problemi sulla lavagna in base a più parametri, come assegnatario, etichette e priorità.

Per raggruppare i problemi

1. Vai al tuo progetto.
2. Nel riquadro di navigazione, scegli Problemi. La visualizzazione predefinita è la bacheca.
3. (Facoltativo) Scegli Problemi attivi per aprire il menu a discesa del commutatore di visualizzazione dei problemi e passare a una visualizzazione diversa dei problemi.
4. Scegli Gruppo.
5. In Raggruppa per, scegli un parametro in base al quale raggruppare:
 - Se scegli Assegnatario o Priorità, scegli l'ordine di gruppo.
 - Se scegli Etichetta, scegli le etichette, quindi scegli Ordine di gruppo.
6. (Facoltativo) Scegli l'opzione Mostra gruppi vuoti per mostrare o nascondere i gruppi a cui non sono attualmente assegnati problemi.
7. La visualizzazione si aggiorna man mano che effettui le tue scelte. Un problema viene visualizzato solo nel gruppo che corrisponde ai parametri configurati.

Problemi di filtraggio

Utilizza il filtro per trovare problemi che contengono un nome, una priorità, un'etichetta, campi personalizzati o un assegnatario specifici.

Per filtrare i problemi

1. Vai al tuo progetto.
2. Nel riquadro di navigazione, scegli Problemi.
3. (Facoltativo) Scegli Problemi attivi per aprire il menu a discesa dello switch di visualizzazione dei problemi e passare a una visualizzazione diversa dei problemi.

Note

Per filtrare in base a una stringa nel nome o nella descrizione del problema, inserisci la stringa nella barra di ricerca dei problemi.

4. Scegli Filtro, quindi scegli + Aggiungi filtro.
5. Scegli i parametri in base ai quali filtrare. Puoi scegliere più filtri e parametri. Puoi configurare i filtri per mostrare i problemi che corrispondono a ogni filtro o a ogni singolo filtro selezionando e o. La visualizzazione verrà aggiornata per mostrare i problemi che corrispondono al filtro.

Progresso di un problema

Ogni problema ha un ciclo di vita. Nel CodeCatalyst, i problemi in genere iniziano come bozza nel backlog. Quando si deve iniziare a lavorare su quel problema, questo viene spostato in un'altra categoria di status e passa da uno stato all'altro fino a quando non è completo, quindi viene archiviato. È possibile spostare o far progredire un problema durante il suo ciclo di vita nei seguenti modi:

- Puoi spostare un problema tra il backlog e la bacheca.
- Puoi spostare un problema in corso attraverso varie fasi di completamento.
- Puoi archiviare un problema completato.

Spostamento di un problema tra backlog e board

Puoi spostare un problema dal backlog alla bacheca una volta che inizi a lavorare sul problema. Puoi anche spostare nuovamente un problema nel backlog se il lavoro viene posticipato.

Per spostare un problema tra il backlog e la bacheca

1. Vai al tuo progetto.
2. Nel riquadro di navigazione, scegli Problemi. La visualizzazione predefinita è la bacheca.
3. Per spostare un problema dalla bacheca al backlog:
 - a. Scegli il problema che desideri spostare. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
 - b. Scegli Backlog dal menu a tendina Status.
4. Per spostare un problema dal backlog alla bacheca:
 - a. Per accedere al backlog, scegli Board e scegli Backlog.
 - b. Scegli il problema che desideri spostare. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
 - c. Scegli Aggiungi alla bacheca o scegli uno stato diverso da Backlog.

Fai progredire un problema attraverso le fasi del ciclo di vita sulla lavagna

Puoi spostare un problema all'interno di una bacheca attraverso diversi stati fino al completamento.

Per spostare un problema all'interno della bacheca

1. Nel riquadro di navigazione, scegli Problemi. La visualizzazione predefinita è la bacheca.
2. Esegui una delle seguenti operazioni:
 - Trascina e rilascia un problema su un altro stato.
 - Scegli un problema, quindi scegli uno stato dal menu a discesa Stato.
 - Scegli un problema, quindi scegli Sposta in: **next-status**.

Per informazioni sull'archiviazione di un problema, consulta [Archiviazione di un problema](#).

Spostamento di problemi tra gruppi

Puoi [raggruppare i problemi](#) nelle visualizzazioni Tutti i problemi e Bacheca in base a vari parametri. Se i problemi sono raggruppati, puoi spostarli da un gruppo all'altro. Lo spostamento di un problema da un gruppo a un altro modificherà automaticamente il campo in cui i problemi sono raggruppati in modo che corrisponda al gruppo target.

Come scenario di esempio, supponiamo che un'azienda CodeCatalyst che lo utilizza abbia assegnato i problemi a due persone, Wang Xiulan e Saanvi Sarkar. Il consiglio di amministrazione è raggruppati per e vi sono due gruppiAssignee, uno per ogni assegnatario. Lo spostamento di un numero dal gruppo Wang Xiulan al gruppo Saanvi Sarkar aggiornerà l'assegnatario del problema a Saanvi Sarkar.

Archiviazione di un problema

Note

I problemi non vengono eliminati all'interno di un progetto, ma archiviati. Per eliminare i problemi, è necessario eliminare il progetto.

È possibile archiviare un problema quando non è più necessario nel progetto. Quando archivi un problema, lo CodeCatalyst rimuove da tutte le visualizzazioni che filtrano i problemi archiviati. I problemi archiviati possono essere visualizzati nella visualizzazione predefinita Problemi archiviati, dove possono essere recuperati se necessario.

Un problema viene archiviato se:

- Hai completato il problema e non ne hai più bisogno nella colonna Fine.
- Non hai intenzione di lavorarci.
- L'hai creato per errore.
- Hai raggiunto il numero massimo di problemi attivi.

Per archiviare un problema

1. Apri il problema che desideri archiviare. Per assistenza nella ricerca del problema, consulta [Individuazione e visualizzazione dei problemi](#).
2. Scegli Azioni, quindi scegli Sposta nell'archivio.

3. (Facoltativo) Per archiviare rapidamente più problemi con lo stato Completato, scegli i puntini di sospensione verticali nella parte superiore di qualsiasi stato Completato sulla lavagna e scegli Archivia problemi.

Per annullare l'archiviazione di un problema

1. Apri il problema che desideri estrarre dall'archiviazione. Puoi visualizzare un elenco dei problemi archiviati aprendo la visualizzazione Problemi archiviati dal menu a discesa del selettore di visualizzazione dei problemi. Per assistenza nella ricerca del problema, consulta. [Individuazione e visualizzazione dei problemi](#)
2. Scegli Unarchive.

Problemi di esportazione

È possibile esportare i problemi nella visualizzazione corrente in un file.xlsx. Per risolvere i problemi relativi all'esportazione, effettuate le seguenti operazioni.

Per esportare problemi

1. Vai al tuo progetto.
2. Nella barra di navigazione scegli Problemi.
3. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e vai alla visualizzazione contenente i problemi che desideri esportare. Verranno esportati solo i problemi mostrati nella vista.
4. Scegliete il menu con i puntini di sospensione e scegliete Esporta in Excel.
5. Il file.xlsx viene scaricato. Per impostazione predefinita, è intitolato il nome del progetto e la data di completamento dell'esportazione.

Organizzazione del lavoro con arretrati, etichette e bacheche

Non tutti i team lavorano allo stesso modo. Puoi configurare il modo in cui i problemi appaiono e possono essere assegnati in Amazon CodeCatalyst per aiutarti a capire con precisione su cosa si sta lavorando e lo stato di tale lavoro. Puoi scegliere il metodo di stima da utilizzare in caso di problemi, in modo che tutti gli utenti utilizzino lo stesso metodo di stima. Puoi creare etichette e stati personalizzati che possono essere utilizzati anche per filtrare la visualizzazione dell'opera. A seconda di come lavora il team, puoi configurare se consentire a più assegnatari per un problema o

consentire l'assegnazione di un problema solo a un singolo utente. Puoi anche creare visualizzazioni personalizzate dei problemi in modo che il lavoro venga visualizzato in modo da mostrare le informazioni più pertinenti per te o per il tuo team.

Categorizzazione del lavoro con le etichette

È possibile personalizzare le etichette in base ai problemi. Ciò include la modifica dell'etichetta e il cambio del colore. Le etichette possono aiutarti a classificare e organizzare il tuo lavoro. Ad esempio, è possibile creare etichette per aspetti specifici del software o per gruppi o team diversi.

Argomenti

- [Creare un'etichetta](#)
- [Modificare un'etichetta](#)
- [Eliminazione di un'etichetta](#)

Creare un'etichetta

In CodeCatalyst, puoi creare etichette aggiungendole quando crei un nuovo numero o quando modifichi un numero esistente. Per ulteriori informazioni, consultare [Creazione di un problema in CodeCatalyst](#) e [Redazione e collaborazione su problemi in CodeCatalyst](#).

Modificare un'etichetta

Utilizzare la procedura seguente per modificare il nome o il colore di un'etichetta esistente.

Per modificare un'etichetta

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. Nel riquadro Etichette c'è un elenco delle etichette utilizzate nel progetto. Scegliete l'icona di modifica accanto all'etichetta che desiderate modificare. Effettuare una o più delle seguenti operazioni:
 - a. Modifica il nome dell'etichetta.
 - b. Per cambiare il colore, scegli la ruota dei colori. Usa il selettore per scegliere un nuovo colore.
4. Per salvare le modifiche apportate all'etichetta, scegli l'icona del segno di spunta.

5. L'etichetta modificata è ora visibile nell'elenco delle etichette disponibili. Puoi anche vedere quanti problemi stanno usando quell'etichetta.

Note

Puoi scegliere il numero visualizzato accanto a ciascuna etichetta per accedere alla pagina Tutti i problemi e visualizzare tutti i problemi che contengono quell'etichetta.

Eliminazione di un'etichetta

Al momento non è possibile eliminare un'etichetta relativa ai problemi in CodeCatalyst. Se rimuovi un'etichetta da tutti i numeri, l'etichetta verrà visualizzata nella sezione Etichette non utilizzate della sezione Etichette delle impostazioni del numero. Le etichette non utilizzate vengono visualizzate alla fine dell'elenco delle etichette quando si utilizzano filtri o si aggiungono etichette a un numero. Puoi trovare una panoramica di tutte le etichette (usate e non utilizzate) e dei problemi che le presentano nelle impostazioni del problema.

Organizzazione del lavoro con campi personalizzati

Puoi creare campi personalizzati per organizzare e visualizzare il lavoro relativo al tuo progetto. I campi personalizzati vengono aggiunti all'elenco dei filtri disponibili in Filtro in modo da poter filtrare i problemi in base ai campi personalizzati. I campi personalizzati sono coppie di nomi e valori. Si filtra in base al nome del campo personalizzato e quindi al valore di tale campo personalizzato.

Un problema può avere più campi personalizzati.

Creazione di un campo personalizzato

In CodeCatalyst, puoi creare campi personalizzati aggiungendoli quando crei un problema o quando modifichi un problema esistente. Per ulteriori informazioni, consultare [Creazione di un problema in CodeCatalyst](#) e [Redazione e collaborazione su problemi in CodeCatalyst](#).

Eliminazione di un campo personalizzato

Per eliminare un campo personalizzato, è necessario rimuovere il campo personalizzato da ogni numero a cui viene aggiunto. Quando un campo personalizzato viene eliminato, non verrà più visualizzato in Filtro. Puoi utilizzare i filtri per visualizzare tutti i problemi relativi a un campo personalizzato e rimuoverli modificando i problemi. Per ulteriori informazioni, consulta [Individuazione e visualizzazione dei problemi](#) e [Modificare un problema](#).

Il monitoraggio funziona con stati personalizzati

Puoi aggiungere stati personalizzati sulla tua bacheca. Ogni stato personalizzato deve appartenere a una delle seguenti categorie: Bozza, Non iniziata, Avviata o Completata. Le categorie di stato vengono utilizzate per organizzare gli stati e compilare le visualizzazioni predefinite. Per ulteriori informazioni sugli stati e le categorie di stato, vedere [Stato e categorie di stato](#) e per ulteriori informazioni sulle visualizzazioni, vedere [Individuazione e visualizzazione dei problemi](#)

Per creare uno stato

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. In Stati, scegli l'icona con il segno più accanto alla categoria in cui desideri inserire lo stato.
4. Assegna un nome allo stato, quindi scegli l'icona del segno di spunta.

Note

Scegli l'icona X per annullare l'aggiunta di uno stato.

Lo stato personalizzato è ora visibile sulla tua bacheca e viene visualizzato come opzione quando crei un problema.

Per modificare uno stato

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. In Stati, scegli l'icona di modifica accanto allo stato che desideri modificare o modificare.
4. Modifica lo stato, quindi scegli l'icona del segno di spunta.

Lo stato modificato è ora visibile sulla tua bacheca.

Per spostare uno stato

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli l'icona con i puntini di sospensione e scegli Impostazioni.

3. In Stati, scegli lo stato che desideri spostare.
4. Trascina e rilascia lo stato dove vuoi che sia.

Note

Puoi spostare uno stato solo all'interno della categoria designata.

Gli stati sono ora riordinati sulla tua bacheca.

Per disattivare uno stato

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. In Stati, scegli lo stato che desideri disattivare.
4. Sullo stato che desideri disattivare, scegli l'interruttore sullo stato. Lo stato ora è disattivato.

Note

Lo stato di disattivazione appare sulla bacheca finché non vengono rimossi tutti i problemi. I problemi non possono essere aggiunti a uno stato disattivato.

5. Per riattivare uno stato disattivato, scegli l'interruttore sullo stato. Lo stato non è più visualizzato in grigio.

Note

Deve esserci almeno uno stato attivo in ogni categoria. Se nella categoria è presente un solo stato, non è possibile disattivarlo.

Configurazione della stima dello sforzo da risolvere

Segui questi passaggi per configurare l'impostazione per le stime dello sforzo per i problemi in CodeCatalyst

Per configurare la stima dello sforzo per i problemi

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. Nella sezione Stima, nella sezione Impostazioni di base, scegli come verranno visualizzati i valori di stima. I tipi di stime disponibili sono il dimensionamento delle magliette, il sequenziamento di Fibonacci o le stime Hide. Quando il tipo di stima viene aggiornato, nessun dato andrà perso e il valore di stima di tutti i numeri verrà convertito automaticamente. La mappatura delle conversioni è mostrata nella tabella seguente.

Taglia della maglietta	Sequenza di Fibonacci
XS	1
XS	2
S	3
M	5
L	8
XL	13

Abilitazione o disabilitazione di più assegnatari

Segui questi passaggi per configurare l'impostazione per più assegnatari per i problemi in CodeCatalyst

Per abilitare o disabilitare più assegnatari

1. Nel riquadro di navigazione, scegli Problemi.
2. Scegli Problemi attivi per aprire il menu a discesa Issues View Switcher e scegli Impostazioni.
3. In Assegnatario, nella sezione Impostazioni di base, attiva l'indicatore per consentire a più assegnatari di essere assegnati allo stesso problema. Un problema può avere fino a 10 assegnatari. Se non abiliti questa opzione, potrai assegnare un solo assegnatario a un problema.

Creazione di una visualizzazione dei problemi

È possibile creare [visualizzazioni](#) per visualizzare rapidamente i problemi che corrispondono a un particolare set di filtri. In questo modo è possibile risparmiare tempo e visualizzare rapidamente i problemi precedentemente filtrati, raggruppati o ordinati.

Per creare una visualizzazione dei problemi

1. Nel riquadro di navigazione, scegli Problemi.
2. (Facoltativo) A seconda del caso d'uso, potresti voler creare una vista da una vista esistente. Per passare a una visualizzazione diversa, scegli Problemi attivi per aprire il menu a discesa del commutatore di visualizzazione dei problemi e scegli la visualizzazione.
3. (Facoltativo) Configura filtri, raggruppamento e ordinamento prima di creare la visualizzazione. Puoi aggiungerli durante la creazione di una vista, ma se lo fai prima, puoi vedere in anteprima ciò che viene mostrato nella vista prima di crearla.
4. Apri il menu a discesa Issues View Switcher dalla barra dell'intestazione. Per creare una visualizzazione della bacheca in cui i problemi vengono visualizzati in colonne in base allo stato, scegli il + nella colonna Bacheca. Per creare una visualizzazione a griglia in cui i problemi vengono visualizzati in un elenco, scegli il + nella colonna Griglia. Puoi cambiare il tipo di visualizzazione prima che venga creata se cambi idea.
5. Nella finestra di dialogo Crea visualizzazione, immettete un nome per la vista.
6. I campi Filtri, Raggruppa problemi per e Ordina problemi per vengono compilati in base alle impostazioni della visualizzazione corrente. Aggiornateli se necessario.
7. Scegliete Crea vista per creare la vista e passare ad essa.

Quote per le emissioni in CodeCatalyst

La tabella seguente descrive le quote e i limiti per i problemi in Amazon CodeCatalyst. Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta.. [Quote per CodeCatalyst](#)

Risorsa	Quota predefinita
Problemi attivi	Massimo 1.000 per progetto.
Dimensioni degli allegati	Massimo 500 MB per allegato.

Risorsa	Quota predefinita
	Lo spazio massimo totale di archiviazione degli allegati è influenzato dai limiti di archiviazione complessivi per lo spazio disponibile. Per ulteriori informazioni, consultare Prezzi .
Numero totale di problemi (attivi e archiviati)	Massimo 100.000 per progetto.
Visualizzazioni salvate	Massimo 50 visualizzazioni di numeri salvate per progetto.
Numero di pull request che puoi collegare a un problema	Massimo 50 pull request per numero.
Stati (per progetto)	Massimo 50 per progetto.
Stati (per numero)	Massimo 50 per numero.
Etichette (per progetto)	Massimo 200 per progetto.
Etichette (per numero)	Massimo 50 per numero.
Campi personalizzati (per numero)	Massimo 50 per numero.
Assegnatari	Massimo 10 per numero.
Commenti	Massimo 1.000 per numero.
Processi	Massimo 100 per numero.

Configura identità, autorizzazioni e accesso in CodeCatalyst

Quando accedi ad Amazon CodeCatalyst per la prima volta, crei un AWS Builder ID. AWS I Builder IDs non esistono in. AWS Identity and Access Management Il nome utente che scegli durante il primo accesso diventa l'ID utente univoco per la tua identità.

In CodeCatalyst, puoi accedere per la prima volta in due modi:

- Come parte della creazione di uno spazio.
- Come parte dell'accettazione di un invito a un progetto o a uno spazio in CodeCatalyst.

Il ruolo o i ruoli associati alla tua identità determinano le azioni che puoi eseguire CodeCatalyst. I ruoli di progetto, ad esempio Amministratore di progetto e Collaboratore, sono specifici di un progetto, quindi è possibile avere un ruolo in un progetto e un ruolo diverso in un altro progetto. Se crei uno spazio, ti assegna CodeCatalyst automaticamente il ruolo di amministratore dello spazio. Quando gli utenti accettano inviti a un progetto, CodeCatalyst aggiunge tali identità allo spazio e assegna loro il ruolo di accesso limitato. Quando inviti gli utenti ai progetti, scegli il ruolo che desideri che abbiano nel progetto, che determina quali azioni possono e non possono intraprendere all'interno del progetto. La maggior parte degli utenti che lavorano a un progetto necessita solo del ruolo di Collaboratore per svolgere le proprie attività. Per ulteriori informazioni, consulta [Concessione dell'accesso con ruoli utente](#).

Oltre a un ruolo di progetto, gli utenti di un progetto necessitano di un token di accesso personale (PAT) per accedere ai repository dei sorgenti di un progetto quando utilizzano client Git o ambienti di sviluppo integrati (IDEs). I membri del progetto possono utilizzare questo PAT con applicazioni di terze parti come password specifica dell'applicazione associata alla propria identità. CodeCatalyst Ad esempio, quando cloni un repository di origine su un computer locale, devi fornire un PAT oltre al tuo nome utente. CodeCatalyst

È possibile configurare l'accesso tra AWS le risorse CodeCatalyst e utilizzando un [ruolo di servizio](#) per eseguire azioni come l'accesso a AWS CloudFormation stack e risorse quando si distribuiscono azioni nei flussi di lavoro. È necessario configurare l'accesso tra CodeCatalyst e AWS le risorse per le azioni del flusso di lavoro incluse nei modelli di progetto da eseguire.

Argomenti

- [Concessione dell'accesso con ruoli utente](#)
- [Concedi agli utenti l'accesso al repository con token di accesso personali](#)

- [Accesso alle GitHub risorse con connessioni personali](#)
- [Configura il tuo AWS Builder ID per accedere con l'autenticazione a più fattori \(MFA\)](#)
- [Sicurezza in Amazon CodeCatalyst](#)
- [Monitoraggio degli eventi e delle chiamate API tramite la registrazione](#)
- [Quote per identità, autorizzazione e accesso in CodeCatalyst](#)
- [Risoluzione dei problemi](#)

Concessione dell'accesso con ruoli utente

In Amazon CodeCatalyst, puoi assegnare ruoli agli utenti sia a livello di progetto che a livello di spazio. In un progetto, un ruolo specifica ciò che un utente è autorizzato a fare in un progetto con le risorse per quel progetto. Gli utenti ottengono l'appartenenza a uno spazio quando partecipano a un progetto. È possibile aggiungere o rimuovere utenti come amministratori di uno spazio. Il ruolo di amministratore dello spazio dispone delle autorizzazioni più ampie di qualsiasi altro ruolo in CodeCatalyst. È consigliabile assegnare agli utenti le autorizzazioni più limitate necessarie per svolgere il proprio lavoro.

È possibile assegnare ruoli agli utenti nello spazio. Puoi anche assegnare ruoli agli utenti nei progetti in cui sono membri. Ogni utente può avere un solo ruolo in un progetto o spazio, ma gli utenti possono avere ruoli diversi in ogni progetto e spazio. Ad esempio, un utente potrebbe avere il ruolo di amministratore di progetto in un progetto e il ruolo di collaboratore in un altro progetto.

Argomenti

- [Comprendere i ruoli degli utenti per spazi e progetti](#)
- [Visualizzazione delle autorizzazioni disponibili per ogni ruolo](#)
- [Visualizzazione e modifica dei ruoli utente](#)

Comprendere i ruoli degli utenti per spazi e progetti

Sono disponibili tre ruoli per uno spazio:

- Amministratore dello spazio
- Utente avanzato
- Accesso limitato

Agli utenti che accettano un invito a un progetto viene assegnato automaticamente il ruolo di accesso limitato nello spazio che contiene il progetto.

Sono disponibili quattro ruoli per i membri di un progetto:

- Amministratore del progetto
- Collaboratore
- Revisore
- Sola lettura

Quando aggiungi un utente a un progetto, gli assegna CodeCatalyst automaticamente il ruolo di accesso limitato. Se rimuovi un utente da tutti i progetti, rimuove CodeCatalyst automaticamente il ruolo di accesso limitato da quell'utente.

Ruolo di amministratore dello spazio

Il ruolo di amministratore dello spazio è il ruolo più importante in CodeCatalyst. Assegna il ruolo di amministratore dello spazio solo agli utenti che devono amministrare ogni aspetto di uno spazio, poiché questo ruolo dispone di tutte le autorizzazioni. CodeCatalyst Gli utenti con il ruolo di amministratore dello spazio sono gli unici utenti che possono aggiungere o rimuovere altri utenti dal ruolo di amministratore dello spazio ed eliminare lo spazio.

Quando crei uno spazio, ti assegna CodeCatalyst automaticamente il ruolo di amministratore dello spazio. Come procedura ottimale, si consiglia di aggiungere questo ruolo ad almeno un altro utente che possa ricoprire questo ruolo nel caso in cui il creatore dello spazio originale non sia disponibile.

Ruolo di Power User

Il ruolo Power user è il secondo ruolo più importante negli CodeCatalyst spazi, ma non ha accesso ai progetti in uno spazio. È progettato per gli utenti che devono essere in grado di creare progetti in uno spazio e aiutare a gestire gli utenti e le risorse dello spazio. Assegna il ruolo Power user agli utenti che sono team leader o manager che hanno bisogno della capacità di creare progetti e gestire gli utenti nello spazio come parte del loro lavoro.

Ruolo ad accesso limitato

Il ruolo di accesso limitato è il ruolo che la maggior parte degli utenti avrà negli CodeCatalyst spazi. È il ruolo assegnato automaticamente agli utenti quando accettano un invito a un progetto in uno

spazio. Fornisce le autorizzazioni limitate di cui hanno bisogno per lavorare all'interno dello spazio che contiene quel progetto. Assegna il ruolo di accesso limitato agli utenti che inviti direttamente nello spazio, a meno che il loro lavoro non richieda la gestione di alcuni aspetti dello spazio.

Ruolo di amministratore del progetto

Il ruolo di amministratore di progetto è il ruolo più importante in un CodeCatalyst progetto. Assegna questo ruolo solo agli utenti che devono amministrare ogni aspetto di un progetto, inclusa la modifica delle impostazioni del progetto, la gestione delle autorizzazioni del progetto e l'eliminazione dei progetti.

I ruoli del progetto non dispongono di autorizzazioni a livello di spazio. Pertanto, gli utenti con il ruolo di amministratore del progetto non possono creare progetti aggiuntivi. Solo gli utenti con il ruolo di amministratore dello spazio o utente esperto possono creare progetti.

Note

Il ruolo di amministratore dello spazio dispone di tutte le autorizzazioni. CodeCatalyst

Ruolo di collaboratore

Il ruolo di Collaboratore è destinato alla maggior parte dei membri di un CodeCatalyst progetto. Assegna questo ruolo agli utenti che devono essere in grado di lavorare con codice, flussi di lavoro, problemi e azioni in un progetto.

Ruolo di revisore

Il ruolo Revisore è destinato agli utenti che devono essere in grado di interagire con le risorse di un progetto, come richieste pull e problemi, ma non creare e unire codice, creare flussi di lavoro o avviare o interrompere l'esecuzione del flusso di lavoro in un progetto. CodeCatalyst Assegna il ruolo Revisore agli utenti che devono essere in grado di approvare e commentare le richieste pull, creare, aggiornare, risolvere e commentare i problemi e visualizzare codice e flussi di lavoro in un progetto.

Ruolo di sola lettura

Il ruolo di sola lettura è destinato agli utenti che devono visualizzare le risorse e lo stato delle risorse ma non interagire con esse o contribuire direttamente al progetto. Gli utenti con questo ruolo non possono creare risorse in CodeCatalyst, ma possono visualizzarle e copiarle, ad esempio clonando

gli archivi e scaricando gli allegati ai problemi su un computer locale. Assegna il ruolo di sola lettura agli utenti che devono visualizzare le risorse e lo stato del progetto, ma non interagire direttamente con esso.

Visualizzazione delle autorizzazioni disponibili per ogni ruolo

La tabella seguente mostra le autorizzazioni disponibili per ogni CodeCatalyst ruolo. Usa i link per passare al set di autorizzazioni appropriato.

- [Space permissions](#)
- [Extensions permissions](#)
- [Project permissions](#)
- [Source repository permissions](#)
- [Dev Environment permissions](#)
- [Package repository and package permissions](#)
- [Workflow permissions](#)
- [Issues permissions](#)
- [Blueprint permissions](#)
- [Notifications permissions](#)
- [Search permissions](#)

Autorizza zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura

Autorizzazioni spaziali

Crea spazio							
Modifica i dettagli di fatturazi							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
one dello spazio							
Configura e abilita il Single Sign-On							
Rimuovi il single sign-on							
Abilita le funzionalità di intelligenza artificiale generativa per uno spazio							
Disattiva le funzionalità di intelligenza artificiale generativa per uno spazio							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Eliminare lo spazio							
Aggiungere e altri utenti al ruolo di amministratore dello spazio							
Rimuovi altri utenti dal ruolo di amministratore dello spazio							
Crea un team							
Eliminare il team							
Aggiorna team							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Disattiva le risorse della macchina per lo spazio							
Abilita le risorse della macchina per lo spazio							
Crea progetto							
Associa una connessione a un account AWS allo spazio							
Aggiornare la connessione di un account AWS							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Dissocia la connessione di un account AWS dallo spazio							
Eliminare una connessione a un account AWS e rimuoverla dallo spazio							
Abilita le connessioni agli account con restrizioni del progetto nello spazio ¹							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Disattiva le connessioni agli account con restrizioni del progetto nello spazio ²							
Invita altre persone nello spazio							
Crea una connessione VPC							
Modifica connessione VPC							
Eliminare la connessione VPC							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza i registri delle attività nello spazio							
Visualizza le connessioni degli account AWS							
Visualizza gli incidenti per CodeCatalyst							
Visualizza lo spazio							
Visualizza i team							
Visualizza le connessioni VPC							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura

¹ Con il ruolo Power user, puoi abilitare le restrizioni di progetto per un account, ma puoi configurare l'accesso solo per i progetti di cui sei membro.

² Con il ruolo Power user, puoi disabilitare le restrizioni di progetto per un account, ma puoi configurare l'accesso solo per i progetti di cui sei membro.

Estensioni e autorizzazioni	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Installa le estensioni							
Aggiorna le estensioni							
Eliminare le estensioni							
Connect a un GitHub account							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Disconnettere un account GitHub							
Connect a un sito Jira							
Disconnetti un sito Jira							
Visualizza i dettagli di configurazione per le estensioni installate							
Visualizza le estensioni							

Autorizza zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Autorizzazioni del progetto	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Modificare le impostazioni del progetto							
Disattivare le risorse della macchina per il progetto							
Abilitare le risorse della macchina per il progetto							
Eliminare il progetto							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Invitare utenti a un progetto							
Modifica i ruoli degli utenti in un progetto							
Rimuovere utenti da un progetto							
Aggiungere un team a un progetto							
Rimuovi team da un progetto							
Cambia il ruolo del team nel progetto							

Autorizza- zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza il progetto							
Visualizza l'attività del progetto							
Visualizza i team del progetto							
Visualizza i progetti							
Autorizzazioni del repository di origine	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea repository							
Collega repository							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Scollegare i repository							
Eliminare i repository							
Modificare le impostazioni del repository							
Visualizzare i repository							
Visualizzare le impostazioni del repository							
Clonazione di repository							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea filiali							
Crea regole per le filiali							
Cambia ramo predefinito							
Eliminare i rami							
Unione dei rami							
Aggiorna le regole delle filiali							
Visualizza le filiali							
Visualizza le regole delle filiali							
Crea cartelle							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Eliminare cartelle							
Modifica cartelle							
Visualizza cartelle							
Creare file							
Eliminazione del file							
Modificare i file							
Visualizza i file							
Creare e inviare commit							
Visualizza i commit							
Creare richieste pull							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea regole di approvazione per le richieste pull							
Sostituisci i requisiti di unione per le richieste pull							
Aggiorna le richieste pull							
Aggiorna le regole di approvazione per le richieste pull							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizzazione delle richieste pull							
Visualizzazione delle regole di approvazione per le richieste pull							
Chiusura delle richieste pull							
Approvazione delle richieste pull							
Commentare le pull request							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Interagisci con Amazon Q nei commenti sulle pull request							
Creare una revisione per una pull request creata da Amazon Q							
Collega un problema a una pull request							
Scollega un problema da una pull request							

Autorizza zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Autorizzazioni Dev Environment	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea il tuo ambiente di sviluppo							
Ferma il tuo ambiente di sviluppo							
Stop Dev Environments creati da altri utenti							
Riprendi il tuo ambiente di sviluppo							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza i tuoi ambienti di sviluppo							
Visualizza gli ambienti di sviluppo creati da altri utenti							
Modifica il tuo ambiente di sviluppo							
Modifica gli ambienti di sviluppo creati da altri utenti							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Elimina il tuo ambiente di sviluppo							
Elimina gli ambienti di sviluppo creati da altri utenti							
Crea un devfile per gli ambienti di sviluppo							
Modifica un devfile per gli ambienti di sviluppo							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Eliminare un devfile per gli ambienti di sviluppo							
Visualizza a un devfile per ambienti di sviluppo							
Archivio dei pacchetti e autorizzazioni dei pacchetti							
Crea un archivio di pacchetti							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza gli archivi dei pacchetti							
Modifica l'archivio o dei pacchetti							
Elimina l'archivio o dei pacchetti							
Crea un repository di pacchetti gateway							
Visualizza gli archivi dei pacchetti gateway							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Elimina l'archivio dei pacchetti del gateway							
Aggiungi un archivio di pacchetti upstream							
Modifica l'ordine di ricerca dei repository upstream							
Rimuovi l'archivio dei pacchetti upstream							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Connect a un repository di pacchetti							
Leggi i pacchetti da un repository di pacchetti							
Pubblica pacchetti in un repository di pacchetti							
Leggi e conserva i pacchetti da un repository upstream							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza i pacchetti							
Visualizza le versioni dei pacchetti							
Visualizza le risorse delle versioni del pacchetto							
Elenca le dipendenze della versione del pacchetto							
Aggiorna lo stato della versione del pacchetto							

Autorizza- zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Aggiorna la configurazione dell'origine del pacchetto							
Elimina la versione del pacchetto							
Autorizzazioni del flusso di lavoro	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea flusso di lavoro							
Aggiorna flusso di lavoro							
Elimina flusso di lavoro							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Avvia flusso di lavoro							
Interrompi il workflow							
Crea segreti per il flusso di lavoro							
Aggiorna i segreti del workflow							
Elimina i segreti del flusso di lavoro							
Crea ambienti.							
Eliminare gli ambienti							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea una flotta							
Aggiorna la flotta							
Elimina flotta							
Gestisci le risorse del flusso di lavoro in altri account							
Associa una Account AWS connessione a un ambiente							
Associa un ruolo IAM predefinito a un ambiente							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Associare una connessione VPC a un ambiente							
Dissociare una connessione VPC da un ambiente							
Associa un ambiente connesso al VPC a un flusso di lavoro							
Dissocia un ambiente connesso a VPC con un flusso di lavoro							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Approva le esecuzioni del flusso di lavoro							
Tieni traccia di un commit in un flusso di lavoro							
Visualizza gli ambienti							
Visualizza i registri delle azioni di compilazione							
Visualizza le flotte							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza i registri delle azioni di test							
Visualizza il flusso di lavoro							
Visualizza i flussi di lavoro eseguiti							
Visualizza i risultati dell'esecuzione del flusso di lavoro							
Visualizza i segreti del flusso di lavoro							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Rilascia le autorizzazioni	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea problema							
Problema di aggiornamento							
Visualizza i problemi							
Create task (Crea attività)							
Attività di aggiornamento							
Visualizza le attività							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Archivia un problema							
Assegna un problema ad Amazon Q							
Interagisci con Amazon Q nei commenti su un problema							
Annullare l'assegnazione di Amazon Q a un problema							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Consiglia attività per un problema con Amazon Q							
Crea attività consigliate da Amazon Q							
Aggiorna i problemi creati da altri utenti							
Visualizza i commenti su un problema							
Creare un commento su un problema							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Aggiornare un commento su un problema							
Crea un'etichetta							
Aggiornare un'etichetta							
Visualizzare le etichette							
Aggiungere un'etichetta a un problema							
Rimuovere un'etichetta da un problema							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea uno stato personalizzato per i problemi							
Aggiorna uno stato personalizzato							
Visualizza uno stato personalizzato							
Sposta uno stato personalizzato							
Disattiva uno stato personalizzato							
Aggiungere e un allegato a un problema							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza a un allegato al problema							
Rimuovere un allegato da un problema							
Collega un problema a un altro							
Scollegare un problema da un altro							
Aggiorna il link di un problema							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Visualizza i link relativi a un problema							
Collega una pull request a un problema							
Scollegare una pull request da un problema							
Collega un progetto Jira							
Scollega un progetto Jira							

Autorizza- zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Autorizzazioni Blueprint	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Crea un progetto blueprint personalizzato							
Pubblica un'anteprima di un blueprint personalizzato							
Pubblica un blueprint personalizzato							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Aggiungi un progetto personalizzato a un catalogo di progetti spaziali							
Rimuovi un blueprint personalizzato da un catalogo di space blueprint							
Gestisci le autorizzazioni di pubblicazione per un blueprint personalizzato							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Gestisci la versione del catalogo per un blueprint personalizzato							
Aggiorna un blueprint personalizzato							
Eliminare una versione personalizzata del blueprint							
Eliminare un blueprint personalizzato							

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Convertire un repository di origine in un blueprint personalizzato							
Aggiungere un blueprint personalizzato a un progetto							
Dissociare un blueprint personalizzato da un progetto							

Autorizza- zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Aggiorna la versione di un blueprint personalizzato applicato							
Modifica le impostazioni di un blueprint personalizzato							
Visualizza i blueprint personalizzati pubblicati							
Autorizzazioni per le notifiche	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura

Autorizzazione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Configura il canale di notifica							
Rimuovi il canale di notifica							
Modifica le impostazioni di notifica							
Visualizza le impostazioni di notifica							
Ricevi automaticamente notifiche sugli incidenti CodeCatalyst							

Autorizza- zione	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Configura le notifiche e-mail per l'account e-mail associato							
Autorizzazioni di ricerca	Ruolo di amministratore dello spazio	Ruolo di Power User	Ruolo ad accesso limitato	Ruolo di amministratore del progetto	Ruolo di collaboratore	Ruolo di revisore	Ruolo di sola lettura
Cerca all'interno di un progetto							
Cerca nello spazio							

Visualizzazione e modifica dei ruoli utente

È possibile visualizzare il ruolo assegnato a un utente. Questo ti aiuta a capire quali azioni possono intraprendere in un progetto. Puoi anche modificare il loro ruolo se necessitano di autorizzazioni aggiuntive.

Per visualizzare il ruolo di un utente in un progetto

1. Vai al progetto in cui desideri visualizzare i ruoli associati a ciascun membro del progetto.

Tip

Puoi scegliere quale progetto visualizzare nella barra di navigazione in alto.

2. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Nella scheda Membri, il ruolo di ogni membro del progetto viene visualizzato in Ruolo.

Per modificare i ruoli degli utenti in un progetto

1. Vai al progetto in cui desideri modificare i ruoli associati ai membri del progetto.

Tip

Puoi scegliere quale progetto visualizzare nella barra di navigazione in alto.

2. Nel riquadro di navigazione, scegli Impostazioni del progetto.
3. Nella scheda Membri, in Membri del progetto, scegli gli utenti di cui desideri modificare i ruoli. Scegli Azione, quindi scegli Modifica ruolo.
4. In Ruolo, scegli il ruolo del progetto, quindi scegli Conferma.

Visualizzazione e modifica dei ruoli nello spazio

Tutti gli utenti che accettano gli inviti a un progetto CodeCatalyst diventano membri dello spazio del progetto. È possibile visualizzare l'elenco dei membri dello spazio. Puoi modificare i ruoli degli utenti da Amministratore con accesso limitato a Space per gestire meglio lo spazio e le relative risorse. Il ruolo di amministratore di Space è l'unico ruolo che consente agli utenti di creare progetti in CodeCatalyst.

Warning

Il ruolo di amministratore dello spazio è il ruolo più importante in CodeCatalyst. Gli utenti con questo ruolo possono eseguire qualsiasi azione CodeCatalyst, inclusa l'eliminazione dello

spazio. Assegna questo ruolo solo agli utenti che richiedono questo livello di accesso al tuo spazio. Per ulteriori informazioni, consulta [Ruolo di amministratore dello spazio](#).

Per modificare il ruolo di un utente nello spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedere allo spazio.

 Tip

Se appartieni a più di uno spazio, puoi scegliere quale spazio visualizzare nella barra di navigazione in alto.

3. Scegli la scheda Membri.
4. Scegli l'utente di cui desideri modificare il ruolo, quindi scegli Cambia ruolo.
5. In Cambia ruolo, scegli il ruolo che desideri assegnare, quindi scegli Conferma.

Concedi agli utenti l'accesso al repository con token di accesso personali

Per accedere ad alcune CodeCatalyst risorse, come gli archivi di sorgenti, su un computer locale con un client Git o un ambiente di sviluppo integrato (IDE), è necessario inserire una password specifica per l'applicazione. È possibile creare un token di accesso personale (PAT) da utilizzare a questo scopo. PATs che crei sono associati alla tua identità utente in tutti gli spazi e i progetti di CodeCatalyst. Puoi creare più di un PAT per la tua CodeCatalyst identità.

Puoi visualizzare i nomi e le date di scadenza dei file PATs che hai creato e puoi eliminare quelli che non ti servono più. È possibile copiare il segreto PAT solo al momento della creazione.

 Note

Per impostazione predefinita, PATs scade dopo 1 anno.

Creando PATs

PATs sono associati alla tua identità utente in CodeCatalyst. È possibile copiare un segreto PAT solo al momento della creazione.

Creazione PATs (console)

Puoi usare la console per creare PATs in CodeCatalyst.

Per creare un token di accesso personale (console)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. In Token di accesso personali, scegli Crea.

Viene visualizzata la pagina Crea PAT.

4. In Nome PAT, inserisci un nome descrittivo per il tuo PAT.
5. In Data di scadenza, mantieni la data predefinita o scegli l'icona del calendario per selezionare una data personalizzata. La data di scadenza predefinita è 1 anno dalla data corrente.
6. Scegli Create (Crea).

Tip

Puoi anche creare questo token quando scegli Clone repository come repository di origine.

7. Per copiare il segreto PAT, scegli Copia. Memorizza il segreto PAT dove potrai recuperarlo.

 Important

Il segreto PAT viene visualizzato solo una volta. Non è possibile recuperarlo dopo aver chiuso la finestra. Se non hai salvato il segreto PAT in una posizione sicura, puoi crearne un altro.

Creazione PATs (CLI)

Puoi usare la CLI per creare PATs . CodeCatalyst

Per creare un token di accesso personale ()AWS CLI

1. Nel terminale o nella riga di comando, esegui il create-access-token comando come segue.

```
aws codecatalyst create-access-token
```

In caso di successo, il comando restituisce informazioni sul PAT creato, come nell'esempio seguente.

```
{
  "secret": "value",
  "name": "marymajor-22222EXAMPLE",
  "expiresTime": "2024-02-04T01:56:04.402000+00:00"
}
```

- 2.

È possibile visualizzare il segreto PAT una sola volta, quando si crea il PAT. Se hai smarrito un segreto PAT o temi che non sia archiviato in modo sicuro, puoi crearne un altro.

Puoi visualizzare il file PATs associato al tuo account utente utilizzando il. AWS CLIÈ possibile visualizzare solo le informazioni sul PAT e non il valore del segreto PAT stesso.

 Note

Assicurati di utilizzare una versione recente di con cui AWS CLI lavorare. CodeCatalyst Le versioni precedenti potrebbero non contenere i CodeCatalyst comandi. È necessario configurare il AWS CLI profilo prima di poterlo utilizzare con CodeCatalyst. Per ulteriori informazioni, consulta [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#).

Visualizzazione PATs

È possibile visualizzare PATs in CodeCatalyst. L'elenco mostra tutto PATs ciò che hai associato alla tua identità utente. Il tuo PAT è associato al tuo profilo utente in tutti gli spazi e i progetti di CodeCatalyst. PATs Le scadute non vengono visualizzate perché vengono eliminate dopo la scadenza.

Visualizzazione PATs (console)

È possibile utilizzare la console per la visualizzazione PATs associata alla propria identità utente in CodeCatalyst.

Per visualizzare i token di accesso personali (console)

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

 Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. In Token di accesso personali, visualizza i nomi e le date di scadenza dei tuoi attuali PATs.

Visualizzazione PATs (CLI)

Puoi utilizzare la CLI per visualizzare i dati PATs associati alla tua identità utente in. CodeCatalyst

Per visualizzare i token di accesso personali (PATs) AWS CLI

- Nel terminale o nella riga di comando, esegui il `list-access-tokens` comando come segue.

```
aws codecatalyst list-access-tokens
```

In caso di successo, il comando restituisce informazioni sull'account utente PATs associato, come nell'esempio seguente.

```
{
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",
      "name": "marymajor-2222EXAMPLE",
      "expiresTime": "2024-02-04T01:56:04.402000+00:00"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
      "name": "marymajor-1111EXAMPLE",
      "expiresTime": "2023-03-12T01:58:40.694000+00:00"
    }
  ]
}
```

Eliminazione PATs

Puoi eliminare i dati PATs associati alla tua identità utente in CodeCatalyst.

Eliminazione PATs (console)

È possibile utilizzare la console per eliminare PATs . CodeCatalyst

Per eliminare un token di accesso personale (console)

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. In Token di accesso personali, scegli il selettore accanto al PAT che desideri eliminare, quindi scegli Elimina.

<name>Sul PAT Elimina: ? pagina, per confermare l'eliminazione, digita delete nel campo di testo. Scegli Elimina.

Eliminazione PATs (CLI)

È possibile eliminare un PAT associato alla propria identità utente utilizzando il. AWS CLI A tale scopo, è necessario fornire l'ID del PAT, che è possibile visualizzare utilizzando il delete-access-token comando.

Note

Assicurati di utilizzare una versione recente di con CodeCatalyst cui AWS CLI lavorare. Le versioni precedenti potrebbero non contenere i CodeCatalyst comandi. Per ulteriori informazioni sull'utilizzo di AWS CLI with CodeCatalyst, vedere [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#).

Per eliminare un token di accesso personale (AWS CLI)

- Nel terminale o nella riga di comando, esegui il delete-access-token comando, fornendo l'ID del PAT che desideri eliminare. Ad esempio, esegui il comando seguente per eliminare un PAT con un ID di. **123EXAMPLE**

```
aws codecatalyst delete-access-token --id a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb
```

In caso di successo, questo comando non restituisce alcuna risposta.

Accesso alle GitHub risorse con connessioni personali

Puoi utilizzare connessioni personali per autorizzare e connettere le tue GitHub risorse di terze parti. CodeCatalyst Ad esempio, utilizza una connessione personale per autorizzare l'accesso CodeCatalyst al tuo GitHub account e creare un repository come origine per il tuo progetto o blueprint. La connessione è mappata sulla tua CodeCatalyst identità e può essere utilizzata per connetterti a uno o più repository di origine. Le connessioni che crei sono associate alla tua identità utente in tutti gli spazi e i progetti di. CodeCatalyst

Note

Puoi gestire le connessioni personali con i blueprint nelle GitHub organizzazioni a cui hai accesso per farlo.

È possibile creare una connessione personale per un'identità utente (CodeCatalyst alias) in tutti gli spazi, per tipo di provider.

Puoi utilizzare le tue connessioni personali per CodeCatalyst creare un GitHub repository per un progetto, scegliere un repository di GitHub origine per un blueprint e gestire le pull request per il tuo repository. CodeCatalyst GitHub

Note

L'uso di connessioni personali per associare i blueprint a un GitHub repository non è lo stesso dell'uso delle estensioni per collegare un repository. CodeCatalyst GitHub Per ulteriori informazioni sulle estensioni consulta [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#).

Creazione di connessioni personali

È possibile utilizzare la console per creare una connessione personale associata alla propria identità utente in CodeCatalyst.

Per creare una connessione personale

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

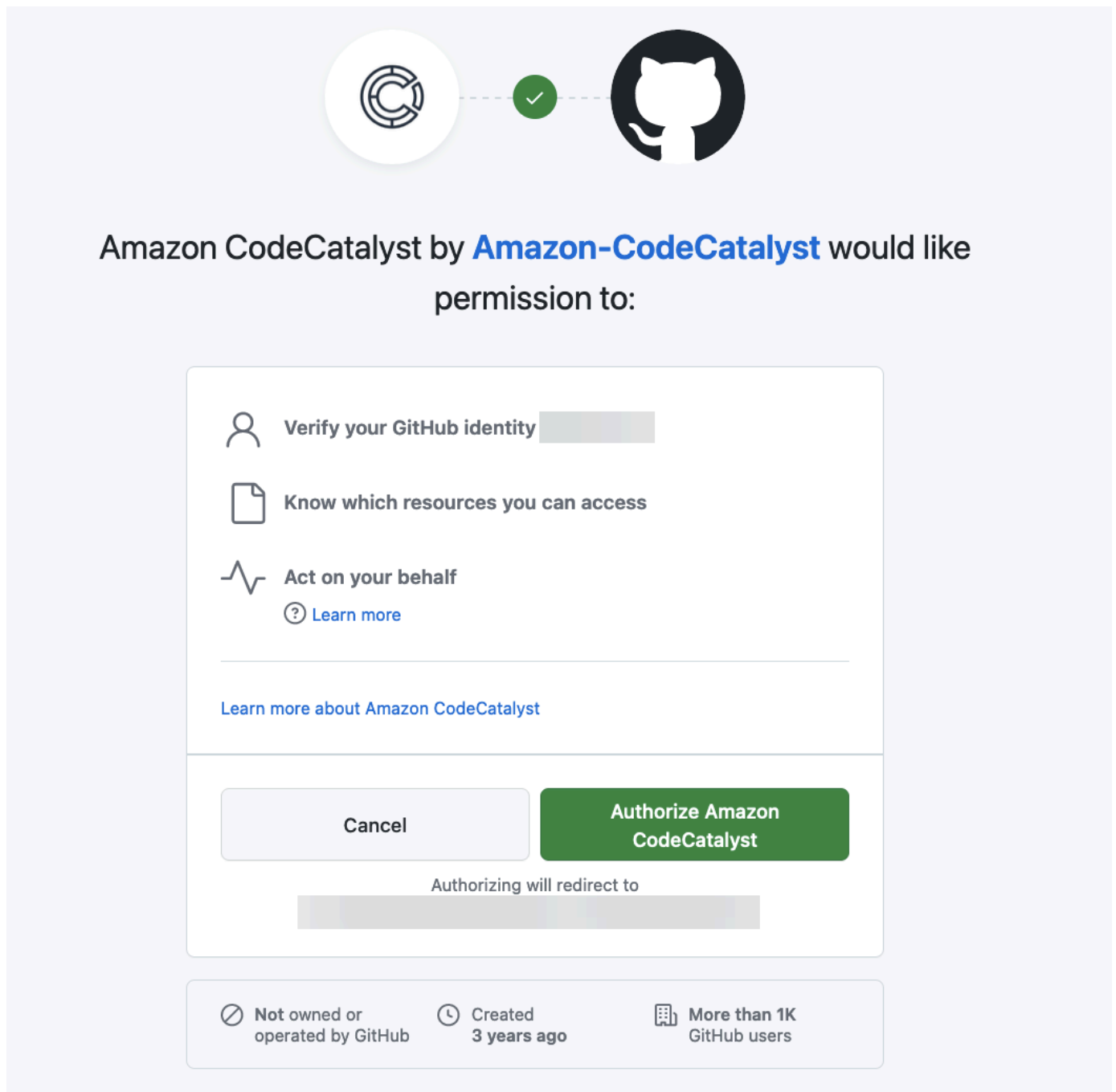
 Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. In Connessioni personali, scegli Crea.

Viene visualizzata la pagina Crea connessione.

4. Scegli Create (Crea). Viene visualizzata la pagina Crea connessione.
5. Nella pagina Crea connessione, in Provider, scegli GitHub. In Nome connessione, digita un nome per la connessione. Scegli Create (Crea).
6. Se richiesto, accedi al tuo GitHub account.
7. Nella pagina di conferma della connessione, scegli Accetta.
8. Nella pagina di conferma dell'installazione, scegli il pulsante di autorizzazione per confermare che desideri installare l'applicazione del connettore.



Eliminazione delle connessioni personali

È possibile eliminare una connessione personale associata alla propria identità utente in CodeCatalyst.

 Note

L'eliminazione della connessione personale in CodeCatalyst non disinstalla l'applicazione dal tuo GitHub account. Se si crea una nuova connessione personale, è possibile utilizzare l'installazione dell'app. Per disinstallare l'applicazione GitHub, è possibile revocarla e reinstallarla in un secondo momento.

Per eliminare una connessione personale in CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Nella barra dei menu in alto, scegli il badge del tuo profilo, quindi scegli Le mie impostazioni. Si apre la pagina Le CodeCatalyst mie impostazioni.

 Tip

Puoi anche trovare il tuo profilo utente accedendo alla pagina dei membri di un progetto o di uno spazio e scegliendo il tuo nome dall'elenco dei membri.

3. In Connessioni personali, scegli il selettore accanto alla connessione che desideri eliminare, quindi scegli Elimina.

<name>Sulla connessione Elimina:? pagina, per confermare l'eliminazione, digita delete nel campo di testo. Scegli Elimina.

1. Accedi GitHub e vai alle impostazioni del tuo account per le app installate. Scegli l'icona del tuo profilo, scegli Impostazioni, quindi scegli Applicazioni.
2. Nella scheda GitHub App autorizzate, nell'elenco delle applicazioni autorizzate, visualizza l'app installata per CodeCatalyst. Per revocare l'installazione, scegli Revoca.

Configura il tuo AWS Builder ID per accedere con l'autenticazione a più fattori (MFA)

Sia che tu abbia creato il tuo profilo AWS Builder ID per uso personale o professionale, ti consigliamo di configurare l'autenticazione a più fattori (MFA) come ulteriore livello di sicurezza. Ti consigliamo in particolare di configurare MFA se sei membro di uno spazio e collabori con altri su progetti. Poiché

più di una persona può avere accesso a un progetto, esistono maggiori opportunità di violazioni della sicurezza.

Quando abiliti la MFA, devi accedere ad Amazon CodeCatalyst con il tuo indirizzo e-mail e la tua password. Questa parte dell'accesso è il primo fattore, in cui usi qualcosa che conosci. Quindi accedi con un codice o una chiave di sicurezza. Questo è il secondo fattore, che è qualcosa che hai. Il secondo fattore potrebbe essere un codice di autenticazione generato dal dispositivo mobile o toccando o premendo una chiave di sicurezza collegata al computer. Nel loro insieme, questi molteplici fattori forniscono una maggiore sicurezza impedendo l'accesso non autorizzato.

Come registrare un dispositivo per l'utilizzo con l'autenticazione a più fattori

Utilizza la seguente procedura su Il mio profilo > Autenticazione a più fattori per registrare il nuovo dispositivo per l'autenticazione a più fattori (MFA).

Note

Ti consigliamo di scaricare l'app di autenticazione appropriata sul tuo dispositivo prima di iniziare i passaggi di questa procedura. Per un elenco di app che puoi utilizzare per i dispositivi MFA, consulta. [Applicazioni di autenticazione](#)

Per registrare il dispositivo per l'utilizzo con MFA

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. In alto a destra, scegli la freccia accanto all'icona con la tua prima iniziale, quindi scegli Profilo utente. Si apre la pagina del CodeCatalyst profilo.
3. Nella pagina del profilo, scegli Gestisci profilo e sicurezza. Si apre la pagina del profilo AWS Builder ID.
4. Sul lato sinistro della pagina, scegli Sicurezza.
5. Nella pagina Autenticazione a più fattori, scegli Registra dispositivo.
6. Nella pagina Registra dispositivo MFA, scegli uno dei seguenti tipi di dispositivi MFA e segui le istruzioni:
 - Chiave di sicurezza o autenticatore integrato
 1. Nella pagina Registra la chiave di sicurezza dell'utente, segui le istruzioni fornite dal browser o dalla piattaforma.

 Note

Questa esperienza varia in base al sistema operativo e al browser, quindi segui le istruzioni visualizzate dal browser o dalla piattaforma. Una volta completata la registrazione del dispositivo, avrai la possibilità di associare un nome utente descrittivo al dispositivo appena registrato. Se desideri modificare questa impostazione, scegli Rinomina, inserisci il nuovo nome, quindi scegli Salva.

- App Authenticator

1. Nella pagina Configura l'app di autenticazione, potresti notare le informazioni di configurazione per il nuovo dispositivo MFA, inclusa una grafica con codice QR. L'immagine è una rappresentazione della chiave segreta disponibile per l'immissione manuale sui dispositivi che non supportano i codici QR.
2. Utilizzando il dispositivo MFA fisico, procedi come segue:
 - a. Apri un'app di autenticazione MFA compatibile. Per un elenco delle app testate che puoi utilizzare con i dispositivi MFA, consulta [App di autenticazione testate](#). Se l'app MFA supporta più dispositivi, scegli l'opzione per creare un nuovo dispositivo MFA.
 - b. Determina se l'app MFA supporta i codici QR, quindi esegui una delle seguenti operazioni nella pagina Configura l'app di autenticazione:
 - i. Scegli Show QR code (Mostra codice QR) e utilizza l'app per eseguire la scansione del codice QR. Ad esempio, è possibile selezionare l'icona della fotocamera o un'opzione simile a Scan code (Scannerizza codice). Quindi usa la fotocamera del dispositivo per la scansione del codice.
 - ii. Scegli Mostra chiave segreta, quindi inserisci quella chiave segreta nella tua app MFA.

 Important

Quando configuri un dispositivo MFA per AWS Builder ID, salva una copia del codice QR o della chiave segreta in un posto sicuro. Questo può essere utile se perdi il telefono o devi reinstallare l'app MFA Authenticator. Se si verifica una di queste situazioni, puoi riconfigurare rapidamente l'app per utilizzare la stessa configurazione MFA.

3. Nella pagina Configura l'app di autenticazione, in Codice di autenticazione, inserisci la password monouso attualmente visualizzata sul dispositivo MFA fisico.

 Important

Invia la richiesta immediatamente dopo la generazione del codice. Se generi il codice e poi attendi troppo a lungo per inviare la richiesta, il dispositivo MFA viene associato correttamente al tuo profilo AWS Builder ID, ma il dispositivo MFA non è sincronizzato. Ciò accade perché le password monouso temporanee (TOTP) scadono dopo un breve periodo di tempo. Se ciò accade, è possibile sincronizzare nuovamente il dispositivo.

4. Scegliere Assign MFA (Assegna MFA). Il dispositivo MFA può ora iniziare a generare password monouso ed è ora pronto per l'uso.

Applicazioni di autenticazione

Le app di autenticazione sono autenticatori di terze parti basati su password monouso (OTP). Gli utenti possono utilizzare un'applicazione di autenticazione installata sul proprio dispositivo mobile o tablet come dispositivo MFA autorizzato. L'applicazione di autenticazione di terze parti deve essere conforme alla RFC 6238, un algoritmo TOTP (time-based one-time password) basato su standard in grado di generare codici di autenticazione a sei cifre.

Quando viene richiesta l'autenticazione a più fattori, gli utenti devono inserire un codice valido dall'app di autenticazione nella casella di immissione visualizzata. Ogni dispositivo MFA assegnato a un utente deve essere univoco. È possibile registrare due app di autenticazione per ogni utente.

App di autenticazione testate

Sebbene qualsiasi applicazione conforme a TOTP funzioni con IAM Identity Center MFA, la tabella seguente elenca le app di autenticazione di terze parti più note tra cui scegliere.

Sistema operativo	Applicazione di autenticazione testata
Android	Authy , Duo Mobile , Autenticatore , LastPass Autenticatore Microsoft , Autenticatore Google

Sistema operativo	Applicazione di autenticazione testata
iOS	Authy , Duo Mobile , Autenticatore , LastPass Autenticatore Microsoft , Autenticatore Google

Modifica dei dispositivi MFA

Dopo aver registrato un dispositivo MFA, puoi cambiarne il nome o eliminarlo. Ti consigliamo di avere sempre almeno un dispositivo MFA abilitato per un ulteriore livello di sicurezza. È possibile registrare fino a cinque dispositivi. Per scoprire come aggiungerne altri, consulta [Come registrare un dispositivo per l'utilizzo con l'autenticazione a più fattori](#).

Ridenominazione di un dispositivo MFA

Ridenominazione del dispositivo MFA

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. In alto a destra, scegli la freccia accanto all'icona con la tua prima iniziale, quindi scegli Profilo utente. Si apre la pagina del CodeCatalyst profilo.
3. Nella pagina del profilo, scegli Gestisci profilo e sicurezza. Si apre la pagina del profilo AWS Builder ID.
4. Scegli Autenticazione a più fattori sul lato sinistro della pagina. Vedrai che Rinomina è disattivato quando arrivi alla pagina.
5. Seleziona il dispositivo MFA che desideri modificare. Scegliere Rinomina. Quindi viene visualizzata una modalità.
6. Nella richiesta che si apre, inserisci il nuovo nome in Nome dispositivo MFA, quindi scegli Rinomina. Il dispositivo rinominato viene visualizzato in Dispositivi di autenticazione a più fattori (MFA).

Eliminazione di un dispositivo MFA

Eliminazione di un dispositivo MFA

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. In alto a destra, scegli la freccia accanto all'icona con la tua prima iniziale, quindi scegli Profilo utente. Si apre la pagina del CodeCatalyst profilo.

3. Nella pagina del profilo, scegli Gestisci profilo e sicurezza. Si apre la pagina del profilo AWS Builder ID.
4. Scegli Autenticazione a più fattori sul lato sinistro della pagina. Vedrai che Elimina è disattivato quando arrivi alla pagina.
5. Seleziona il dispositivo MFA che desideri modificare. Scegli Elimina. Viene visualizzato un modale che dice Elimina dispositivo MFA? . Segui le istruzioni per eliminare il dispositivo.
6. Scegli Elimina. Il dispositivo eliminato non viene più visualizzato in Dispositivi di autenticazione a più fattori (MFA).

Sicurezza in Amazon CodeCatalyst

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti degli spazi più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#) descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per maggiori informazioni sui programmi di conformità applicabili CodeCatalyst, consulta la sezione [AWS Servizi rientranti nell'ambito del programma di conformitàAWS](#) .
- Sicurezza nel cloud: la tua responsabilità è determinata dai AWS servizi che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i tuoi requisiti aziendali e le leggi e le normative applicabili

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amazon CodeCatalyst. Ti mostra come configurare per CodeCatalyst soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere CodeCatalyst le tue risorse.

Indice

- [Privacy dei dati in Amazon CodeCatalyst](#)
- [Protezione dei dati in Amazon CodeCatalyst](#)

- [Identity and Access Management e Amazon CodeCatalyst](#)
- [Convalida della conformità per Amazon CodeCatalyst](#)
- [Resilienza in Amazon CodeCatalyst](#)
- [Sicurezza dell'infrastruttura in Amazon CodeCatalyst](#)
- [Analisi della configurazione e delle vulnerabilità in Amazon CodeCatalyst](#)
- [I tuoi dati e la tua privacy in Amazon CodeCatalyst](#)
- [Le migliori pratiche per le azioni del flusso di lavoro in Amazon CodeCatalyst](#)
- [Comprendere il modello CodeCatalyst di fiducia](#)

Privacy dei dati in Amazon CodeCatalyst

CodeCatalyst raccoglie informazioni aggregate in IAM Identity Center. CodeCatalyst utilizza questi dati aggregati per fornire l'accesso a spazi e progetti in. CodeCatalyst Vengono raccolte le seguenti informazioni aggregate:

- Nome completo dell'utente
- Email dell'utente
- Nome utente dell'utente

CodeCatalyst i dati vengono crittografati automaticamente quando sono inattivi. Non è richiesta alcuna azione da parte del cliente. Quando un utente, un centro di identità o uno spazio viene eliminato, CodeCatalyst elimina i dati del cliente entro 24 ore.

CodeCatalyst utilizza chiavi AWS di proprietà AWS KMS . CodeCatalyst non supporta la crittografia a riposo utilizzando chiavi KMS gestite dal cliente per gli attributi di identità recuperati da IAM Identity Center.

[Per ulteriori informazioni sulle AWS KMS chiavi, consulta la documentazione per l'utente all'indirizzo AWS KMS keys.](#)

Protezione dei dati in Amazon CodeCatalyst

Sicurezza e conformità sono una responsabilità condivisa tra Amazon CodeCatalyst e il cliente, così come il [modello di responsabilità AWS condivisa](#) si applica all'uso delle AWS risorse utilizzate in un flusso di lavoro. Come descritto in questo modello, CodeCatalyst è responsabile della protezione dell'infrastruttura globale del servizio. L'utente è responsabile del controllo dei contenuti ospitati su

questa infrastruttura. Questo modello di responsabilità condivisa si applica alla protezione dei dati in CodeCatalyst.

Ai fini della protezione dei dati, ti consigliamo di proteggere le credenziali del tuo account e di configurare l'autenticazione a più fattori al momento dell'accesso. Per ulteriori informazioni, consulta [Configura il tuo AWS Builder ID per accedere con l'autenticazione a più fattori \(MFA\)](#).

Non inserire informazioni riservate o sensibili, come gli indirizzi e-mail dei clienti, nei tag o nei campi in formato libero come il campo Nome. Ciò include i nomi delle risorse e qualsiasi altro identificatore inserito oltre a quelli connessi. CodeCatalyst Account AWS Ad esempio, non inserire informazioni riservate o sensibili come parte dei nomi di spazi, progetti o flotte di distribuzione. Tutti i dati immessi nei tag, nei nomi o nei campi in formato libero utilizzati per i nomi potrebbero essere utilizzati per i registri di fatturazione o diagnostica o potrebbero essere inclusi nei percorsi URL. Questo vale per l'utilizzo della console, dell'API AWS CLI, dell' CodeCatalyst Action Development Kit o altro. AWS SDKs

Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere alcuna informazione sulle credenziali di sicurezza nell'URL per convalidare la tua richiesta a quel server.

CodeCatalyst gli archivi di origine vengono crittografati automaticamente quando sono inattivi. Non è richiesta alcuna azione da parte del cliente. CodeCatalyst crittografa anche i dati del repository in transito utilizzando il protocollo HTTPS.

CodeCatalyst non supporta la crittografia a riposo utilizzando chiavi KMS gestite dal cliente per gli attributi di identità recuperati da IAM Identity Center.

CodeCatalyst supporta MFA. Per ulteriori informazioni, consulta [Configura il tuo AWS Builder ID per accedere con l'autenticazione a più fattori \(MFA\)](#).

Crittografia dei dati

CodeCatalyst archivia e trasferisce in modo sicuro i dati all'interno del servizio. Tutti i dati sono crittografati in transito e a riposo. Tutti i dati creati o archiviati dal servizio, inclusi i metadati del servizio, vengono archiviati nativamente nel servizio e crittografati.

Note

Sebbene le informazioni sui problemi vengano archiviate in modo sicuro all'interno del servizio, le informazioni sui problemi aperti vengono archiviate anche nella cache locale del browser in cui sono stati visualizzati i problemi, i backlog e i singoli problemi. Per una

sicurezza ottimale, assicurati di cancellare la cache del browser per rimuovere queste informazioni.

Se utilizzi risorse collegate a CodeCatalyst, ad esempio la connessione di un account a un repository collegato Account AWS o a un repository collegato in GitHub, i dati in transito CodeCatalyst da quella risorsa collegata vengono crittografati, ma la gestione dei dati in quella risorsa collegata è gestita dal servizio collegato. Per ulteriori informazioni, consulta la documentazione relativa al servizio collegato e [Le migliori pratiche per le azioni del flusso di lavoro in Amazon CodeCatalyst](#).

Gestione delle chiavi

CodeCatalyst utilizza chiavi KMS AWS di proprietà dell'utente. Ciò significa che le chiavi sono gestite da AWS e il cliente non deve crearle o gestirle direttamente. Ciò semplifica la gestione delle chiavi per gli scenari di crittografia più comuni all'interno AWS.

Riservatezza del traffico inter-rete

Quando crei uno spazio in CodeCatalyst, scegli Regione AWS dove verranno archiviati i dati e le risorse per quello spazio. I dati e i metadati del progetto non lo abbandonano Regione AWS mai. [Tuttavia, per supportare la navigazione all'interno CodeCatalyst, un insieme limitato di spazio, progetto e metadati utente viene replicato Regioni AWS in tutta la partizione.](#) Non verrà replicato Regioni AWS all'esterno di quella partizione. Ad esempio, se scegli Stati Uniti occidentali (Oregon) come Regione AWS quando crei il tuo spazio, i tuoi dati non verranno replicati nelle regioni delle regioni della Cina o. AWS GovCloud (US) Per ulteriori informazioni, consulta [Gestione Regioni AWS](#), [infrastruttura AWS globale](#) e endpoint di [AWS servizio](#).

I dati replicati Regioni AWS all'interno di una partizione includono:

- Un valore hash crittografato che rappresenta il nome dello spazio per garantire l'unicità dei nomi degli spazi. Questo valore non è leggibile dall'uomo e non mostra i nomi effettivi degli spazi
- L'ID univoco dello spazio
- Metadati per lo spazio che facilita la navigazione tra gli spazi
- La posizione Regione AWS in cui si trova lo spazio
- L'unicità IDs di tutti i progetti nello spazio
- L'ID del ruolo che indica il ruolo di un utente in uno spazio o in un progetto
- Al momento dell'iscrizione CodeCatalyst, dati e metadati relativi alla procedura di registrazione, tra cui:

- L'ID univoco di ID Builder AWS
- Il nome visualizzato per l'utente nella loro ID Builder AWS
- L'alias dell'utente nel loro ID Builder AWS
- L'indirizzo e-mail utilizzato quando l'utente si è registrato al ID Builder AWS
- Lo stato di avanzamento della procedura di registrazione
- Se si crea uno spazio come parte della procedura di registrazione, l' Account AWS ID utilizzato come account di fatturazione per lo spazio

I nomi degli spazi sono univoci in tutto CodeCatalyst. Assicurati di non includere dati sensibili nel nome dello spazio.

Quando lavori con risorse collegate e account connessi, ad esempio una connessione a un Account AWS o a un GitHub repository, ti consigliamo di configurare le posizioni di origine e di destinazione con il massimo livello di sicurezza supportato da ciascuna di esse. CodeCatalyst protegge la connessione tra Account AWS e le zone di disponibilità utilizzando Transport Layer Security (TLS)

1.2. Regioni AWS

Identity and Access Management e Amazon CodeCatalyst

In Amazon CodeCatalyst, crei e utilizzi un AWS Builder ID per accedere ai tuoi spazi e ai tuoi progetti. Un AWS Builder ID non è un'identità in AWS Identity and Access Management (IAM) e non esiste in un Account AWS. Tuttavia, CodeCatalyst si integra con IAM durante la verifica di uno spazio a fini di fatturazione e quando è connesso a un Account AWS per creare e utilizzare le relative risorse.

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. Gli amministratori IAM controllano chi può essere autenticato (chi ha effettuato l'accesso) e autorizzato (chi dispone di autorizzazioni) a utilizzare le risorse. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Quando crei uno spazio in Amazon CodeCatalyst, devi collegarne un Account AWS come account di fatturazione per il tuo spazio. Per verificare lo spazio CodeCatalyst devi disporre delle autorizzazioni Account AWS di amministratore o disporre dell'autorizzazione. Hai anche la possibilità di aggiungere un ruolo IAM per il tuo spazio da CodeCatalyst utilizzare per creare e accedere alle risorse in quello connesso Account AWS. Questo è chiamato [ruolo di servizio](#). Puoi scegliere di

creare connessioni a più di un account Account AWS e creare ruoli di servizio per CodeCatalyst ciascuno di questi account.

Note

La fatturazione CodeCatalyst avviene nell'account Account AWS designato come fatturazione. Tuttavia, se crei un ruolo di CodeCatalyst servizio in questo ruolo Account AWS o in qualsiasi altro ruolo connesso Account AWS, le risorse create e utilizzate dal ruolo di CodeCatalyst servizio verranno fatturate in quel ruolo connesso. Account AWS Per ulteriori informazioni, consulta [Managing billing](#) nella Amazon CodeCatalyst Administrator Guide.

Argomenti

- [Politiche basate sull'identità in IAM](#)
- [Azioni politiche in IAM](#)
- [Risorse politiche in IAM](#)
- [Chiavi relative alle condizioni delle politiche in IAM](#)
- [Esempi di politiche basate sull'identità per le connessioni CodeCatalyst](#)
- [Utilizzo dei tag per controllare l'accesso alle risorse di connessione dell'account](#)
- [CodeCatalyst riferimento alle autorizzazioni](#)
- [Utilizzo di ruoli collegati ai servizi per CodeCatalyst](#)
- [AWS politiche gestite per Amazon CodeCatalyst](#)
- [Concedi l'accesso alle AWS risorse del progetto con i ruoli IAM](#)

Politiche basate sull'identità in IAM

Le politiche basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità. Tale identità potrebbe essere un utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica

all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Esempi di policy basate su identità per CodeCatalyst

Per visualizzare esempi di politiche CodeCatalyst basate sull'identità, vedere. [Esempi di politiche basate sull'identità per le connessioni CodeCatalyst](#)

Azioni politiche in IAM

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale preside può eseguire quali azioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Le azioni politiche di solito hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
    "prefix:action1",  
    "prefix:action2"  
]
```

Risorse politiche in IAM

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale preside può eseguire quali azioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"

```

Chiavi relative alle condizioni delle politiche in IAM

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale preside può eseguire quali azioni su quali risorse e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione OR logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione globali di AWS, consulta [Chiavi di contesto delle condizioni globali di AWS](#) nella Guida per l'utente IAM.

Esempi di politiche basate sull'identità per le connessioni CodeCatalyst

Nel CodeCatalyst, Account AWS sono necessari per gestire la fatturazione di uno spazio e accedere alle risorse nei flussi di lavoro del progetto. Una connessione all'account viene utilizzata per autorizzare l'aggiunta Account AWS a uno spazio. Le politiche basate sull'identità vengono utilizzate nelle connessioni. Account AWS

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse CodeCatalyst. Inoltre, non possono eseguire attività utilizzando Console di gestione AWS, AWS Command Line Interface (AWS CLI) o l'API. AWS Un amministratore IAM deve creare policy IAM che concedano a utenti e ruoli l'autorizzazione per eseguire operazioni sulle risorse di cui hanno bisogno. L'amministratore deve quindi collegare queste policy agli utenti che ne hanno bisogno.

Il seguente esempio di policy IAM concede le autorizzazioni per le azioni relative alle connessioni degli account. Usali per limitare l'accesso a cui connettere gli CodeCatalyst account.

Esempio 1: consenti a un utente di accettare richieste di connessione in un'unica soluzione Regione AWS

La seguente politica di autorizzazione consente solo agli utenti di visualizzare e accettare richieste di connessione tra CodeCatalyst e Account AWS. Inoltre, la politica utilizza una condizione per consentire solo le azioni nella regione us-west-2 e non da altre. Regioni AWS Per visualizzare e approvare la richiesta, l'utente accede Console di gestione AWS con lo stesso account specificato nella richiesta.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:GetPendingConnection"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    }
  ]
}
```

Esempio 2: consentire la gestione delle connessioni nella console per una singola unità Regione AWS

La seguente politica di autorizzazione consente agli utenti di gestire le connessioni tra CodeCatalyst e Account AWS in una singola regione. La politica utilizza una condizione per consentire solo le azioni nella regione us-west-2 e non da altre. Regioni AWS Dopo aver creato una connessione, è possibile creare il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo scegliendo l'opzione in.

Console di gestione AWS Nella politica di esempio, la condizione per l'iam:PassRoleazione include i principali del servizio per CodeCatalyst. Solo i ruoli con tale accesso verranno creati in. Console di gestione AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "codecatalyst.amazonaws.com",
            "codecatalyst-runner.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

```

    ]
  }
}

```

Esempio 3: Negare la gestione delle connessioni

La seguente politica di autorizzazione nega agli utenti la possibilità di gestire le connessioni tra e. CodeCatalyst Account AWS

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*"
    }
  ]
}

```

Utilizzo dei tag per controllare l'accesso alle risorse di connessione dell'account

I tag possono essere allegati alla risorsa o passati nella richiesta ai servizi che supportano il tagging. Le risorse nelle politiche possono avere tag e alcune azioni nelle politiche possono includere tag. Le chiavi di condizione per l'etichettatura includono le chiavi di `aws:ResourceTag` condizione `aws:RequestTag` e. Quando si crea una policy IAM, è possibile utilizzare le chiavi di condizione di tag per controllare:

- Quali utenti possono eseguire azioni su una risorsa di connessione, in base ai tag già presenti.
- Quali tag possono essere passati in una richiesta di operazione.
- Se delle chiavi di tag specifiche possono essere utilizzate in una richiesta.

Gli esempi seguenti mostrano come specificare le condizioni dei tag nelle politiche per gli utenti che utilizzano le connessioni degli CodeCatalyst account. Per ulteriori informazioni su queste chiavi di condizione, consulta [Chiavi relative alle condizioni delle politiche in IAM](#).

Esempio 1: consenti azioni basate sui tag nella richiesta

La seguente politica concede agli utenti l'autorizzazione ad approvare le connessioni all'account.

A questo scopo, consente le operazioni `AcceptConnection` `TagResource` se la richiesta specifica un tag denominato `Project` con il valore `ProjectA`. La chiave di condizione `aws:RequestTag` viene utilizzata per controllare quali tag possono essere passati in una richiesta IAM. La condizione `aws:TagKeys` garantisce che la chiave tag rileva la distinzione tra maiuscole e minuscole.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

Esempio 2: consenti azioni basate sui tag delle risorse

La seguente politica concede agli utenti il permesso di eseguire azioni sulle risorse di connessione all'account e ottenere informazioni sulle stesse.

A tal fine, consente azioni specifiche se la connessione ha un tag denominato `Project` con lo stesso valore `ProjectA`. La chiave di condizione `aws:ResourceTag` viene utilizzata per controllare quali tag possono essere passati in una richiesta IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:GetConnection",
        "codecatalyst:DeleteConnection",
        "codecatalyst:AssociateIamRoleToConnection",
        "codecatalyst:DisassociateIamRoleFromConnection",
        "codecatalyst:ListIamRolesForConnection",
        "codecatalyst:PutBillingAuthorization"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "ProjectA"
        }
      }
    }
  ]
}
```

CodeCatalyst riferimento alle autorizzazioni

Questa sezione fornisce un riferimento alle autorizzazioni per le azioni utilizzate con la risorsa di connessione dell'account a Account AWS cui si è connessi. CodeCatalyst La sezione seguente descrive le azioni basate solo sulle autorizzazioni correlate alla connessione degli account.

Autorizzazioni richieste per le connessioni agli account

Le seguenti autorizzazioni sono necessarie per utilizzare le connessioni degli account.

CodeCatalyst autorizzazioni per le connessioni degli account	Autorizzazioni richieste	Resources
AcceptConnection	Necessario per accettare una richiesta di connessione di questo account a uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	Supporta solo un carattere jolly (*) nell'elemento Resource della policy.
AssociatelaRoleToConnection	Necessario per associare un ruolo IAM a una connessione di account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
DeleteConnection	Necessario per eliminare una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
DisassociatelaRoleFromConnection	Necessario per dissociare un ruolo IAM da una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
GetBillingAuthorization	Necessario per descrivere l'autorizzazione di fatturazione	arn:aws:codecatalyst:region: <i>account_ID</i>

CodeCatalyst autorizzazioni per le connessioni degli account	Autorizzazioni richieste	Resources
	per una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	<code>arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i></code>
GetConnection	Necessario per stabilire una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	<code>arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i></code>
GetPendingConnection	Necessario per ricevere una richiesta in sospeso per connettere questo account a uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	Supporta solo un carattere jolly (*) nell'elemento Resource della policy.
ListConnections	Necessario per elencare le connessioni tra account che non sono in sospeso. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	Supporta solo un carattere jolly (*) nell'elemento Resource della policy.
ListIamRolesForConnection	Necessario per elencare i ruoli IAM associati a una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	<code>arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i></code>

CodeCatalyst autorizzazioni per le connessioni degli account	Autorizzazioni richieste	Resources
ListTagsForResource	Necessario per elencare i tag associati a una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
PutBillingAuthorization	Necessario per creare o aggiornare l'autorizzazione di fatturazione per la connessione di un account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>
RejectConnection	Necessario per rifiutare una richiesta di connessione di questo account a uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	Supporta solo un carattere jolly (*) nell'elemento Resource della policy.
TagResource	Necessario per creare o modificare i tag associati alla connessione di un account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>

CodeCatalyst autorizzazioni per le connessioni degli account	Autorizzazioni richieste	Resources
UntagResource	Necessario per rimuovere i tag associati a una connessione all'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>

Autorizzazioni richieste per le applicazioni IAM Identity Center

Le seguenti autorizzazioni sono necessarie per lavorare con le applicazioni IAM Identity Center.

CodeCatalyst autorizzazioni per le applicazioni IAM Identity Center	Autorizzazioni richieste	Resources
AssociateIdentityCenterApplicationToSpace	Necessario per associare un'applicazione IAM Identity Center a uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
AssociateIdentityToIdentityCenterApplication	Necessario per associare un'identità a un'applicazione IAM Identity Center per uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
BatchAssociateIdentitiesToIdentityCenterApplication	Necessario per associare più identità a un'applicazione	arn:aws:codecatalyst:region: <i>account_ID</i>

CodeCatalyst autorizzazioni per le applicazioni IAM Identity Center	Autorizzazioni richieste	Resources
	IAM Identity Center per uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	<i>D</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
BatchDisassociateIdentitiesFromIdentityCenterApplication	Necessario per dissociare più identità da un'applicazione IAM Identity Center per uno spazio. CodeCatalyst Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
CreateIdentityCenterApplication	Necessario per creare un'applicazione IAM Identity Center. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
CreateSpaceAdminRoleAssignment	Necessario per creare un'assegnazione di ruolo di amministratore per un determinato CodeCatalyst spazio e un'applicazione IAM Identity Center. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

CodeCatalyst autorizzazioni per le applicazioni IAM Identity Center	Autorizzazioni richieste	Resources
DeletIdentityCenterApplication	Necessario per eliminare un'applicazione IAM Identity Center. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
DisassociateIdentityCenterApplicationFromSpace	Necessario per dissociare un'applicazione IAM Identity Center da uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
DisassociateIdentityFromIdentityCenterApplication	Necessario per dissociare un'identità da un'applicazione IAM Identity Center per uno CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
GetIdentityCenterApplication	Necessario per ottenere informazioni su un'applicazione IAM Identity Center. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

CodeCatalyst autorizzazioni per le applicazioni IAM Identity Center	Autorizzazioni richieste	Resources
ListIdentityCenterApplications	Necessario per visualizzare un elenco di tutte le applicazioni IAM Identity Center presenti nell'account. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	Supporta solo un carattere jolly (*) nell'elemento Resource della policy.
ListIdentityCenterApplicationsForSpace	Necessario per visualizzare un elenco di applicazioni IAM Identity Center per CodeCatalyst spazio. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
ListSpacesForIdentityCenterApplication	Necessario per visualizzare un elenco di CodeCatalyst spazi in base all'applicazione IAM Identity Center. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>
SynchronizeIdentityCenterApplication	Necessario per sincronizzare un'applicazione IAM Identity Center con l'archivio di identità di supporto. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

CodeCatalyst autorizzazioni per le applicazioni IAM Identity Center	Autorizzazioni richieste	Resources
UpdateIdentityCenterApplication	Necessario per aggiornare un'applicazione IAM Identity Center. Si tratta solo di un'autorizzazione di policy IAM, non di un'operazione API.	arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i>

Utilizzo di ruoli collegati ai servizi per CodeCatalyst

Amazon CodeCatalyst utilizza ruoli [collegati ai servizi AWS Identity and Access Management](#) (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. CodeCatalyst I ruoli collegati ai servizi sono predefiniti CodeCatalyst e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione CodeCatalyst perché non è necessario aggiungere manualmente le autorizzazioni necessarie. CodeCatalyst definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. CodeCatalyst Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. In questo modo proteggi CodeCatalyst le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta [AWS i servizi che funzionano con IAM](#) e cerca i servizi con Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Autorizzazioni di ruolo collegate ai servizi per CodeCatalyst

CodeCatalyst utilizza il ruolo collegato al servizio denominato `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`— Consente ad

Amazon l'accesso in CodeCatalyst solo lettura ai profili delle istanze dell'applicazione e agli utenti e ai gruppi di directory associati per tuo conto.

Il `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` service-linked role si affida ai seguenti servizi per l'assunzione del ruolo:

- `codecatalyst.amazonaws.com`

La politica di autorizzazione del ruolo denominata

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy` consente di CodeCatalyst completare le seguenti azioni sulle risorse specificate:

- Azione: `View application instance profiles and associated directory users and groups` per CodeCatalyst spaces that support identity federation and SSO users and groups

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

Creazione di un ruolo collegato al servizio per CodeCatalyst

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei uno spazio nella Console di gestione AWS, o nell' AWS API AWS CLI, CodeCatalyst crea automaticamente il ruolo collegato al servizio.

Important

Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo. Inoltre, se utilizzavi il CodeCatalyst servizio prima del 17 novembre 2023, quando ha iniziato a supportare ruoli collegati al servizio, hai creato il `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` ruolo nel tuo account. Per saperne di più, vedi [A new role appeared in my Account AWS](#).

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei uno spazio, CodeCatalyst crea nuovamente il ruolo collegato al servizio per te.

Puoi anche utilizzare la console IAM per creare un ruolo collegato ai servizi con lo use case View application instance profiles e gli utenti e i gruppi di directory associati. Nella AWS CLI o nell' AWS API, crea un ruolo collegato al servizio con il nome del servizio. `codecatalyst.amazonaws.com` Per ulteriori informazioni, consulta [Creazione di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM. Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

Modifica di un ruolo collegato al servizio per CodeCatalyst

CodeCatalyst non consente di modificare il `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` ruolo collegato al servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato al servizio per CodeCatalyst

Non è necessario eliminare manualmente il `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` ruolo. Quando elimini uno spazio nella Console di gestione AWS, o nell' AWS API AWS CLI, CodeCatalyst pulisce le risorse ed elimina automaticamente il ruolo collegato al servizio.

Puoi anche utilizzare la console IAM, the AWS CLI o l' AWS API per eliminare manualmente il ruolo collegato al servizio. Per farlo, sarà necessario prima eseguire manualmente la pulizia delle risorse associate al ruolo collegato ai servizi e poi eliminarlo manualmente.

Note

Se il CodeCatalyst servizio utilizza il ruolo quando tenti di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare CodeCatalyst le risorse utilizzate da `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`

- [Eliminare lo spazio.](#)

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` ruolo collegato al servizio. Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi CodeCatalyst

CodeCatalyst supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [AWS Regioni ed endpoint](#).

CodeCatalyst non supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Puoi utilizzare il plugin `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` ruolo nelle seguenti regioni.

Nome della Regione	Identità della regione	Support in CodeCatalyst
US East (N. Virginia)	us-east-1	No
Stati Uniti orientali (Ohio)	us-east-2	No
US West (N. California)	us-west-1	No
US West (Oregon)	us-west-2	Sì
Africa (Cape Town)	af-south-1	No
Asia Pacifico (Hong Kong)	ap-east-1	No
Asia Pacifico (Giacarta)	ap-southeast-3	No
Asia Pacific (Mumbai)	ap-south-1	No
Asia Pacifico (Osaka-Locale)	ap-northeast-3	No
Asia Pacifico (Seul)	ap-northeast-2	No
Asia Pacific (Singapore)	ap-southeast-1	No
Asia Pacific (Sydney)	ap-southeast-2	No

Nome della Regione	Identità della regione	Support in CodeCatalyst
Asia Pacifico (Tokyo)	ap-northeast-1	No
Canada (Central)	ca-central-1	No
Europe (Frankfurt)	eu-central-1	No
Europa (Irlanda)	eu-west-1	Sì
Europe (London)	eu-west-2	No
Europa (Milano)	eu-south-1	No
Europe (Paris)	eu-west-3	No
Europa (Stoccolma)	eu-north-1	No
Medio Oriente (Bahrein)	me-south-1	No
Medio Oriente (Emirati Arabi Uniti)	me-central-1	No
Sud America (São Paulo)	sa-east-1	No
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	No
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	No

AWS politiche gestite per Amazon CodeCatalyst

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Si consiglia pertanto di ridurre ulteriormente le autorizzazioni definendo [policy gestite dal cliente](#) specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

AWS politica gestita: AmazonCodeCatalystSupportAccess

Si tratta di una politica che concede a tutti gli amministratori e i membri dello spazio le autorizzazioni per utilizzare il piano di supporto premium Business o Enterprise associato all'account di fatturazione dello spazio. Queste autorizzazioni consentono agli amministratori dello spazio e ai membri di utilizzare il piano di supporto premium per le risorse a cui dispongono delle autorizzazioni nell'ambito delle politiche di autorizzazione. CodeCatalyst

Dettagli delle autorizzazioni

Questa policy include le seguenti autorizzazioni:

- **support**— Concede le autorizzazioni per consentire agli utenti di cercare, creare e risolvere casi di AWS Supporto. Concede inoltre le autorizzazioni per descrivere le comunicazioni, i livelli di gravità, gli allegati e i dettagli dei relativi casi di supporto.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
    "Action": [
      "support:DescribeAttachment",
      "support:DescribeCaseAttributes",
      "support:DescribeCases",
      "support:DescribeCommunications",
      "support:DescribeIssueTypes",
      "support:DescribeServices",
      "support:DescribeSeverityLevels",
      "support:DescribeSupportLevel",
      "support:SearchForCases",
      "support:AddAttachmentsToSet",
      "support:AddCommunicationToCase",
      "support:CreateCase",
      "support:InitiateCallForCase",
      "support:InitiateChatForCase",
      "support:PutCaseAttributes",
      "support:RateCaseCommunication",
      "support:ResolveCase"
    ],
    "Resource": "*"
  }
}
```

AWS politica gestita: AmazonCodeCatalystFullAccess

Si tratta di una politica che concede le autorizzazioni per gestire il tuo CodeCatalyst spazio e gli account connessi nella pagina Amazon CodeCatalyst Spaces del. Console di gestione AWS. Questa applicazione viene utilizzata per configurare Account AWS la connessione al tuo spazio in. CodeCatalyst

Dettagli delle autorizzazioni

Questa policy include le seguenti autorizzazioni:

- `codecatalyst`— Concede le autorizzazioni complete per la pagina Amazon CodeCatalyst Spaces nel. Console di gestione AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeCatalystResourceAccess",
      "Effect": "Allow",
      "Action": [
        "codecatalyst:*",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeCatalystAssociateIAMRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "codecatalyst.amazonaws.com",
            "codecatalyst-runner.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

AWS politica gestita: AmazonCodeCatalystReadOnlyAccess

Si tratta di una politica che concede le autorizzazioni per visualizzare ed elencare le informazioni per gli spazi e gli account connessi nella pagina Amazon CodeCatalyst Spaces del. Console di gestione AWS Questa applicazione viene utilizzata per configurare Account AWS la connessione al tuo spazio in. CodeCatalyst

Dettagli delle autorizzazioni

Questa policy include le seguenti autorizzazioni:

- `codecatalyst`— Concede autorizzazioni di sola lettura per la pagina Amazon CodeCatalyst Spaces in. Console di gestione AWS

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:Get*",
        "codecatalyst:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS politica gestita:

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy`

Non puoi

allegare `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy`; alle tue entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente di CodeCatalyst eseguire azioni per tuo conto. Per ulteriori informazioni, consulta [Utilizzo di ruoli collegati ai servizi per CodeCatalyst](#).

Questa policy consente ai clienti di visualizzare i profili delle istanze dell'applicazione e gli utenti e i gruppi delle directory associate durante la gestione degli spazi in. CodeCatalyst I clienti visualizzeranno queste risorse durante la gestione di spazi che supportano la federazione delle identità e gli utenti e i gruppi SSO.

Dettagli delle autorizzazioni

Questa policy include le seguenti autorizzazioni:

- **sso**— Concede le autorizzazioni per consentire agli utenti di visualizzare i profili delle istanze dell'applicazione gestiti in IAM Identity Center per gli spazi associati in. CodeCatalyst

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy",
      "Effect": "Allow",
      "Action": [
        "sso:ListInstances",
        "sso:ListApplications",
        "sso:ListApplicationAssignments",
        "sso:DescribeInstance",
        "sso:DescribeApplication"
      ],
      "Resource": "*"
    }
  ]
}
```

CodeCatalyst aggiornamenti alle politiche gestite AWS

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite CodeCatalyst da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della [cronologia dei CodeCatalyst documenti](#).

Modifica	Descrizione	Data
AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy : nuova policy	CodeCatalyst ha aggiunto la politica. Concede le autorizzazioni per consentire agli CodeCatalyst utenti di visualizzare i profili delle istanze dell'applicazione e gli utenti e i gruppi delle directory associate.	17 novembre 2023
AmazonCodeCatalystSupportAccess : nuova policy	CodeCatalyst ha aggiunto la politica. Concede le autorizzazioni per consentire agli CodeCatalyst utenti di cercare, creare e risolvere casi di supporto, nonché di visualizzare le comunicazioni e i dettagli correlati.	20 aprile 2023
AmazonCodeCatalystFullAccess : nuova policy	CodeCatalyst ha aggiunto la politica. Garantisce l'accesso completo a. CodeCatalyst	20 aprile 2023
AmazonCodeCatalystReadOnlyAccess : nuova policy	CodeCatalyst ha aggiunto la politica. Concede l'accesso in sola lettura a. CodeCatalyst	20 aprile 2023
CodeCatalyst ha iniziato a tenere traccia delle modifiche	CodeCatalyst ha iniziato a tenere traccia delle modifiche	20 aprile 2023

Modifica	Descrizione	Data
	per le sue politiche AWS gestite.	

Concedi l'accesso alle AWS risorse del progetto con i ruoli IAM

CodeCatalyst puoi accedere alle AWS risorse collegando il tuo Account AWS a uno CodeCatalyst spazio. È quindi possibile creare i seguenti ruoli di servizio e associarli quando si collega l'account.

Per ulteriori informazioni sugli elementi utilizzati in una policy JSON, consulta [IAM JSON Policy Elements Reference](#) nella IAM User Guide.

- Per accedere alle risorse nei tuoi CodeCatalyst progetti e flussi di lavoro, devi prima concedere l'autorizzazione CodeCatalyst ad accedere a tali risorse per tuo conto. Account AWS A tale scopo, è necessario creare un ruolo di servizio in un ambiente connesso Account AWS che CodeCatalyst possa assumere per conto degli utenti e dei progetti dello spazio. Puoi scegliere di creare e utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-**spaceName** servizio oppure puoi creare ruoli di servizio personalizzati e configurare queste politiche e ruoli IAM manualmente. Come best practice, assegna a questi ruoli il numero minimo di autorizzazioni necessarie.

Note

Per i ruoli di servizio personalizzati, è necessario il responsabile del CodeCatalyst servizio. Per ulteriori informazioni sul CodeCatalyst service principal e sul modello di fiducia, vedere [Comprendere il modello CodeCatalyst di fiducia](#).

- Per gestire il supporto per uno spazio tramite Connected Account AWS, puoi scegliere di creare e utilizzare il ruolo di AWSRoleForCodeCatalystSupportservizio che consente CodeCatalyst agli utenti di accedere al supporto. Per ulteriori informazioni sul supporto per uno CodeCatalyst spazio, consulta [Supporto per Amazon CodeCatalyst](#).

Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-**spaceName**del servizio

Puoi aggiungere un ruolo IAM per il tuo spazio da CodeCatalyst utilizzare per creare e accedere a risorse in un ambiente connesso Account AWS. Questo è chiamato [ruolo di servizio](#). Il modo più semplice per creare un ruolo di servizio è aggiungerne uno quando si crea lo spazio e scegliere

Il `CodeCatalystWorkflowDevelopmentRole-spaceName` opzione per quel ruolo. Questo non solo crea il ruolo di servizio con lo spazio `AdministratorAccess` allegato, ma crea anche la politica di fiducia che CodeCatalyst consente di assumere il ruolo per conto degli utenti nei progetti nello spazio. Il ruolo di servizio è limitato allo spazio, non ai singoli progetti. Per creare questo ruolo, consulta [Creazione del `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo per il tuo account e il tuo spazio](#). Puoi creare un solo ruolo per ogni spazio in ogni account.

Note

Questo ruolo è consigliato solo per gli account di sviluppo e utilizza la politica `AdministratorAccess` AWS gestita, che gli consente l'accesso completo alla creazione di nuove politiche e risorse Account AWS.

La politica allegata al `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo è progettata per funzionare con progetti creati con progetti esistenti nello spazio. Consente agli utenti di tali progetti di sviluppare, creare, testare e distribuire codice utilizzando le risorse disponibili nella rete. Account AWS Per ulteriori informazioni, vedere [Creazione di un ruolo per un AWS servizio](#).

La politica associata al `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo è la politica `AdministratorAccess` gestita in AWS. Si tratta di una politica che garantisce l'accesso completo a tutte le AWS azioni e le risorse. Per visualizzare il documento sulla policy JSON nella console IAM, consulta [AdministratorAccess](#).

La seguente politica di fiducia consente di CodeCatalyst assumere il `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo. Per ulteriori informazioni sul modello CodeCatalyst di fiducia, vedere [Comprendere il modello CodeCatalyst di fiducia](#).

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
```

```
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:codecatalyst::space/spaceId/project/*"
      }
    }
  }
}
```

Creazione del CodeCatalystWorkflowDevelopmentRole-**spaceName** ruolo per il tuo account e il tuo spazio

Segui questi passaggi per creare il CodeCatalystWorkflowDevelopmentRole-**spaceName** ruolo che verrà utilizzato per i flussi di lavoro nel tuo spazio. Per ogni account a cui desideri assegnare ruoli IAM da utilizzare nei progetti, nel tuo spazio, devi aggiungere un ruolo come il ruolo di sviluppatore.

Prima di iniziare, devi disporre dei privilegi di amministratore Account AWS o essere in grado di lavorare con il tuo amministratore. Per ulteriori informazioni su come Account AWS vengono utilizzati i ruoli IAM CodeCatalyst, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS](#).

Per creare e aggiungere CodeCatalyst CodeCatalystWorkflowDevelopmentRole-**spaceName**

1. Prima di iniziare a utilizzare la CodeCatalyst console, apri il Console di gestione AWS, quindi assicurati di aver effettuato l'accesso con lo stesso Account AWS nome del tuo spazio.
2. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
3. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
4. Scegli il link relativo alla Account AWS posizione in cui desideri creare il ruolo. Viene visualizzata la pagina dei Account AWS dettagli.
5. Scegli Gestisci ruoli da Console di gestione AWS.

La pagina Aggiungi ruolo IAM CodeCatalyst allo spazio Amazon si apre in Console di gestione AWS. Questa è la pagina di Amazon CodeCatalyst spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

6. Scegli Crea il ruolo di amministratore CodeCatalyst dello sviluppo in IAM. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di fiducia per il ruolo di sviluppo. Il ruolo avrà un nomeCodeCatalystWorkflowDevelopmentRole-**spaceName**.

Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere [Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName* del servizio](#).

Note

Questo ruolo è consigliato solo per gli account sviluppatore e utilizza la politica AdministratorAccess AWS gestita, che gli consente l'accesso completo alla creazione di nuove politiche e risorse Account AWS.

7. Scegli Crea ruolo di sviluppo.
8. Nella pagina delle connessioni, in Ruoli IAM disponibili per CodeCatalyst, visualizza il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo nell'elenco dei ruoli IAM aggiunti al tuo account.
9. Per tornare al tuo spazio, scegli Vai su Amazon CodeCatalyst.

Comprensione del ruolo AWSRoleForCodeCatalystSupport del servizio

Puoi aggiungere un ruolo IAM per il tuo spazio che CodeCatalyst gli utenti di uno spazio possono utilizzare per creare e accedere a casi di supporto. Questo è chiamato [ruolo di servizio](#) per il supporto. Il modo più semplice per creare un ruolo di servizio per l'assistenza è aggiungerne uno quando si crea lo spazio e scegliere l'AWSRoleForCodeCatalystSupport opzione per quel ruolo. Questo non solo crea la politica e il ruolo, ma crea anche la politica di fiducia che consente di CodeCatalyst assumere il ruolo per conto degli utenti nei progetti dello spazio. Il ruolo di servizio è limitato allo spazio, non ai singoli progetti. Per creare questo ruolo, consulta [Creazione del AWSRoleForCodeCatalystSupport ruolo per il tuo account e il tuo spazio](#).

La politica allegata al AWSRoleForCodeCatalystSupport ruolo è una politica gestita che fornisce l'accesso alle autorizzazioni di supporto. Per ulteriori informazioni, consulta [AWS politica gestita: AmazonCodeCatalystSupportAccess](#).

Il ruolo di fiducia per la politica consente di CodeCatalyst assumere il ruolo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst.amazonaws.com",
                "codecatalyst-runner.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
}
```

Creazione del `AWSRoleForCodeCatalystSupport` ruolo per il tuo account e il tuo spazio

Segui questi passaggi per creare il `AWSRoleForCodeCatalystSupport` ruolo che verrà utilizzato per i casi di assistenza nel tuo spazio. Il ruolo deve essere aggiunto all'account di fatturazione designato per lo spazio.

Prima di iniziare, devi disporre dei privilegi amministrativi Account AWS o essere in grado di lavorare con il tuo amministratore. Per ulteriori informazioni su come Account AWS vengono utilizzati i ruoli IAM CodeCatalyst, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS](#).

Per creare e aggiungere CodeCatalyst `AWSRoleForCodeCatalystSupport`

1. Prima di iniziare a utilizzare la CodeCatalyst console, apri il Console di gestione AWS, quindi assicurati di aver effettuato l'accesso con lo stesso Account AWS nome del tuo spazio.
2. Accedi al tuo CodeCatalyst spazio. Selezionare Settings (Impostazioni), quindi scegliere Account AWS.
3. Scegli il link relativo alla Account AWS posizione in cui desideri creare il ruolo. Viene visualizzata la pagina dei Account AWS dettagli.
4. Scegli Gestisci ruoli da Console di gestione AWS.

La pagina Aggiungi ruolo IAM CodeCatalyst allo spazio Amazon si apre in Console di gestione AWS. Questa è la pagina Amazon CodeCatalyst Spaces. Potrebbe essere necessario effettuare il login per accedere alla pagina.

5. In Dettagli CodeCatalyst dello spazio, scegli Aggiungi ruolo CodeCatalyst Support. Questa opzione crea un ruolo di servizio che contiene la politica di autorizzazioni e la politica di attendibilità per il ruolo di sviluppo in anteprima. Il ruolo avrà un nome `AWSRoleForCodeCatalystSupport` con un identificatore univoco aggiunto. Per ulteriori

informazioni sul ruolo e sulla politica relativa ai ruoli, vedere. [Comprensione del ruolo AWSRoleForCodeCatalystSupportdel servizio](#)

6. Nella pagina Aggiungi ruolo per CodeCatalyst Support, lascia selezionata l'impostazione predefinita, quindi scegli Crea ruolo.
7. In Ruoli IAM disponibili per CodeCatalyst, visualizza il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo nell'elenco dei ruoli IAM aggiunti al tuo account.
8. Per tornare al tuo spazio, scegli Vai su Amazon CodeCatalyst.

Configurazione dei ruoli IAM per le azioni del flusso di lavoro in CodeCatalyst

Questa sezione descrive in dettaglio i ruoli e le policy IAM che puoi creare da utilizzare con il tuo CodeCatalyst account. Per istruzioni su come creare ruoli di esempio, consulta [Creazione manuale di ruoli per le azioni del flusso di lavoro](#). Dopo aver creato il ruolo IAM, copia l'ARN del ruolo per aggiungere il ruolo IAM alla connessione del tuo account e associarlo all'ambiente del tuo progetto. Per ulteriori informazioni, consulta [Aggiungere ruoli IAM alle connessioni degli account](#).

CodeCatalyst creare un ruolo per l'accesso ad Amazon S3

Per le azioni CodeCatalyst di creazione del flusso di lavoro, puoi utilizzare il ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName* di servizio predefinito oppure puoi creare un ruolo IAM denominato CodeCatalystBuildRoleforS3Access. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle CloudFormation risorse del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Scrivi su bucket Amazon S3.
- Supporta la creazione di risorse con CloudFormation. Ciò richiede l'accesso ad Amazon S3.

Questo ruolo utilizza la seguente politica:

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.


```
"Resource": "*"

```

CodeCatalyst crea un ruolo per CloudFormation

Per le azioni CodeCatalyst di creazione del flusso di lavoro, puoi utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-*spaceName* servizio predefinito oppure puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle CloudFormation risorse del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Supporta la creazione di risorse con CloudFormation. Questo è necessario insieme al ruolo di CodeCatalyst costruzione per l'accesso ad Amazon S3 e al ruolo di CodeCatalyst distribuzione per CloudFormation

A questo ruolo devono essere associate le seguenti politiche AWS gestite:

- AWSCloudFormationFullAccess
- IAMFullAccess
- Amazon S3 FullAccess
- APIGatewayAmministratore Amazon
- AWSLambdaFullAccess

CodeCatalyst crea un ruolo per CDK

Per CodeCatalyst i flussi di lavoro che eseguono azioni di compilazione CDK, come l'applicazione Web moderna a tre livelli, puoi utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-*spaceName* servizio predefinito oppure puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'avvio e l'esecuzione dei comandi di compilazione CDK per le risorse del tuo CloudFormation Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Scrivi su bucket Amazon S3.

- Supporta la creazione di costrutti CDK e stack di CloudFormation risorse. Ciò richiede l'accesso ad Amazon S3 per lo storage degli artefatti, Amazon ECR per il supporto degli archivi di immagini e SSM per la governance e il monitoraggio del sistema per le istanze virtuali.

Questo ruolo utilizza la seguente politica:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

CodeCatalyst distribuisce ruolo per CloudFormation

Per le azioni CodeCatalyst di distribuzione del flusso di lavoro che utilizzano CloudFormation, puoi utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-*spaceName* servizio predefinito oppure puoi utilizzare una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle CloudFormation risorse del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Consente di CodeCatalyst richiamare una funzione λ tramite la quale eseguire la blue/green distribuzione. CloudFormation
- Consenti CodeCatalyst di creare e aggiornare stack e changeset in. CloudFormation

Questo ruolo utilizza la seguente politica:

```
{"Action": [  
    "cloudformation:CreateStack",  
    "cloudformation>DeleteStack",  
    "cloudformation:Describe*",  
    "cloudformation:UpdateStack",  
    "cloudformation:CreateChangeSet",  
    "cloudformation>DeleteChangeSet",  
    "cloudformation:ExecuteChangeSet",  
    "cloudformation:SetStackPolicy",
```

```
    "cloudformation:ValidateTemplate",
    "cloudformation:List*",
    "iam:PassRole"
  ],
  "Resource": "resource_ARN",
  "Effect": "Allow"
}
```

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

CodeCatalyst ruolo di implementazione per Amazon EC2

CodeCatalyst le azioni di distribuzione del flusso di lavoro utilizzano un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una politica con autorizzazioni mirate che è CodeCatalyst necessaria per eseguire attività EC2 sulle risorse Amazon del tuo Account AWS. La politica predefinita per il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo non include le autorizzazioni per Amazon EC2 o Amazon EC2 Auto Scaling.

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Crea EC2 distribuzioni Amazon.
- Leggi i tag su un'istanza o identifica un' EC2 istanza Amazon tramite i nomi dei gruppi di Auto Scaling.
- Leggi, crea, aggiorna ed elimina i gruppi, gli hook del ciclo di vita e le politiche di scalabilità di Amazon EC2 Auto Scaling.
- Pubblica informazioni su argomenti di Amazon SNS.
- Recupera informazioni sugli CloudWatch allarmi.
- Leggi e aggiorna Elastic Load Balancing.

Questo ruolo utilizza la seguente politica:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

CodeCatalyst ruolo di implementazione per Amazon ECS

Per le azioni relative al CodeCatalyst flusso di lavoro, puoi creare un ruolo IAM con le autorizzazioni necessarie. Puoi utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-*spaceName* servizio predefinito oppure puoi creare un ruolo IAM per le azioni di distribuzione da utilizzare per le CodeCatalyst distribuzioni Lambda. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle risorse Amazon ECS del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Avvia la distribuzione continua di Amazon ECS per conto di un CodeCatalyst utente, in un account specificato nella CodeCatalyst connessione.
- Leggi, aggiorna ed elimina i set di attività di Amazon ECS.
- Aggiorna i gruppi target, gli ascoltatori e le regole di Elastic Load Balancing.
- Richiama le funzioni Lambda.
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi. CloudWatch
- Pubblica informazioni su argomenti di Amazon SNS.

Questo ruolo utilizza la seguente politica:

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Action": [  
      "ecs:DescribeServices",
```

```

    "ecs:CreateTaskSet",
    "ecs:DeleteTaskSet",
    "ecs:ListClusters",
    "ecs:RegisterTaskDefinition",
    "ecs:UpdateServicePrimaryTaskSet",
    "ecs:UpdateService",
    "elasticloadbalancing:DescribeTargetGroups",
    "elasticloadbalancing:DescribeListeners",
    "elasticloadbalancing:ModifyListener",
    "elasticloadbalancing:DescribeRules",
    "elasticloadbalancing:ModifyRule",
    "lambda:InvokeFunction",
    "lambda:ListFunctions",
    "cloudwatch:DescribeAlarms",
    "sns:Publish",
    "sns:ListTopics",
    "s3:GetObject",
    "s3:GetObjectVersion",
    "codedeploy:CreateApplication",
    "codedeploy:CreateDeployment",
    "codedeploy:CreateDeploymentGroup",
    "codedeploy:GetApplication",
    "codedeploy:GetDeployment",
    "codedeploy:GetDeploymentGroup",
    "codedeploy:ListApplications",
    "codedeploy:ListDeploymentGroups",
    "codedeploy:ListDeployments",
    "codedeploy:StopDeployment",
    "codedeploy:GetDeploymentTarget",
    "codedeploy:ListDeploymentTargets",
    "codedeploy:GetDeploymentConfig",
    "codedeploy:GetApplicationRevision",
    "codedeploy:RegisterApplicationRevision",
    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [

```

```
"iam:PassRole"
],
"Effect":"Allow",
"Resource":"*",
"Condition":{"StringLike":{"iam:PassedToService":["
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
}
}
}]
}
```

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"

```

CodeCatalyst ruolo di implementazione per Lambda

Per le azioni relative al CodeCatalyst flusso di lavoro, puoi creare un ruolo IAM con le autorizzazioni necessarie. Puoi utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-*spaceName* servizio predefinito oppure creare un ruolo IAM per le azioni di distribuzione da utilizzare per le CodeCatalyst distribuzioni Lambda. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle risorse Lambda del tuo. Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Leggi, aggiorna e richiama funzioni e alias Lambda.
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi Events. CloudWatch
- Pubblica informazioni su argomenti di Amazon SNS.

Questo ruolo utilizza la seguente politica:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

CodeCatalyst ruolo di implementazione per Lambda

Per le azioni del CodeCatalyst flusso di lavoro, puoi utilizzare il ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName* di servizio predefinito oppure puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle risorse Lambda del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Leggi, aggiorna e richiama funzioni e alias Lambda.
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi. CloudWatch
- Pubblica informazioni su argomenti di Amazon SNS.

Questo ruolo utilizza la seguente politica:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

CodeCatalyst distribuisce ruolo per AWS SAM

Per le azioni del CodeCatalyst flusso di lavoro, puoi utilizzare il ruolo di CodeCatalystWorkflowDevelopmentRole-**spaceName** servizio predefinito oppure puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività AWS SAM e CloudFormation risorse del tuo. Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Consenti CodeCatalyst di richiamare una funzione Lambda per eseguire la distribuzione di applicazioni serverless AWS SAM e CLI.
- Consenti di CodeCatalyst creare e aggiornare stack e changeset in. CloudFormation

Questo ruolo utilizza la seguente politica:

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

CodeCatalyst ruolo di sola lettura per Amazon EC2

Per le azioni CodeCatalyst del flusso di lavoro, puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una politica con autorizzazioni mirate che è CodeCatalyst necessaria per eseguire attività EC2 sulle risorse Amazon del tuo. Account AWS Il ruolo CodeCatalystWorkflowDevelopmentRole-**spaceName** di servizio non include le autorizzazioni per Amazon EC2 o le azioni descritte per Amazon CloudWatch.

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Verifica lo stato delle EC2 istanze Amazon.
- Ottieni CloudWatch metriche per le EC2 istanze Amazon.

Questo ruolo utilizza la seguente politica:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*" 
```

CodeCatalyst ruolo di sola lettura per Amazon ECS

Per le azioni CodeCatalyst del flusso di lavoro, puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle risorse Amazon ECS del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per eseguire le seguenti operazioni:

- Leggi i set di attività di Amazon ECS.
- Recupera informazioni sugli CloudWatch allarmi.

Questo ruolo utilizza la seguente politica:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*" 
```

CodeCatalyst ruolo di sola lettura per Lambda

Per le azioni CodeCatalyst del flusso di lavoro, puoi creare un ruolo IAM con le autorizzazioni necessarie. Questo ruolo utilizza una policy con autorizzazioni mirate che CodeCatalyst richiede l'esecuzione di attività sulle risorse Lambda del tuo Account AWS

Questo ruolo fornisce le autorizzazioni per quanto segue:

- Leggi le funzioni e gli alias Lambda.
- Accedi ai file di revisione nei bucket Amazon S3.
- Recupera informazioni sugli allarmi. CloudWatch

Questo ruolo utilizza la policy seguente .

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni sul flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

Creazione manuale di ruoli per le azioni del flusso di lavoro

CodeCatalyst le azioni del flusso di lavoro utilizzano i ruoli IAM creati dall'utente, denominati build role, deploy role e stack role.

Segui questi passaggi per creare questi ruoli in IAM.

Per creare un ruolo di distribuzione

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - c. Nel riquadro di navigazione, scegli Policy.
 - d. Scegli Create Policy (Crea policy).
 - e. Scegliere la scheda JSON.
 - f. Eliminare il codice esistente.
 - g. Incolla il codice seguente:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-deploy-policy
```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di distribuzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione, cerca **codecatalyst-deploy-policy** e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per Nome del ruolo, inserisci:

```
codecatalyst-deploy-role
```

- i. Per la descrizione del ruolo, inserisci:

```
CodeCatalyst deploy role
```

j. Scegli Crea ruolo.

Ora hai creato un ruolo di distribuzione con una politica di fiducia e una politica di autorizzazioni.

3. Ottenere l'ARN del ruolo di distribuzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato `()codecatalyst-deploy-role`.
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di distribuzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Per creare un ruolo di costruzione

1. Crea una politica per il ruolo, come segue:

- a. Accedi a AWS.
- b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
- c. Nel riquadro di navigazione, scegli Policy.
- d. Scegli Create Policy (Crea policy).
- e. Scegliere la scheda JSON.
- f. Eliminare il codice esistente.
- g. Incolla il codice seguente:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*" 
```

- h. Scegliere Next: Tags (Successivo: Tag).

- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

codecatalyst-build-policy

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di costruzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Nelle politiche di autorizzazione, cerca **codecatalyst-build-policy** e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per Nome del ruolo, inserisci:

codecatalyst-build-role

- i. Per la descrizione del ruolo, inserisci:

CodeCatalyst build role

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di fiducia e una politica di autorizzazioni.

3. Ottieni l'ARN del ruolo di costruzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (**codecatalyst-build-role**).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di costruzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Per creare un ruolo stack

Note

Non è necessario creare un ruolo stack, sebbene sia consigliabile farlo per motivi di sicurezza. Se non crei il ruolo stack, dovrai aggiungere al ruolo di distribuzione le politiche di autorizzazione descritte più avanti in questa procedura.

1. Accedi per AWS utilizzare l'account in cui desideri distribuire il tuo stack.
2. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
3. Nel riquadro di navigazione, scegli Ruoli, quindi scegli Crea ruolo.
4. Nella parte superiore, scegli AWS servizio.
5. Dall'elenco dei servizi, scegli CloudFormation.
6. Scegli Successivo: autorizzazioni.
7. Nella casella di ricerca, aggiungi le politiche necessarie per accedere alle risorse del tuo stack. Ad esempio, se lo stack include una AWS Lambda funzione, è necessario aggiungere una policy che garantisca l'accesso a Lambda.

Tip

Se non sei sicuro delle politiche da aggiungere, puoi ometterle per ora. Quando provi l'azione, se non disponi delle autorizzazioni corrette, CloudFormation genera errori che mostrano quali autorizzazioni devi aggiungere.

8. Scegliere Next: Tags (Successivo: Tag).
9. Scegliere Next:Review (Successivo: Rivedi).
10. Per il nome del ruolo, inserisci:

codecatalyst-stack-role

11. Scegli Crea ruolo.
12. Per ottenere l'ARN del ruolo stack, procedi come segue:
 - a. Nel riquadro di navigazione, seleziona Ruoli.
 - b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato ()codecatalyst-stack-role.
 - c. Scegli il ruolo dall'elenco.
 - d. Nella pagina Riepilogo, copia il valore Role ARN.

Utilizzo AWS CloudFormation per creare politiche e ruoli in IAM

Puoi scegliere di creare e utilizzare AWS CloudFormation modelli per creare le politiche e i ruoli necessari per accedere alle risorse in e Account AWS per i tuoi CodeCatalyst progetti e flussi di lavoro. CloudFormation è un servizio che ti aiuta a modellare e configurare AWS le tue risorse in modo da poter dedicare meno tempo alla gestione di tali risorse e più tempo a concentrarti sulle applicazioni che girano su AWS. Se intendi creare ruoli in più ruoli Account AWS, la creazione di un modello può aiutarti a svolgere questa attività più rapidamente.

Il modello di esempio seguente crea un ruolo e una politica di implementazione.

```
Parameters:
  CodeCatalystAccountId:
    Type: String
    Description: Account ID from the connections page
  ExternalId:
    Type: String
    Description: External ID from the connections page
Resources:
  CrossAccountRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              AWS:
```

```

        - !Ref CodeCatalystAccountId
    Action:
        - 'sts:AssumeRole'
    Condition:
        StringEquals:
            sts:ExternalId: !Ref ExternalId
Path: /
Policies:
    - PolicyName: CodeCatalyst-CloudFormation-action-policy
      PolicyDocument:
        Version: "2012-10-17"
        Statement:
            - Effect: Allow
              Action:
                - 'cloudformation:CreateStack'
                - 'cloudformation>DeleteStack'
                - 'cloudformation:Describe*'
                - 'cloudformation:UpdateStack'
                - 'cloudformation:CreateChangeSet'
                - 'cloudformation>DeleteChangeSet'
                - 'cloudformation:ExecuteChangeSet'
                - 'cloudformation:SetStackPolicy'
                - 'cloudformation:ValidateTemplate'
                - 'cloudformation:List*'
                - 'iam:PassRole'
              Resource: '*'

```

Creazione manuale del ruolo per il blueprint dell'applicazione Web

Il blueprint dell'applicazione CodeCatalyst Web utilizza i ruoli IAM creati dall'utente, denominati build role for CDK, deploy role e stack role.

Segui questi passaggi per creare il ruolo in IAM.

Per creare un ruolo di costruzione

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - c. Nel riquadro di navigazione, scegli Policy.
 - d. Scegliere Create Policy (Crea policy).

- e. Scegliere la scheda JSON.
- f. Eliminare il codice esistente.
- g. Incolla il codice seguente:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-webapp-build-policy
```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

- 2. Crea il ruolo di costruzione, come segue:
 - a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
 - b. Scegli una politica di fiducia personalizzata.
 - c. Elimina la politica di fiducia personalizzata esistente.
 - d. Aggiungi la seguente politica di fiducia personalizzata:
 - e. Scegli Next (Successivo).
 - f. Allega la politica delle autorizzazioni al ruolo di costruzione. Nella pagina Aggiungi autorizzazioni, nella sezione Politiche di autorizzazione, cerca `codecatalyst-webapp-build-policy` e seleziona la relativa casella di controllo.
 - g. Scegli Next (Successivo).
 - h. Per Nome del ruolo, inserisci:

```
codecatalyst-webapp-build-role
```

- i. Per la descrizione del ruolo, inserisci:

CodeCatalyst Web app build role

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di fiducia e una politica di autorizzazioni.

3. Allega la politica delle autorizzazioni al ruolo di compilazione, come segue:

- a. Nel riquadro di navigazione, scegli Ruoli, quindi cerca `codecatalyst-webapp-build-role`.
- b. Scegli `codecatalyst-webapp-build-role` di visualizzarne i dettagli.
- c. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi scegli Allega criteri.
- d. Cerca `codecatalyst-webapp-build-policy`, seleziona la relativa casella di controllo, quindi scegli Allega politiche.

Ora hai allegato la politica delle autorizzazioni al ruolo di compilazione. Il ruolo build ora ha due politiche: una politica di autorizzazioni e una politica di fiducia.

4. Ottieni l'ARN del ruolo di costruzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (`codecatalyst-webapp-build-role`).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di costruzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Creazione manuale di ruoli per il blueprint SAM

Il blueprint CodeCatalyst SAM utilizza i ruoli IAM creati dall'utente, denominati build role for CloudFormation e deploy role for SAM.

Segui questi passaggi per creare i ruoli in IAM.

Per creare un ruolo di costruzione per CloudFormation

1. Crea una politica per il ruolo, come segue:
 - a. Accedi a AWS.
 - b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - c. Nel riquadro di navigazione, scegli Policy.
 - d. Scegliere Create Policy (Crea policy).
 - e. Scegliere la scheda JSON.
 - f. Eliminare il codice esistente.
 - g. Incolla il codice seguente:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:*",
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"
```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

codecatalyst-SAM-build-policy

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di costruzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Allega la politica delle autorizzazioni al ruolo di costruzione. Nella pagina Aggiungi autorizzazioni, nella sezione Politiche di autorizzazione, cerca **codecatalyst-SAM-build-policy** e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per Nome del ruolo, inserisci:

codecatalyst-SAM-build-role

- i. Per la descrizione del ruolo, inserisci:

CodeCatalyst SAM build role

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di fiducia e una politica di autorizzazioni.

3. Allega la politica delle autorizzazioni al ruolo di compilazione, come segue:

- a. Nel riquadro di navigazione, scegli Ruoli, quindi cerca **codecatalyst-SAM-build-role**.
- b. Scegli **codecatalyst-SAM-build-role** di visualizzarne i dettagli.

- d. Cercacodecatalyst-SAM-build-policy, seleziona la relativa casella di controllo, quindi scegli Allega politiche.

Ora hai allegato la politica delle autorizzazioni al ruolo di compilazione. Il ruolo build ora ha due politiche: una politica di autorizzazioni e una politica di fiducia.

4. Ottieni l'ARN del ruolo di costruzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (codecatalyst-SAM-build-role).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di costruzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Per creare un ruolo di distribuzione per SAM

1. Crea una politica per il ruolo, come segue:

- a. Accedi a AWS.
- b. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
- c. Nel riquadro di navigazione, scegli Policy.
- d. Scegliere Create Policy (Crea policy).
- e. Scegliere la scheda JSON.
- f. Eliminare il codice esistente.
- g. Incolla il codice seguente:

 Note

La prima volta che il ruolo viene utilizzato per eseguire azioni del flusso di lavoro, utilizza il carattere jolly nell'informativa sulla politica delle risorse, quindi definisci la politica inserendo il nome della risorsa dopo che è disponibile.

```
"Resource": "*"

```

- h. Scegliere Next: Tags (Successivo: Tag).
- i. Scegliere Next:Review (Successivo: Rivedi).
- j. In Nome, inserisci:

```
codecatalyst-SAM-deploy-policy

```

- k. Scegli Crea policy.

Ora hai creato una politica di autorizzazioni.

2. Crea il ruolo di costruzione, come segue:

- a. Nel pannello di navigazione, scegli Ruoli e quindi Crea ruolo.
- b. Scegli una politica di fiducia personalizzata.
- c. Elimina la politica di fiducia personalizzata esistente.
- d. Aggiungi la seguente politica di fiducia personalizzata:
- e. Scegli Next (Successivo).
- f. Allega la politica delle autorizzazioni al ruolo di costruzione. Nella pagina Aggiungi autorizzazioni, nella sezione Politiche di autorizzazione, cerca `codecatalyst-SAM-deploy-policy` e seleziona la relativa casella di controllo.
- g. Scegli Next (Successivo).
- h. Per Nome del ruolo, inserisci:

```
codecatalyst-SAM-deploy-role

```

- i. Per la descrizione del ruolo, inserisci:

```
CodeCatalyst SAM deploy role

```

- j. Scegli Crea ruolo.

Ora hai creato un ruolo di sviluppo con una politica di fiducia e una politica di autorizzazioni.

3. Allega la politica delle autorizzazioni al ruolo di compilazione, come segue:

- a. Nel riquadro di navigazione, scegli Ruoli, quindi cercacodecatalyst-SAM-deploy-role.
- b. Scegli codecatalyst-SAM-deploy-role di visualizzarne i dettagli.
- c. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi scegli Allega criteri.
- d. Cercacodecatalyst-SAM-deploy-policy, seleziona la relativa casella di controllo, quindi scegli Allega politiche.

Ora hai allegato la politica delle autorizzazioni al ruolo di compilazione. Il ruolo build ora ha due politiche: una politica di autorizzazioni e una politica di fiducia.

4. Ottieni l'ARN del ruolo di costruzione, come segue:

- a. Nel riquadro di navigazione, seleziona Ruoli.
- b. Nella casella di ricerca, inserisci il nome del ruolo che hai appena creato (codecatalyst-SAM-deploy-role).
- c. Scegli il ruolo dall'elenco.

Viene visualizzata la pagina di riepilogo del ruolo.

- d. In alto, copia il valore ARN.

Ora hai creato il ruolo di costruzione con le autorizzazioni appropriate e ottenuto il relativo ARN.

Convalida della conformità per Amazon CodeCatalyst

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la [Documentazione AWS sulla sicurezza](#).

Resilienza in Amazon CodeCatalyst

L'infrastruttura AWS globale è costruita attorno a Regioni AWS zone di disponibilità. Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni sulle zone Regioni AWS di disponibilità, vedere [AWS Global Infrastructure](#). Per ulteriori informazioni su quali CodeCatalyst dati vengono replicati Regioni AWS, consulta [Protezione dei dati in Amazon CodeCatalyst](#).

Sicurezza dell'infrastruttura in Amazon CodeCatalyst

In quanto servizio gestito, Amazon CodeCatalyst è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere CodeCatalyst attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Analisi della configurazione e delle vulnerabilità in Amazon CodeCatalyst

La configurazione e i controlli IT sono una responsabilità condivisa tra te AWS e te, nostro cliente. Per ulteriori informazioni, consulta il [modello di responsabilità AWS condivisa](#).

I tuoi dati e la tua privacy in Amazon CodeCatalyst

Amazon CodeCatalyst prende sul serio la tua privacy e la sicurezza delle tue informazioni è la nostra massima priorità. Puoi leggere ulteriori informazioni su come gestiamo le tue informazioni nell'[AWS Informativa sulla privacy](#).

Per richiedere e visualizzare i tuoi dati, consulta [Richiesta dei tuoi dati](#) in. Riferimenti generali di AWS

Eliminazione del profilo AWS Builder ID

L'eliminazione del tuo profilo è un'azione permanente che non può essere annullata. Il processo di eliminazione inizia immediatamente dopo aver scelto Elimina. Amazon CodeCatalyst inizia a eliminare il tuo profilo e tutte le informazioni personali associate. Il completamento di questo processo può richiedere fino a 90 giorni.

Quando il tuo profilo viene eliminato, non puoi accedere o recuperare i tuoi dati in Amazon CodeCatalyst. Ciò include i token di accesso personali, i ruoli, le iscrizioni degli utenti e tutti CodeCatalyst gli spazi Amazon di cui sei l'unico membro. Non puoi più accedere ad Amazon CodeCatalyst.

Per informazioni su come eliminare il tuo profilo AWS Builder ID, consulta [Eliminazione del tuo ID AWS Builder](#) in. Riferimenti generali di AWS

Le migliori pratiche per le azioni del flusso di lavoro in Amazon CodeCatalyst

Esistono diverse best practice di sicurezza da prendere in considerazione durante lo sviluppo dei flussi di lavoro. CodeCatalyst Le seguenti sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

Argomenti

- [Informazioni sensibili](#)
- [Termini di licenza](#)
- [Codice non affidabile](#)
- [GitHub Azioni](#)

Informazioni sensibili

Non incorporare informazioni sensibili nel tuo YAML. Invece di incorporare credenziali, chiavi o token nel tuo YAML, ti consigliamo di utilizzare dei segreti. CodeCatalyst I segreti offrono un modo semplice per archiviare e fare riferimento a informazioni sensibili all'interno del tuo YAML.

Termini di licenza

Assicurati di prestare attenzione ai termini di licenza dell'azione che scegli di utilizzare.

Codice non affidabile

Le azioni sono generalmente moduli autonomi e monouso che possono essere condivisi all'interno di un progetto, di uno spazio o di una comunità più ampia. L'utilizzo di codice altrui può essere un grande vantaggio in termini di praticità ed efficienza, ma introduce anche un nuovo vettore di minacce. Consulta le seguenti sezioni per assicurarti di seguire le migliori pratiche per proteggere i flussi di lavoro CI/CD.

GitHub Azioni

GitHub Le azioni sono open source, create e gestite dalla comunità. Seguiamo il [modello di responsabilità condivisa](#) e consideriamo il codice sorgente di GitHub Actions come dati dei clienti di cui sei responsabile. GitHub Actions può avere accesso ai segreti, ai token del repository, al codice sorgente, ai link degli account e al tempo di elaborazione. Assicurati di avere fiducia nell'affidabilità e nella sicurezza delle GitHub azioni che intendi eseguire.

Linee guida più specifiche e migliori pratiche di sicurezza per GitHub Actions:

- [Rafforzamento della sicurezza](#)
- [Prevenzione delle richieste proprie](#)
- [Input non attendibile](#)
- [Come fidarsi dei propri elementi costitutivi](#)

Comprendere il modello CodeCatalyst di fiducia

Il modello di CodeCatalyst fiducia di Amazon CodeCatalyst consente di assumere il ruolo di servizio nel mondo connesso Account AWS. Il modello collega il ruolo IAM, i responsabili del CodeCatalyst servizio e lo CodeCatalyst spazio. La policy di fiducia utilizza la chiave di `aws:SourceArn` condizione per concedere le autorizzazioni allo CodeCatalyst spazio specificato nella chiave di condizione. Per ulteriori informazioni su questa chiave di condizione, consulta [aws: SourceArn](#) nella IAM User Guide.

Una policy di attendibilità documento di policy JSON in cui si definiscono i principali considerati attendibili per assumere il ruolo. Una policy di attendibilità del ruolo è una policy basata sulle risorse

collegata a un ruolo in IAM. Per ulteriori informazioni, consulta [Termini e concetti](#) nella Guida per l'utente IAM. Per dettagli sui principali servizi per CodeCatalyst, consulta [Principali del servizio per CodeCatalyst](#).

Nella seguente politica di attendibilità, ai principali di servizio elencati nell'Principalelemento vengono concesse le autorizzazioni previste dalla politica basata sulle risorse e il Condition blocco viene utilizzato per limitare l'accesso alla risorsa limitata.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codecatalyst::space/spaceId/project/*"
        }
      }
    }
  ]
}
```

Nella politica di fiducia, i responsabili del CodeCatalyst servizio hanno accesso tramite la chiave di `aws:SourceArn` condizione, che contiene l'Amazon Resource Name (ARN) per CodeCatalyst l'ID dello spazio. L'ARN utilizza il seguente formato:

```
arn:aws:codecatalyst::space/spaceId/project/*
```

⚠ Important

Utilizzate l'ID dello spazio solo nei tasti di condizione, ad esempio `aws:SourceArn`. Non utilizzare l'ID dello spazio nelle istruzioni delle politiche IAM come ARN di risorse.

Come best practice, riduci il più possibile le autorizzazioni nella policy.

- È possibile utilizzare il carattere jolly (*) nella chiave di `aws:SourceArn` condizione per specificare tutti i progetti nello spazio con. `project/*`
- È possibile specificare le autorizzazioni a livello di risorsa nella chiave di `aws:SourceArn` condizione per un progetto specifico nello spazio con. `project/projectId`

Principali del servizio per CodeCatalyst

Si utilizza l'Principalelemento in una policy JSON basata sulle risorse per specificare il principale a cui è consentito o negato l'accesso a una risorsa. I principali che è possibile specificare nella policy di attendibilità includono utenti, ruoli, account e servizi. Non è possibile utilizzare l'Principalelemento in una policy basata sull'identità; allo stesso modo, non è possibile identificare un gruppo di utenti come principale in una policy (ad esempio una policy basata sulle risorse) perché i gruppi si riferiscono alle autorizzazioni, non all'autenticazione, e i principali sono entità IAM autenticate.

Nella policy di fiducia, è possibile specificare l'Principalelemento di una policy basata sulle Servizi AWS risorse o le chiavi di condizione che supportano i principali. I principali del servizio sono definiti dal servizio. Di seguito sono riportati i principali di servizio definiti per: CodeCatalyst

- `codecatalyst.amazonaws.com` - Questo responsabile del servizio viene utilizzato per un ruolo a cui verrà concesso l'accesso. CodeCatalyst AWS
- `codecatalyst-runner.amazonaws.com` - Questo responsabile del servizio viene utilizzato per un ruolo che garantirà l'accesso alle risorse nelle distribuzioni per i flussi di lavoro. CodeCatalyst AWS

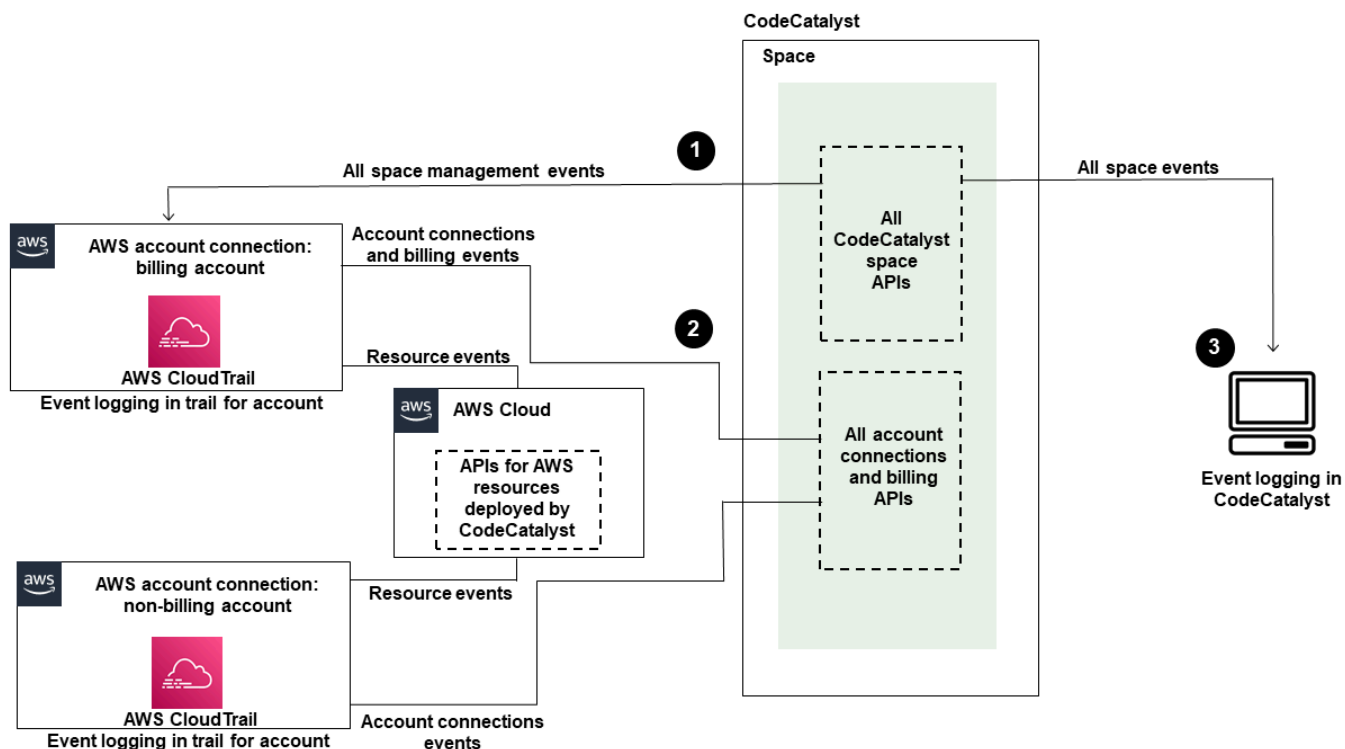
[Per ulteriori informazioni, consulta Elementi della policy JSON: Principal nella IAM User Guide.AWS](#)

Monitoraggio degli eventi e delle chiamate API tramite la registrazione

In Amazon CodeCatalyst, gli eventi di gestione dello spazio vengono raccolti AWS CloudTrail e registrati nel percorso dell'account di fatturazione dello spazio. CloudTrail la registrazione è il metodo principale per gestire la registrazione CodeCatalyst degli eventi e un metodo secondario è la visualizzazione della registrazione degli eventi. CodeCatalyst

Gli eventi nell'account vengono registrati con il percorso e il bucket designato configurato per. Account AWS

Il diagramma seguente mostra come tutti gli eventi di gestione dello spazio vengono registrati per l'account di fatturazione, mentre gli eventi dell'account e connections/billing gli eventi relativi alle AWS risorse vengono registrati CloudTrail per il rispettivo account. CloudTrail



Il diagramma illustra i passaggi seguenti:

1. Quando viene creato uno spazio, uno Account AWS viene connesso allo spazio e viene designato come account di fatturazione. Il percorso utilizzato è il percorso creato CloudTrail per l'account di fatturazione, in cui vengono registrati gli eventi spaziali. CloudTrail acquisisce le chiamate API e

gli eventi correlati effettuati da o per conto di uno CodeCatalyst spazio e invia i file di registro a un bucket S3 specificato dall'utente. Se l'account di fatturazione passa a un altro account AWS, gli eventi spaziali vengono registrati nel trail and bucket di quell'account. Per ulteriori informazioni sugli eventi di CodeCatalyst gestione registrati da, consulta. CloudTrail [CodeCatalyst informazioni in CloudTrail](#)

2. Gli altri account collegati allo spazio, incluso l'account di fatturazione, registrano un sottoinsieme di eventi per le connessioni all'account e gli eventi di fatturazione. CodeCatalyst i flussi di lavoro che generano eventi relativi all'account per AWS le risorse distribuite per quell'account vengono inoltre registrati nel trail and bucket di. Account AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto di CodeCatalyst uno spazio e consegna i file di registro a un bucket S3 specificato dall'utente. Per ulteriori informazioni sugli eventi di CodeCatalyst gestione registrati da, consulta. CloudTrail [Accesso agli eventi registrati utilizzando la registrazione degli eventi](#)
3. È inoltre possibile monitorare CodeCatalyst le azioni nel proprio spazio entro un periodo di tempo specifico nello spazio con il [list-event-logs](#) comando che utilizza il AWS CLI. Per ulteriori informazioni, consulta [la Amazon CodeCatalyst API Reference Guide](#). Devi avere il ruolo di amministratore dello spazio per richiamare l'elenco degli eventi per CodeCatalyst le azioni nel tuo spazio. Per ulteriori informazioni, consulta [Accesso agli eventi registrati utilizzando la registrazione degli eventi](#).

Note

ListEventLogsgarantisce gli eventi degli ultimi 30 giorni in un determinato spazio. È inoltre possibile visualizzare e recuperare un elenco degli eventi di gestione degli ultimi 90 giorni CodeCatalyst nella AWS CloudTrail console visualizzando la cronologia degli eventi o creando un percorso per creare e mantenere un record degli eventi che si estende negli ultimi 90 giorni. Per ulteriori informazioni, vedere [Utilizzo della cronologia CloudTrail degli eventi](#) e [Utilizzo dei CloudTrail percorsi](#).

Note

AWS le risorse distribuite in account connessi per i CodeCatalyst flussi di lavoro non vengono registrate come parte della CloudTrail registrazione dello spazio. CodeCatalyst Ad esempio, CodeCatalyst le risorse includono uno spazio o un progetto. AWS le risorse includono

un servizio Amazon ECS o una funzione Lambda. È necessario configurare la CloudTrail registrazione separatamente per ogni area Account AWS in cui vengono distribuite le risorse.

Ecco un possibile flusso per il monitoraggio degli eventi. CodeCatalyst

Mary Major è un'amministratrice CodeCatalyst dello spazio e visualizza tutti gli eventi CodeCatalyst di gestione delle risorse a livello di spazio e di progetto nello spazio registrato. CloudTrail Vedi ad [CodeCatalyst informazioni in CloudTrail](#) esempio gli eventi registrati. CloudTrail

Per le risorse create in CodeCatalyst, come Dev Environments, Mary visualizza la cronologia degli eventi nell'account di fatturazione dello spazio e analizza gli eventi in cui gli ambienti di sviluppo sono stati creati dai membri del progetto. CodeCatalyst L'evento fornisce l'identity store, il tipo di identità IAM e le credenziali per l'ID AWS Builder per l'utente che ha creato l'ambiente di sviluppo. Per le risorse create AWS quando vengono distribuite dai flussi di lavoro in CodeCatalyst, ad esempio una funzione Lambda per una distribuzione senza server, il Account AWS proprietario può visualizzare la cronologia degli eventi per il percorso associato all'azione di distribuzione del flusso di lavoro separata Account AWS (che è anche un account connesso a CodeCatalyst).

Per ulteriori informazioni, Mary può anche visualizzare gli eventi relativi CodeCatalyst APIs a tutti gli utenti dello spazio utilizzando il comando in. [list-event-logs](#) AWS CLI

Argomenti

- [Monitoraggio delle chiamate API Account AWS utilizzando la AWS CloudTrail registrazione](#)
- [Accesso agli eventi registrati utilizzando la registrazione degli eventi](#)

Monitoraggio delle chiamate API Account AWS utilizzando la AWS CloudTrail registrazione

Amazon CodeCatalyst è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o un Servizio AWS. CloudTrail acquisisce le chiamate API effettuate per conto di CodeCatalyst in connected Account AWS come eventi. Se crei un trail, puoi abilitare la consegna continua di CloudTrail eventi a un bucket S3, inclusi gli eventi per. CodeCatalyst Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi.

CodeCatalyst supporta la registrazione delle seguenti azioni come eventi nei file di CloudTrail registro:

- Gli eventi di gestione CodeCatalyst degli spazi verranno registrati nell'account Account AWS di fatturazione designato per lo spazio. Per ulteriori informazioni, consulta [CodeCatalyst eventi spaziali](#).

Note

Gli eventi di dati per CodeCatalyst gli spazi sono accessibili utilizzando la CLI come descritto in. [Accesso agli eventi registrati utilizzando la registrazione degli eventi](#)

- Gli eventi relativi alle risorse utilizzate nelle azioni CodeCatalyst del flusso di lavoro che si verificano in un ambiente connesso Account AWS verranno registrati come eventi in tale ambiente. Account AWS Per ulteriori informazioni, consulta [CodeCatalyst connessioni all'account ed eventi di fatturazione](#).

Important

Sebbene sia possibile associare più account a uno spazio, la CloudTrail registrazione degli eventi CodeCatalyst negli spazi e nei progetti si applica solo all'account di fatturazione.

L'account di fatturazione spaziale è il tuo a Account AWS cui vengono addebitate le CodeCatalyst risorse oltre il AWS livello gratuito. È possibile collegare più account a uno spazio, mentre solo un account può essere l'account di fatturazione designato. L'account di fatturazione o altri account connessi per lo spazio possono avere ruoli IAM utilizzati per distribuire AWS risorse e infrastrutture, come un cluster Amazon ECS o un bucket S3, dai flussi di lavoro. CodeCatalyst Puoi utilizzare il flusso di lavoro YAML per identificare quello su cui hai eseguito la distribuzione. Account AWS

Note

AWS le risorse distribuite in account connessi per i CodeCatalyst flussi di lavoro non vengono registrate come parte della registrazione dello spazio. CloudTrail CodeCatalyst Ad esempio, CodeCatalyst le risorse includono uno spazio o un progetto. AWS le risorse includono un servizio Amazon ECS o una funzione Lambda. CloudTrail la registrazione deve essere configurata separatamente per ogni area Account AWS in cui vengono distribuite le risorse.

CodeCatalyst l'accesso agli account collegati include le seguenti considerazioni:

- L'accesso agli CloudTrail eventi viene gestito con IAM nell'account connesso e non in CodeCatalyst
- Le connessioni di terze parti, come il collegamento a un GitHub repository, comporteranno la registrazione dei nomi delle risorse di terze parti nei CloudTrail log.

Note

CloudTrail la registrazione CodeCatalyst degli eventi avviene a livello di spazio e non isola gli eventi in base ai confini del progetto.

Per ulteriori informazioni su CloudTrail, consulta la Guida per l'[AWS CloudTrail utente](#).

Note

Questa sezione descrive CloudTrail la registrazione di tutti gli eventi registrati in uno CodeCatalyst spazio registrato e a Account AWS cui sono collegati. CodeCatalyst Inoltre, per esaminare tutti gli eventi registrati in uno CodeCatalyst spazio, è possibile utilizzare anche il AWS CLI comando `and. aws codecatalyst list-event-logs` Per ulteriori informazioni, consulta [Accesso agli eventi registrati utilizzando la registrazione degli eventi](#).

CodeCatalyst eventi spaziali

Le azioni intraprese CodeCatalyst per la gestione delle risorse a livello di spazio e di progetto vengono registrate nell'account di fatturazione dello spazio. Per la CloudTrail registrazione di uno CodeCatalyst spazio, gli eventi vengono registrati con le seguenti considerazioni.

- CloudTrail gli eventi si applicano all'intero spazio e non sono limitati a nessun singolo progetto.
- Quando si connette un account Account AWS a uno CodeCatalyst spazio, gli eventi registrabili per le connessioni degli account verranno registrati in quello spazio. Account AWS Dopo aver abilitato questa connessione, non è possibile disabilitarla.
- Quando colleghi un Account AWS a uno CodeCatalyst spazio e lo imposti come account di fatturazione per lo spazio, gli eventi verranno registrati in quello spazio. Account AWS Dopo aver abilitato questa connessione, non è possibile disabilitarla.

Gli eventi per le risorse a livello di spazio e di progetto vengono registrati solo nell'account di fatturazione. Per modificare l'account di CloudTrail destinazione, aggiorna l'account di fatturazione. CodeCatalyst All'inizio del successivo ciclo di fatturazione mensile, la modifica avrà effetto per il nuovo account di fatturazione in. CodeCatalyst Successivamente, l'account di CloudTrail destinazione viene aggiornato.

Di seguito sono riportati alcuni esempi di eventi correlati alle azioni CodeCatalyst per la gestione delle risorse a livello di spazio e di progetto. AWS Quanto segue APIs viene rilasciato tramite SDK e CLI. Gli eventi verranno registrati nell'account di fatturazione Account AWS specificato per lo spazio. CodeCatalyst

- [CreateDevEnvironment](#)
- [CreateProject](#)
- [DeleteDevEnvironment](#)
- [GetDevEnvironment](#)
- [GetProject](#)
- [GetSpace](#)
- [GetSubscription](#)
- [ListDevEnvironments](#)
- [ListDevEnvironmentSessions](#)
- [ListEventLogs](#)
- [ListProjects](#)
- [ListSourceRepositories](#)
- [StartDevEnvironment](#)
- [StartDevEnvironmentSession](#)
- [StopDevEnvironment](#)
- [StopDevEnvironmentSession](#)
- [UpdateDevEnvironment](#)

CodeCatalyst connessioni all'account ed eventi di fatturazione

Di seguito sono riportati alcuni esempi di eventi correlati alle azioni relative alle connessioni all'CodeCatalystaccount o alla fatturazione: AWS

- `AcceptConnection`
- `AssociateIAMRoletoConnection`
- `DeleteConnection`
- `DissassociateIAMRolefromConnection`
- `GetBillingAuthorization`
- `GetConnection`
- `GetPendingConnection`
- `ListConnections`
- `ListIAMRolesforConnection`
- `PutBillingAuthorization`
- `RejectConnection`

CodeCatalyst informazioni in CloudTrail

CloudTrail è abilitato su un Account AWS quando crei quell'account. Quando lo colleghi Account AWS a uno CodeCatalyst spazio, gli eventi relativi a quello spazio che si verificano in quello spazio Account AWS vengono CloudTrail registrati in quell'account AWS. Gli eventi registrabili CodeCatalyst vengono registrati come CloudTrail eventi nei CloudTrail log dell'account connesso e nella cronologia degli eventi nella CloudTrail console, insieme ad altri eventi registrabili AWS in quell'account.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata da un utente con il proprio AWS ID Builder.
- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Accesso agli CloudTrail eventi

Per una registrazione continua degli eventi del tuo Account AWS, compresi gli eventi relativi alle CodeCatalyst attività in Account AWS, crea un percorso. Un trail consente a CloudTrail di fornire i file di log in un bucket S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di registro al bucket S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Un trail è una configurazione che consente la consegna di eventi come file di registro a un bucket S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

Esempio di evento di connessione all' CodeCatalyst account in AWS

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'ListConnectionsazione. Per un Account AWS utente connesso allo spazio, ListConnections viene utilizzato per visualizzare tutte le connessioni dell'account a CodeCatalyst for this Account AWS. L'evento verrà registrato nel file Account AWS specificato e il valore di arn sarà l'Amazon Resource Name (ARN) del ruolo utilizzato per l'azione. accountId

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAI44QH8DHBEXAMPLE",
```

```

    "arn": "role-ARN",
    "accountId": "account-ID",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "role-ARN",
        "accountId": "account-ID",
        "userName": "user-name"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-09-06T15:04:31Z",
        "mfaAuthenticated": "false"
      }
    },
    "eventTime": "2022-09-06T15:08:43Z",
    "eventSource": "account-ID",
    "eventName": "ListConnections",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.168.0.1",
    "userAgent": "aws-cli/1.18.147 Python/2.7.18 Linux/5.4.207-126.363.amzn2int.x86_64
botocore/1.18.6",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "account-ID",
    "eventCategory": "Management"
  }
}

```

Esempio di evento relativo alle risorse CodeCatalyst del progetto in AWS

L'esempio seguente mostra una voce di CloudTrail registro che illustra l>CreateDevEnvironmentazione. Un account di fatturazione collegato allo spazio e Account AWS che rappresenta l'account di fatturazione designato per lo spazio viene utilizzato per eventi a livello di progetto nello spazio, come la creazione di un ambiente di sviluppo.

In `userIdentity`, `accountId` sul campo, si tratta dell'ID account IAM Identity Center (432677196278) che ospita il pool di identità per tutte le identità AWS Builder ID. Questo ID account contiene le seguenti informazioni sull' CodeCatalyst utente dell'evento.

- Il `type` campo indica il tipo di entità IAM per la richiesta. Per CodeCatalyst gli eventi relativi allo spazio e alle risorse del progetto, questo valore è `IdentityCenterUser`. Il `accountId` campo specifica l'account proprietario dell'entità utilizzata per ottenere le credenziali.
- Il `userId` campo contiene l' AWS identificatore Builder ID per l'utente.
- Il `identityStoreArn` campo contiene l'ARN del ruolo per l'account e l'utente dell'archivio di identità.

Il `recipientAccountId` campo contiene l'ID dell'account di fatturazione per lo spazio, con un valore di esempio di 111122223333.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "IdentityCenterUser",
    "accountId": "432677196278",
    "onBehalfOf": {
      "userId": "user-ID",
      "identityStoreArn": "arn:aws:identitystore::432677196278:identitystore/d-9067642ac7"
    },
    "credentialId": "ABCDDefGhiJKLMn11Lmn_1AbCDEFGhIjk-AaBCdEFGHIjKLMnOPqrs11abEXAMPLE"
  },
  "eventTime": "2023-05-18T17:10:50Z",
  "eventSource": "codecatalyst.amazonaws.com",
  "eventName": "CreateDevEnvironment",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.168.0.1",
  "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0",
  "requestParameters": {
    "spaceName": "MySpace",
    "projectName": "MyProject",
    "ides": [{
      "runtime": "public.ecr.aws/q6e8p2q0/cloud9-ide-runtime:2.5.1",
      "name": "Cloud9"
    }],
  },
}
```

```
"instanceType": "dev.standard1.small",
"inactivityTimeoutMinutes": 15,
"persistentStorage": {
  "sizeInGiB": 16
},
},
"responseElements": {
  "spaceName": "MySpace",
  "projectName": "MyProject",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 "
},
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventCategory": "Management"
}
```

Note

In alcuni casi, lo user agent potrebbe non essere noto. In questo caso, CodeCatalyst fornirà il valore di Unknown nel userAgent campo dell' CloudTrail evento.

Interrogazione dei percorsi CodeCatalyst degli eventi

Puoi creare e gestire query per i tuoi CloudTrail log utilizzando una tabella di query in Amazon Athena. Per ulteriori informazioni sulla creazione di una query, consulta la sezione [Interrogazione dei AWS CloudTrail log](#) nella Amazon Athena User Guide.

Accesso agli eventi registrati utilizzando la registrazione degli eventi

Quando gli utenti eseguono azioni in Amazon CodeCatalyst, queste azioni vengono registrate come eventi. Puoi utilizzare il AWS CLI per visualizzare i registri degli eventi in uno spazio in un periodo di tempo specificato. È possibile visualizzare questi eventi per esaminare le azioni intraprese nello spazio, tra cui la data e l'ora delle azioni, il nome dell'utente che ha eseguito l'azione e l'indirizzo IP a cui l'utente ha effettuato la richiesta.

 Note

Gli eventi di gestione di uno CodeCatalyst spazio vengono registrati CloudTrail per l'account di fatturazione connesso. Per ulteriori informazioni sugli eventi di CodeCatalyst gestione registrati da CloudTrail, vedere. [CodeCatalyst informazioni in CloudTrail](#)

Per visualizzare un registro degli eventi per uno spazio, è necessario averlo installato e configurato AWS CLI con un profilo per CodeCatalyst e devi avere il ruolo di amministratore dello spazio per lo spazio. Per ulteriori informazioni, consulta [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#) e [Ruolo di amministratore dello spazio](#).

 Note

Per visualizzare la registrazione degli eventi che si verificano per conto di CodeCatalyst in connected Account AWS o per visualizzare la registrazione degli eventi per lo spazio o le risorse del progetto nell'account di fatturazione connesso, è possibile utilizzare. AWS CloudTrail Per ulteriori informazioni, consulta [Monitoraggio delle chiamate API Account AWS utilizzando la AWS CloudTrail registrazione](#).

1. Apri un terminale o una riga di comando ed esegui il `aws codecatalyst list-event-logs` comando, specificando:
 - Il nome dello spazio con l'opzione `--space-name`.
 - La data e l'ora in cui si desidera iniziare a esaminare gli eventi, nel formato timestamp dell'ora universale coordinata (UTC) come specificato in [RFC 3339](#), con l'opzione. `--start-time`
 - [La data e l'ora in cui desideri interrompere la revisione degli eventi, nel formato timestamp dell'ora universale coordinata \(UTC\) come specificato in RFC 3339, con l'opzione.](#) `--end-time`
 - (Facoltativo) Il numero massimo di risultati da restituire in una singola risposta, con l'opzione. `--max-results` Se il numero di risultati è maggiore del numero specificato, la risposta includerà un `nextToken` elemento che potrai utilizzare per restituire i risultati successivi.
 - (Facoltativo) Limita i risultati a un tipo di evento specifico che desideri venga restituito, con l'opzione. `--event-name`

Questo esempio restituisce gli eventi registrati nello spazio denominato *ExampleCorp* dal periodo di tempo *2022-11-30* al e *2022-12-01* indica che nella risposta viene restituito un massimo di *2* eventi.

```
aws codecatalyst list-event-logs --space-name ExampleCorp --start-time 2022-11-30
--end-time 2022-12-01 --event-name list-event-logs --max-results 2
```

2. Se gli eventi si sono verificati in questo intervallo di tempo, il comando restituisce risultati simili ai seguenti:

```
{
  "nextToken": "EXAMPLE",
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "eventName": "listEventLogs",
      "eventType": "AwsApiCall",
      "eventCategory": "MANAGEMENT",
      "eventSource": "manage",
      "eventTime": "2022-12-01T22:47:24.605000+00:00",
      "operationType": "READONLY",
      "userIdentity": {
        "userType": "USER",
        "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111"
        "userName": "MaryMajor"
      },
      "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "requestPayload": {
        "contentType": "application/json",
        "data": "{\"spaceName\":\"ExampleCorp\",\"startTime\":
        \"2022-12-01T00:00:00Z\",\"endTime\":\"2022-12-10T00:00:00Z\",\"maxResults\":
        \"2\"}"
      },
      "sourceIpAddress": "127.0.0.1",
      "userAgent": "aws-cli/2.9.0 Python/3.9.11 Darwin/21.3.0 exe/x86_64
prompt/off command/codecatalyst.list-event-logs"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
      "eventName": "createProject",
      "eventType": "AwsApiCall",
```

```

    "eventCategory": "MANAGEMENT",
    "eventSource": "manage",
    "eventTime": "2022-12-01T09:15:32.068000+00:00",
    "operationType": "MUTATION",
    "userIdentity": {
      "userType": "USER",
      "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111",
      "userName": "MaryMajor"
    },
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "requestPayload": {
      "contentType": "application/json",
      "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject\", \"displayName\":\"MyFirstProject\"}"
    },
    "responsePayload": {
      "contentType": "application/json",
      "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject\", \"displayName\":\"MyFirstProject\", \"id\":\"a1b2c3d4-5678-90ab-cdef-EXAMPLE4444\"}"
    },
    "sourceIpAddress": "192.0.2.23",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0) Gecko/20100101 Firefox/102.0"
  }
]
}

```

3. Esegui nuovamente il `list-event-logs` comando con l'`--next-token` opzione e il valore del token restituito per recuperare il successivo set di eventi registrati che corrispondono alla richiesta.

Quote per identità, autorizzazione e accesso in CodeCatalyst

La tabella seguente descrive le quote e i limiti per l'identità, l'autorizzazione e l'accesso in Amazon CodeCatalyst. Per ulteriori informazioni sulle quote in Amazon CodeCatalyst, consulta [Quote per CodeCatalyst](#).

Risorsa	Informazioni
Alias in CodeCatalyst	Qualsiasi combinazione di caratteri consentiti compresa tra 3 e 100 caratteri di lunghezza

Risorsa	Informazioni
	deve iniziare con una lettera. Caratteri validi: A-Z, a-z e 0-9. Gli alias non possono: • contenere meno di 3 caratteri • contenere spazi o uno qualsiasi dei seguenti caratteri: ? ^ * [\ ~ :
Numero massimo di inviti inviati da un utente al giorno	500
Numero massimo di inviti inviati a un indirizzo e-mail al giorno	25
Numero massimo di token di accesso personali (PAT) per utente	100
Numero massimo di connessioni personali per ogni identità utente (CodeCatalystalias) in tutti gli spazi, per tipo di provider	1
Password in CodeCatalyst	Qualsiasi combinazione di caratteri consentiti tra 8 e 64 caratteri di lunghezza. Caratteri validi: A-Z, a-z e 0-9. La password può includere i seguenti caratteri non alfanumerici: (~ ! @ # \$ % ^ & * _ - + = ` \ { } [] : ; " ' < > , . ? /)
Nomi PAT in CodeCatalyst	Qualsiasi combinazione di caratteri consentiti compresa tra 1 e 100 caratteri
Manca poco alla scadenza dell'invito di un membro del progetto	Scade dopo 24 ore
Manca poco alla scadenza dell'invito di un membro dello spazio	Scade dopo 24 ore

Risorsa	Informazioni
Manca poco alla scadenza della verifica dell'indirizzo e-mail	Scade 10 minuti dopo l'invio

Risoluzione dei problemi

Questa sezione può aiutarti a risolvere alcuni problemi comuni che potresti riscontrare durante l'accesso al tuo profilo Amazon CodeCatalyst .

Problemi di registrazione

Potresti riscontrare alcuni problemi durante la registrazione. Abbiamo alcune soluzioni.

Il mio indirizzo email è già in uso

Se l'e-mail che hai inserito è già in uso e la riconosci come tua, allora potresti già avere un profilo con noi. Accedi con questa identità esistente. Se non possiedi l'e-mail esistente, registrati con un'altra e-mail inutilizzata.

Non è possibile completare la verifica dell'e-mail

Se non hai ricevuto l'email di verifica

1. Controlla la cartella dello spam, della posta indesiderata e degli elementi eliminati.

Note

Questa e-mail di verifica proviene dall'indirizzo `no-reply@signin.aws` o `no-reply@login.awsapps.com`. Ti consigliamo di configurare il tuo sistema di posta in modo che accetti le e-mail da questi indirizzi del mittente e non le gestisca come posta indesiderata o spam.

2. Attendi 5 minuti e aggiorna la tua casella di posta. Controlla di nuovo la cartella dello spam, della posta indesiderata e della posta eliminata.
3. Se continui a non visualizzare l'email di verifica, scegli Reinvia il codice. Se sei già uscito da quella pagina, riavvia il flusso di lavoro per la registrazione ad [Amazon CodeCatalyst](#).

La mia password non soddisfa i requisiti minimi

Per motivi di sicurezza, la password deve includere 8-20 caratteri, lettere maiuscole e minuscole e numeri.

Problemi di accesso

Ho dimenticato la mia password

Segui la procedura riportata in [Ho dimenticato la mia password](#).

La mia password non funziona

È necessario rispettare questi requisiti ogni volta che si imposta o si modifica la password:

- Le password distinguono tra maiuscole e minuscole.
- Le password devono avere una lunghezza compresa tra 8 e 64 caratteri con lettere maiuscole e minuscole, numeri e almeno un carattere non alfanumerico.
- Non puoi riutilizzare le ultime tre password .

Non riesco ad abilitare l'MFA

Per abilitare l'autenticazione a più fattori, aggiungi uno o più dispositivi MFA al tuo profilo seguendo i passaggi riportati in [Configura il tuo AWS Builder ID per accedere con l'autenticazione a più fattori \(MFA\)](#).

Non riesco ad aggiungere un dispositivo MFA

Se non riesci ad aggiungere un altro dispositivo MFA, è possibile che sia stato raggiunto il limite di dispositivi MFA che puoi registrare. Potrebbe essere necessario rimuovere un dispositivo MFA esistente prima di aggiungerne uno nuovo.

Non riesco a rimuovere un dispositivo MFA

Se desideri disabilitare l'autenticazione a più fattori, procedi con la rimozione del dispositivo MFA seguendo i passaggi riportati in [Eliminazione di un dispositivo MFA](#). Tuttavia, se desideri lasciare l'MFA abilitata, prima di provare a eliminare un dispositivo MFA esistente dovrai aggiungere un altro dispositivo MFA. Per ulteriori informazioni sull'aggiunta di un altro dispositivo MFA, consulta [Come registrare un dispositivo per l'utilizzo con l'autenticazione a più fattori](#).

Problemi con la disconnessione

Non riesco a trovare dove disconnettermi

Nell'angolo in alto a destra della pagina, scegli Esci.

La disconnessione non mi disconnette completamente

Il sistema è progettato per disconnettersi immediatamente, ma la disconnessione completa può richiedere fino a un'ora.

Errore «I get a role does not exist» per un flusso di lavoro non riuscito

Problema: dopo aver creato un progetto dall'applicazione Web o da un blueprint serverless, il flusso di lavoro ha esito negativo e viene visualizzato il seguente errore:

CLIENT_ERROR: Il ruolo non esiste

Soluzione possibile: dopo aver configurato un ruolo IAM con le autorizzazioni per eseguire il flusso di lavoro e aver aggiunto il ruolo IAM al flusso di lavoro YAML, il flusso di lavoro continua a fallire perché potrebbe essere necessario aggiungere il ruolo IAM alla connessione dell'account. Aggiungi il ruolo IAM alla connessione dell'account per il tuo spazio come descritto in [Aggiungere ruoli IAM alle connessioni degli account](#)

Ricevo un errore di ruolo a causa di un flusso di lavoro non riuscito

Problema: dopo aver creato un progetto dall'applicazione Web o da un blueprint serverless, il flusso di lavoro ha esito negativo e viene visualizzato il seguente errore:

CLIENT_ERROR: Il ruolo non è impostato correttamente o non esiste

Soluzione possibile: lo spazio in cui è stato creato il progetto potrebbe richiedere la configurazione di una Account AWS connessione o il completamento di una richiesta di connessione all'account. Se il tuo spazio ha già una Account AWS connessione attiva, crea e aggiungi un ruolo IAM con autorizzazioni per eseguire azioni del flusso di lavoro. Aggiungi il ruolo IAM alla connessione del tuo account come descritto in [Aggiungere ruoli IAM alle connessioni degli account](#).

Soluzione possibile: se il progetto è stato creato senza specificare una connessione, la connessione all'account deve essere associata all'ambiente di distribuzione. Se il tuo spazio ha già una Account

AWS connessione attiva e un ruolo IAM aggiunto, devi aggiungere la connessione dell'account con il ruolo IAM al tuo ambiente di distribuzione come descritto in [Aggiungere la connessione all'account e i ruoli IAM all'ambiente di distribuzione](#).

Devo aggiornare il ruolo IAM nel flusso di lavoro di un progetto

Se la Account AWS connessione è configurata completamente e il ruolo IAM viene creato e aggiunto alla connessione dell'account, puoi aggiornare il ruolo IAM nel flusso di lavoro del progetto.

1. Scegli l'opzione CI/CD e scegli il tuo flusso di lavoro. Scegli il pulsante YAML.
2. Scegli Modifica.
3. Sul `ActionRoleArn`: campo, sostituisci l'ARN del ruolo IAM con l'ARN del ruolo IAM aggiornato. Scegli Convalida.
4. Scegli Applica.

Il flusso di lavoro si avvia automaticamente se si trova sul ramo principale. Altrimenti, per rieseguire il flusso di lavoro, scegli Esegui.

Ho una richiesta di revisione per il mio GitHub account dopo aver creato una connessione personale

Dopo aver creato una connessione personale a GitHub, l' CodeCatalyst app viene installata per il tuo GitHub account come GitHub app. Se alcune risorse CodeCatalyst richiedono autorizzazioni di lettura o scrittura aggiornate, potresti dover accedere al tuo GitHub account per aggiornare le autorizzazioni sull'app installata.

1. Accedi GitHub e accedi alle impostazioni del tuo account per le app installate. Scegli l'icona del tuo profilo, scegli Impostazioni, quindi scegli Applicazioni.
2. Nella scheda GitHub App installate, nell'elenco delle applicazioni installate, visualizza l'app installata per CodeCatalyst. Se sono disponibili le autorizzazioni per la revisione, verrà visualizzato un link per la richiesta di revisione.
3. Scegli il link, quindi conferma le tue credenziali quando richiesto. Inserisci le tue credenziali e scegli Verifica.
4. Accetta le nuove autorizzazioni, indica i repository a cui desideri applicare le autorizzazioni, quindi scegli Salva.

Come posso compilare un modulo di supporto?

Puoi andare [su Amazon CodeCatalyst](#) o compilare un [modulo Support Feedback](#). Nella sezione Richiedi informazioni, sotto Come possiamo aiutarti, inserisci che sei un CodeCatalyst cliente Amazon. Fornisci il maggior numero di dettagli possibile in modo che possiamo risolvere il problema nel modo più efficiente.

Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst

Amazon CodeCatalyst include estensioni che ti aiutano ad aggiungere funzionalità e integrarti con prodotti esterni a CodeCatalyst. Con le estensioni del CodeCatalyst catalogo, i team possono personalizzare le proprie esperienze in CodeCatalyst.

Argomenti

- [Estensioni di terze parti disponibili](#)
- [Concetti di estensioni](#)
- [Guida rapida: installazione di estensioni, connessione di provider e collegamento di risorse in CodeCatalyst](#)
- [Installazione di un'estensione in uno spazio](#)
- [Disinstallazione di un'estensione in uno spazio](#)
- [Connessione di GitHub account, aree di lavoro Bitbucket, GitLab utenti e siti Jira CodeCatalyst](#)
- [Disconnessione di GitHub account, aree di lavoro Bitbucket, GitLab utenti e siti Jira CodeCatalyst](#)
- [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#)
- [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#)
- [Visualizzazione di repository di terze parti e ricerca dei problemi di Jira in CodeCatalyst](#)
- [Avvio automatico di un flusso di lavoro eseguito dopo eventi di repository di terze parti](#)
- [Limitazione dell'accesso IP con provider di repository di terze parti](#)
- [Bloccare le fusioni di terze parti quando i flussi di lavoro falliscono](#)
- [Collegamento dei problemi di Jira alle richieste pull CodeCatalyst](#)
- [Visualizzazione CodeCatalyst degli eventi nei problemi di Jira](#)

Estensioni di terze parti disponibili

È possibile aggiungere funzionalità particolari al CodeCatalyst progetto a seconda dell'estensione con cui si sceglie di integrare le risorse.

Integrazione dei GitHub repository in CodeCatalyst

GitHub è un servizio basato sul cloud che aiuta gli sviluppatori a memorizzare e gestire il proprio codice. L'estensione GitHub repositories ti consente di utilizzare GitHub repository collegati nei progetti Amazon CodeCatalyst . Puoi anche collegare i GitHub repository quando crei un nuovo progetto. CodeCatalyst Per ulteriori informazioni, consulta [Creazione di un progetto con un repository di terze parti collegato](#).

Note

- Non puoi usare GitHub repository vuoti o archiviati con i progetti. CodeCatalyst
- L'estensione dei GitHub repository non è compatibile con gli archivi di GitHub Enterprise Server.

Dopo aver installato e configurato l'estensione dei GitHub repository, sarai in grado di:

- Visualizza i tuoi GitHub repository nell'elenco dei repository di origine in CodeCatalyst
- Archivia e gestisci i file di definizione del flusso di lavoro nei tuoi repository GitHub
- Crea, leggi, aggiorna ed elimina i file archiviati in GitHub repository collegati da CodeCatalyst Dev Environments
- Archivia e indicizza i file dai GitHub repository collegati in CodeCatalyst
- Crea CodeCatalyst progetti con archivi esistenti di account collegati GitHub
- Crea un GitHub repository con codice generato da un blueprint quando crei un progetto con un blueprint o aggiungi un blueprint
- Il CodeCatalyst flusso di lavoro Start viene eseguito automaticamente quando il codice viene inviato a GitHub repository collegati o quando le richieste pull vengono create, modificate o chiuse in repository collegati GitHub
- Utilizza i file sorgente dei GitHub repository collegati nei flussi di lavoro CodeCatalyst
- Leggi ed esegui GitHub azioni nei flussi di lavoro CodeCatalyst
- Invia gli stati CodeCatalyst di esecuzione del flusso di lavoro agli GitHub archivi collegati e blocca le fusioni delle GitHub pull request in base agli stati di commit

Integrazione dei repository Bitbucket in CodeCatalyst

Bitbucket è un servizio basato sul cloud che aiuta gli sviluppatori a memorizzare e gestire il proprio codice. L'estensione dei repository Bitbucket consente di utilizzare repository Bitbucket collegati nei progetti Amazon. CodeCatalyst Puoi anche collegare i repository Bitbucket quando crei un nuovo progetto. CodeCatalyst Per ulteriori informazioni, consulta [Creazione di un progetto con un repository di terze parti collegato](#).

Note

- Non puoi utilizzare repository Bitbucket vuoti o archiviati con i progetti. CodeCatalyst
- L'estensione Bitbucket repositories non è compatibile con i repository Bitbucket Data Center.

Dopo aver installato e configurato l'estensione dei repository Bitbucket, sarai in grado di:

- Visualizza i tuoi repository Bitbucket nell'elenco dei repository di origine in CodeCatalyst
- Archivia e gestisci i file di definizione del flusso di lavoro nei tuoi repository Bitbucket.
- Crea, leggi, aggiorna ed elimina i file archiviati nei repository Bitbucket collegati da Dev Environments CodeCatalyst
- Crea CodeCatalyst progetti con repository esistenti di account Bitbucket collegati
- Archivia e indicizza i file dai repository Bitbucket collegati in CodeCatalyst
- Crea un repository Bitbucket con codice generato da un blueprint quando crei un progetto con un blueprint o aggiungi un blueprint
- Il CodeCatalyst flusso di lavoro Start viene eseguito automaticamente quando il codice viene inviato a repository Bitbucket collegati o quando le richieste pull vengono create, modificate o chiuse in repository Bitbucket collegati
- Usa i file sorgente del tuo repository Bitbucket collegato nei flussi di lavoro CodeCatalyst
- Invia gli stati CodeCatalyst di esecuzione del flusso di lavoro ai repository Bitbucket collegati e blocca le fusioni delle pull request di Bitbucket in base agli stati di commit

GitLab Integrazione dei repository in CodeCatalyst

GitLab è un servizio basato sul cloud che aiuta gli sviluppatori a memorizzare e gestire il proprio codice. L'estensione GitLab repositories ti consente di utilizzare repository di GitLab progetti collegati nei progetti Amazon CodeCatalyst . Puoi anche collegare gli archivi dei GitLab progetti quando crei un nuovo progetto. CodeCatalyst Per ulteriori informazioni, consulta [Creazione di un progetto con un repository di terze parti collegato](#).

Note

- Non puoi utilizzare repository di progetti vuoti o archiviati con i GitLab progetti. CodeCatalyst
- L'estensione GitLab repositories non è compatibile con GitLab i repository autogestiti.

Dopo aver installato e configurato l'estensione dei GitLab repository, sarai in grado di:

- Visualizza i repository GitLab del tuo progetto nell'elenco dei repository di origine in CodeCatalyst
- Archivia e gestisci i file di definizione del flusso di lavoro negli archivi dei tuoi GitLab progetti.
- Crea, leggi, aggiorna ed elimina i file archiviati nei repository di GitLab progetti collegati da CodeCatalyst Dev Environments
- Crea CodeCatalyst progetti con archivi esistenti di utenti connessi GitLab
- Archivia e indicizza i file dagli archivi di GitLab progetti collegati in CodeCatalyst
- Crea un archivio di GitLab progetti con codice generato da un blueprint quando crei un progetto con un blueprint o aggiungi un blueprint
- Il CodeCatalyst flusso di lavoro Start viene eseguito automaticamente quando il codice viene inviato agli archivi di GitLab progetti collegati o quando le richieste pull vengono create, modificate o chiuse in archivi di progetti collegati GitLab
- Usa i file sorgente del repository di GitLab progetto collegato nei flussi di lavoro CodeCatalyst
- Invia gli stati CodeCatalyst di esecuzione del flusso di lavoro agli archivi di GitLab progetto collegati e blocca le richieste di GitLab unione in base agli stati di commit

Integrazione dei problemi di Jira in CodeCatalyst

Jira è un'applicazione software che aiuta i team di sviluppo agili a pianificare, assegnare, tracciare, segnalare e gestire il lavoro. L'estensione Jira Software ti consente di utilizzare i progetti Jira nei progetti Amazon CodeCatalyst .

Note

CodeCatalyst è compatibile solo con Jira Software Cloud.

Dopo aver installato e configurato l'estensione Jira Software per un CodeCatalyst progetto Amazon, sarai in grado di:

- Accedi ai progetti Jira da CodeCatalyst collegandoli ai progetti CodeCatalyst
- Aggiorna i problemi di Jira con le pull request CodeCatalyst
- Visualizza lo stato e le esecuzioni del flusso di lavoro delle richieste CodeCatalyst pull collegate nei problemi di Jira

Concetti di estensioni

Ecco alcuni concetti e termini da conoscere quando si utilizzano le estensioni in CodeCatalyst.

Estensioni

Un'estensione è un componente aggiuntivo che puoi installare nel tuo CodeCatalyst spazio per aggiungere nuove funzionalità ai tuoi progetti e integrarlo con servizi esterni. CodeCatalyst Le estensioni possono essere sfogliate e installate dal CodeCatalyst catalogo.

CodeCatalyst catalogo

Il CodeCatalyst catalogo è un elenco centralizzato di tutte le estensioni disponibili in CodeCatalyst. Puoi sfogliare il CodeCatalyst catalogo per trovare estensioni che possono migliorare l'esperienza del tuo team in aree CodeCatalyst come i sorgenti, i flussi di lavoro e altro ancora.

Connessione e collegamento

A seconda delle risorse di terze parti che desideri utilizzare o gestire, devi connettere il tuo GitHub account, l'area di lavoro Bitbucket o il progetto Jira. Quindi, devi collegare il tuo GitHub repository, il repository Bitbucket o il progetto Jira al tuo progetto. CodeCatalyst

- GitHub repository: collega l' GitHub account e quindi collega i GitHub repository.
- Archivi Bitbucket: collega l'area di lavoro Bitbucket e quindi collega i repository Bitbucket.
- GitLab repository: Connetti GitLab l'utente e quindi collega gli archivi GitLab del progetto.
- Jira Software: collega il sito Jira e poi collega i progetti Jira.

Guida rapida: installazione di estensioni, connessione di provider e collegamento di risorse in CodeCatalyst

Questo tutorial fornisce una panoramica delle tre attività seguenti:

1. Installa i GitHub repository, i repository Bitbucket, i repository o l'estensione Jira GitLab Software. In un sito esterno ti viene richiesto di connetterti e fornire l'accesso alle tue risorse CodeCatalyst di terze parti, operazione che viene eseguita come parte del passaggio successivo.

Important

Per installare i GitHub repository, i repository Bitbucket, i GitLab repository o l'estensione Jira Software nel tuo CodeCatalyst spazio, devi aver effettuato l'accesso con un account con il ruolo di amministratore dello spazio.

2. Connect il tuo GitHub account, l'area di lavoro Bitbucket, GitLab l'utente o il sito Jira a. CodeCatalyst

Important

Per connettere il tuo GitHub account, l'area di lavoro Bitbucket, GitLab l'utente o il sito Jira al tuo CodeCatalyst spazio, devi essere sia l'amministratore della fonte di terze parti che l'amministratore dello spazio. CodeCatalyst

 Important

Dopo aver installato un'estensione del repository, il codice di tutti gli archivi a cui ti colleghi CodeCatalyst verrà indicizzato e archiviato. CodeCatalyst Ciò renderà il codice ricercabile in. CodeCatalyst Per comprendere meglio la protezione dei dati per il tuo codice quando utilizzi repository collegati in CodeCatalyst, consulta la sezione [Protezione dei dati](#) nella Amazon CodeCatalyst User Guide.

 Note

Se utilizzi una connessione a un GitHub account, devi creare una connessione personale per stabilire una mappatura delle identità tra la tua CodeCatalyst identità e la tua identità. GitHub Per ulteriori informazioni, consultare [Connessioni personali](#) e [Accesso alle GitHub risorse con connessioni personali](#).

3. Collega il tuo GitHub repository, il repository Bitbucket, l'archivio del progetto o il GitLab progetto Jira al tuo progetto. CodeCatalyst

 Important

- Sebbene tu possa collegare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto come collaboratore, puoi scollegare un repository di terze parti solo come amministratore dello spazio o amministratore del progetto. Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).
- Per collegare il tuo progetto Jira al tuo progetto, devi essere l'amministratore di Space o l'amministratore CodeCatalyst del progetto. CodeCatalyst CodeCatalyst

 Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo CodeCatalyst, modificare il ramo predefinito e quindi

collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

È consigliabile assicurarsi sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Note

- Un GitHub repository, un repository Bitbucket o un repository di GitLab progetto possono essere collegati a un solo progetto in uno spazio. CodeCatalyst
- Non puoi utilizzare repository vuoti o archiviati, repository Bitbucket o GitHub repository di progetti con progetti. GitLab CodeCatalyst
- Non puoi collegare un GitHub repository, un repository Bitbucket o un repository di progetti con lo stesso nome di un repository in un GitLab progetto. CodeCatalyst
- L'estensione GitHub repositories non è compatibile con gli archivi di Enterprise Server. GitHub
- L'estensione Bitbucket repositories non è compatibile con i repository Bitbucket Data Center.
- L'estensione GitLab repositories non è compatibile con gli archivi di progetto autogestiti. GitLab
- Non puoi utilizzare le funzioni Scrivi descrizione per me o Riepiloga commenti con gli archivi collegati. Queste funzionalità sono disponibili solo nelle pull request in. CodeCatalyst
- Un CodeCatalyst progetto può essere collegato a un solo progetto Jira. Un progetto Jira può essere collegato a più CodeCatalyst progetti.

Puoi anche installare i GitHub repository, i repository Bitbucket, l'estensione dei repository, connetterti al tuo GitHub account, all'area di lavoro Bitbucket o all' GitLab utente e collegare GitLab repository di terze parti durante la creazione di un nuovo progetto. CodeCatalyst Per ulteriori informazioni, consulta [Creazione di un progetto con un repository di terze parti collegato](#).

Argomenti

- [Passaggio 1: installa un'estensione di terze parti dal catalogo CodeCatalyst](#)

- [Passaggio 2: Collega il tuo provider di terze parti al tuo CodeCatalyst spazio](#)
- [Passaggio 3: collega le risorse di terze parti al tuo progetto CodeCatalyst](#)
- [Fasi successive](#)

Passaggio 1: installa un'estensione di terze parti dal catalogo CodeCatalyst

Il primo passaggio per utilizzare risorse di terze parti in CodeCatalyst consiste nell'installare l'estensione dei GitHub repositories dal catalogo. CodeCatalyst Per installare l'estensione, effettuate le seguenti operazioni, scegliendo l'estensione per le risorse di terze parti che desiderate utilizzare. GitHub repository, i repository Bitbucket e i repository consentono di utilizzare GitLab repository, repository Bitbucket o GitHub repository di progetti in. GitLab CodeCatalyst Jira Software ti consente di gestire i problemi di Jira in. CodeCatalyst

Per installare un'estensione dal catalogo CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.
3. Vai al CodeCatalyst CodeCatalyst catalogo scegliendo l'icona del catalogo



nel menu in alto. Puoi cercare GitHub repository, repository Bitbucket, GitLab repository o Jira Software. Puoi anche filtrare le estensioni in base alle categorie.

4. (Facoltativo) Per visualizzare ulteriori dettagli sull'estensione, ad esempio le autorizzazioni che avrà l'estensione, scegli il nome dell'estensione.
5. Scegli Installa. Controlla le autorizzazioni richieste dall'estensione e, se vuoi continuare, scegli nuovamente Installa.

Dopo aver installato l'estensione, verrai indirizzato alla pagina dei dettagli dell'estensione. A seconda dell'estensione installata, è possibile visualizzare e gestire i provider connessi e le risorse collegate.

Passaggio 2: Collega il tuo provider di terze parti al tuo CodeCatalyst spazio

Dopo aver installato i GitHub repository, i repository Bitbucket, i repository o l'estensione Jira Software, il passaggio successivo consiste nel connettere il GitHub account, l'area di lavoro Bitbucket, il GitLab repository del GitLab progetto o il sito Jira al tuo spazio. CodeCatalyst

Per connettere il tuo account, l'area di lavoro Bitbucket o il sito Jira a GitHub CodeCatalyst

- Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai installato:
 - GitHub repository: Connect a un GitHub account.
 1. Nella scheda GitHub Account connessi, scegli Connetti GitHub account per accedere al sito esterno di GitHub.
 2. Accedi al tuo GitHub account utilizzando GitHub le tue credenziali, quindi scegli l'account in cui desideri installare Amazon CodeCatalyst.

 Tip

Se in precedenza hai collegato un GitHub account allo spazio, non ti verrà richiesto di autorizzarlo nuovamente. Vedrai invece una finestra di dialogo che ti chiede dove desideri installare l'estensione se sei membro o collaboratore di più di GitHub un'organizzazione, o la pagina di configurazione per l' CodeCatalyst applicazione Amazon se appartieni a una sola GitHub organizzazione. Configura l'applicazione per l'accesso al repository che desideri consentire, quindi scegli Salva. Se il pulsante Salva non è attivo, apporta una modifica alla configurazione, quindi riprova.

3. Scegli se consentire l'accesso CodeCatalyst a tutti i repository attuali e futuri o scegli i GitHub repository specifici in cui desideri utilizzare. CodeCatalyst L'opzione predefinita prevede l'inclusione di tutti i GitHub repository nell' GitHub account, compresi i repository futuri a cui sarà possibile accedere da. CodeCatalyst
4. Controlla le autorizzazioni concesse a CodeCatalyst, quindi scegli Installa.

Dopo aver collegato il tuo GitHub account a CodeCatalyst, verrai reindirizzato alla pagina dei dettagli dell'estensione dei GitHub repository, dove puoi visualizzare e gestire GitHub gli account connessi e gli archivi collegati GitHub .

- Archivi Bitbucket: Connettiti a uno spazio di lavoro Bitbucket.
 1. Nella scheda Connected Bitbucket workspace, scegli Connect Bitbucket workspace per accedere al sito esterno di Bitbucket.
 2. Accedi al tuo spazio di lavoro Bitbucket utilizzando le tue credenziali Bitbucket e controlla le autorizzazioni concesse a. CodeCatalyst

3. Dal menu a discesa Autorizza per l'area di lavoro, scegli l'area di lavoro Bitbucket a cui desideri fornire l'accesso, quindi scegli Concedi l'accesso. CodeCatalyst

 Tip

Se in precedenza hai collegato un'area di lavoro Bitbucket allo spazio, non ti verrà richiesto di autorizzare nuovamente. Vedrai invece una finestra di dialogo che ti chiede dove desideri installare l'estensione se sei un membro o collaboratore in più di uno spazio di lavoro Bitbucket, o la pagina di configurazione per l' CodeCatalyst applicazione Amazon se appartieni a un solo spazio di lavoro Bitbucket. Configura l'applicazione per l'accesso allo spazio di lavoro che desideri consentire, quindi scegli Concedi l'accesso. Se il pulsante Concedi l'accesso non è attivo, apporta una modifica alla configurazione, quindi riprova.

Dopo aver collegato l'area di lavoro Bitbucket a CodeCatalyst, verrai indirizzato alla pagina dei dettagli dell'estensione dei repository Bitbucket, dove puoi visualizzare e gestire le aree di lavoro Bitbucket connesse e gli archivi Bitbucket collegati.

- GitLab repository: Connect a un GitLab utente.
 1. Scegli Connect GitLab user per accedere al sito esterno per GitLab.
 2. Accedi al tuo GitLab utente utilizzando GitLab le tue credenziali e controlla le autorizzazioni concesse a. CodeCatalyst

 Tip

Se in precedenza hai collegato un GitLab utente allo spazio, non ti verrà richiesto di autorizzarlo nuovamente. Verrai invece reindirizzato alla console. CodeCatalyst

3. Scegli Authorize AWS Connector per. GitLab

Dopo aver collegato GitLab l'utente a CodeCatalyst, verrai indirizzato alla pagina dei dettagli GitLab delle estensioni dei repository, dove puoi visualizzare e gestire gli archivi connessi GitLab degli utenti e dei GitLab progetti collegati.

- Jira Software: Connect un sito Jira.

1. Nella scheda Siti Jira connessi, scegli Connetti sito Jira per accedere al sito esterno per Atlassian Marketplace.
2. Scegli Scarica subito per iniziare l'installazione CodeCatalyst sul tuo sito Jira.

 Note

Se l'hai già installato CodeCatalyst sul tuo sito Jira, riceverai una notifica. Scegli Inizia per passare alla fase finale.

3. A seconda del ruolo, esegui una delle seguenti operazioni:
 1. Se sei un amministratore del sito Jira, dal menu a discesa del sito, scegli il sito Jira per installare l' CodeCatalyst applicazione, quindi scegli Installa app.

 Note

Se hai un sito Jira, questo passaggio non verrà visualizzato e verrai indirizzato automaticamente al passaggio successivo.

2. a. Se non sei un amministratore di Jira, dal menu a discesa del sito, scegli il sito Jira per installare l' CodeCatalyst applicazione, quindi scegli Richiedi app. Per ulteriori informazioni sull'installazione delle app Jira, consulta [Chi può installare le app?](#) .
 - b. Inserisci il motivo dell'installazione CodeCatalyst nel campo di testo di input o mantieni il testo predefinito, quindi scegli Invia richiesta.
4. Controlla le azioni eseguite al CodeCatalyst momento dell'installazione dell'applicazione, quindi scegli Scarica ora.
5. Dopo aver installato l'applicazione, scegli Return CodeCatalyst to per tornare a CodeCatalyst.

Dopo aver collegato il tuo sito Jira a CodeCatalyst, puoi visualizzare il sito connesso nella scheda Siti Jira connessi della pagina dei dettagli dell'estensione di Jira Software.

Passaggio 3: collega le risorse di terze parti al tuo progetto CodeCatalyst

Il terzo e ultimo passaggio per utilizzare i tuoi GitHub repository, i repository Bitbucket o gli repository di GitLab progetto o per gestire i problemi di Jira CodeCatalyst consiste nel collegarli al CodeCatalyst progetto in cui desideri utilizzarli.

Per collegare un GitHub repository, un repository Bitbucket, un repository di progetto o un progetto Jira a GitLab un progetto dalla pagina dei dettagli dell'estensione CodeCatalyst

- Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai installato e del provider che hai collegato:
 - GitHub archivi: collega un GitHub archivio.
 1. Nella scheda GitHub Archivi collegati, scegli Collega repository. GitHub
 2. Dal menu a discesa dell'GitHub account, scegli l' GitHub account che contiene il repository che desideri collegare.
 3. Dal menu a discesa del GitHub repository, scegli il repository che desideri collegare a un progetto. CodeCatalyst

Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto nello spazio.

4. (Facoltativo) Se non vedi un GitHub repository nell'elenco dei repository, è possibile che non sia stato configurato per l'accesso al repository nell'applicazione Amazon CodeCatalyst in. GitHub Puoi configurare quali GitHub repository possono essere utilizzati CodeCatalyst nell'account connesso.
 - a. Accedi al tuo [GitHub](#) account, scegli Impostazioni, quindi scegli Applicazioni.
 - b. Nella scheda GitHub App installate, scegli Configura per l' CodeCatalyst applicazione Amazon.
 - c. Effettua una delle seguenti operazioni per configurare l'accesso ai GitHub repository a cui desideri collegarti: CodeCatalyst
 - Per fornire l'accesso a tutti i repository attuali e futuri, scegli Tutti i repository.

- Per consentire l'accesso a repository specifici, seleziona solo repository, scegli il menu a discesa Seleziona repository, quindi scegli un repository a cui desideri consentire il collegamento. CodeCatalyst
5. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto a cui vuoi collegare il repository. GitHub
 6. Scegliere Link (Collegamento).

Se non desideri più utilizzare un GitHub repository CodeCatalyst, puoi scollegarlo da un progetto. CodeCatalyst Quando un repository non è collegato, gli eventi in quel repository non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

- Archivi Bitbucket: collega un repository Bitbucket.
1. Nella scheda Archivi Bitbucket collegati, scegli Collega repository Bitbucket.
 2. Dal menu a discesa Bitbucket workspace, scegli l'area di lavoro Bitbucket che contiene il repository che desideri collegare.
 3. Dal menu a discesa del repository Bitbucket, scegli il repository che desideri collegare a un progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto nello spazio.

4. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto a cui vuoi collegare il repository Bitbucket.
5. Scegliere Link (Collegamento).

Se non desideri più utilizzare un repository Bitbucket in CodeCatalyst, puoi scollegarlo da un progetto. CodeCatalyst Quando un repository non è collegato, gli eventi in quel repository non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

- GitLab repository: collega un archivio di progetto. GitLab

1. Nella scheda Archivi di GitLab progetti collegati, scegli Collega GitLab archivio di progetti.
2. Dal menu a discesa GitLab degli utenti, scegli l' GitLab utente che contiene il repository che desideri collegare.
3. Dal menu a discesa GitLab del repository del progetto, scegli il repository che desideri collegare a un progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto nello spazio.

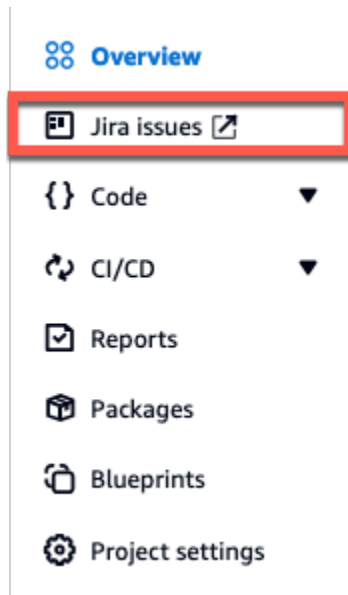
4. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto a cui vuoi collegare l'archivio del GitLab progetto.
5. Scegliere Link (Collegamento).

Se non desideri più utilizzare un repository di GitLab progetto CodeCatalyst, puoi scollegarlo da un progetto. CodeCatalyst Quando un repository di progetto non è collegato, gli eventi in quel repository di progetto non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository di progetto con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

- Jira Software: collega un progetto Jira.

1. Nella scheda Progetti Jira collegati, scegli Collega un progetto Jira.
2. Dal menu a discesa del sito Jira, scegli il sito Jira che contiene il progetto che desideri collegare.
3. Dal menu a discesa del progetto Jira, scegli il progetto che desideri collegare a un progetto. CodeCatalyst
4. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto che desideri collegare a un progetto Jira.
5. Scegliere Link (Collegamento).

Una volta collegato un progetto Jira a un CodeCatalyst progetto, l'accesso ai CodeCatalyst problemi viene completamente disabilitato e i problemi nel pannello di CodeCatalyst navigazione verranno sostituiti con un elemento Jira issues che si collega al progetto Jira.



Se non desideri più utilizzare un progetto Jira in CodeCatalyst, puoi scollegarlo dal tuo progetto. CodeCatalyst Quando un progetto Jira non è collegato, i problemi di Jira non saranno disponibili nel CodeCatalyst progetto e Issues sarà nuovamente il CodeCatalyst fornitore dei problemi. Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Puoi anche collegare il tuo GitHub repository, il repository Bitbucket o il repository del progetto a un GitLab progetto dai repository Source in Code. Per ulteriori informazioni, consulta [Collegamento di risorse da fornitori di terze parti collegati](#).

Fasi successive

Dopo aver installato i GitHub repository, i repository Bitbucket o l'estensione dei repository, aver collegato il fornitore di risorse e aver collegato i GitLab repository di terze parti ai tuoi progetti, puoi utilizzarli nei flussi di lavoro e negli ambienti di sviluppo. CodeCatalyst CodeCatalyst Puoi anche creare repository di terze parti nell' GitHub account connesso, nell'area di lavoro Bitbucket o nell'utente con codice generato da un blueprint. GitLab Per ulteriori informazioni, consultare [Avvio automatico di un flusso di lavoro eseguito dopo eventi di repository di terze parti](#) e [Creazione di un ambiente di sviluppo](#).

Dopo aver installato l'estensione Jira Software, collegato il tuo sito Jira, collegato i tuoi progetti Jira al tuo progetto e collegato una pull request, gli aggiornamenti provenienti da si riflettono nel tuo CodeCatalyst progetto Jira. CodeCatalyst Per ulteriori informazioni sul collegamento delle pull request ai problemi di Jira, consulta. [Collegamento dei problemi di Jira alle richieste pull CodeCatalyst](#) Per ulteriori informazioni sulla visualizzazione CodeCatalyst degli eventi in Jira, consulta. [Visualizzazione CodeCatalyst degli eventi nei problemi di Jira](#)

Installazione di un'estensione in uno spazio

Puoi installare estensioni per il tuo CodeCatalyst spazio che aggiungono funzionalità ai progetti in quello spazio. È possibile visualizzare il CodeCatalyst catalogo scegliendo l'icona

Catalogo 

Per ulteriori informazioni sulle estensioni e sulle relative funzionalità, consulta [Estensioni di terze parti disponibili](#).

Important

Per installare un'estensione, devi aver effettuato l'accesso con un account che abbia il ruolo di amministratore dello spazio nello spazio.

Important

Dopo aver installato un'estensione del repository, il codice di tutti gli archivi a cui ti colleghi CodeCatalyst verrà indicizzato e archiviato. CodeCatalyst Ciò renderà il codice ricercabile in. CodeCatalyst Per comprendere meglio la protezione dei dati per il tuo codice quando utilizzi repository collegati in CodeCatalyst, consulta la sezione [Protezione dei dati](#) nella Amazon CodeCatalyst User Guide.

Per installare un'estensione dal catalogo CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.
3. Vai al CodeCatalyst catalogo scegliendo l'icona del catalogo



nel menu in alto. Puoi cercare GitHub repository, repository Bitbucket, GitLab repository o Jira Software. Puoi anche filtrare le estensioni in base alle categorie.

4. (Facoltativo) Scegli il nome dell'estensione per visualizzare maggiori dettagli sull'estensione, come le autorizzazioni che avrà l'estensione.
5. Scegli Installa. Controlla le autorizzazioni richieste dall'estensione e, se vuoi continuare, scegli nuovamente Installa.

Dopo aver installato un'estensione, vedrai la pagina dei dettagli dell'estensione installata. Sfoglia le schede per ulteriori informazioni sull'estensione. Nella pagina dei dettagli è anche possibile eseguire ulteriori configurazioni dell'estensione, se necessario.

Disinstallazione di un'estensione in uno spazio

Puoi disinstallare le estensioni precedentemente installate nel tuo CodeCatalyst spazio. La disinstallazione di un'estensione può rimuovere le risorse relative a tale estensione dal tuo CodeCatalyst spazio o dai tuoi progetti.

Important

Per disinstallare un'estensione, devi accedere con un account che abbia il ruolo di amministratore dello spazio nello spazio.

Per disinstallare un'estensione dal tuo CodeCatalyst spazio

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.
3. Effettua una delle seguenti operazioni per visualizzare un elenco delle estensioni installate per il tuo spazio:
 - a. Scegli Impostazioni, quindi scegli Estensioni installate.
 - b. Scegli l'icona del catalogo

nel menu in alto.
4. Scegli Configura sull'estensione che desideri disinstallare.

5. Scegli Disinstalla nella pagina dei dettagli dell'estensione.
6. Controlla le informazioni nella finestra di dialogo Disinstalla l'estensione. Segui le istruzioni, quindi scegli Disinstalla per disinstallare l'estensione.

Connessione di GitHub account, aree di lavoro Bitbucket, GitLab utenti e siti Jira CodeCatalyst

Per utilizzare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto o gestire un progetto Jira in CodeCatalyst, devi prima connettere la fonte di terze parti al tuo spazio. CodeCatalyst Per ulteriori informazioni sulle estensioni e sulle relative funzionalità, consulta.

[Estensioni di terze parti disponibili](#)

Important

Per connettere il tuo GitHub account, l'area di lavoro Bitbucket, GitLab l'utente o il sito Jira al tuo CodeCatalyst spazio, devi essere sia l'amministratore della fonte di terze parti che l'amministratore dello spazio. CodeCatalyst

Note

Se utilizzi una connessione a un GitHub account, devi creare una connessione personale per stabilire una mappatura delle identità tra la tua identità e la tua CodeCatalyst identità. GitHub Per ulteriori informazioni, consultare [Connessioni personali](#) e [Accesso alle GitHub risorse con connessioni personali](#).

Per connettere il tuo GitHub account, l'area di lavoro, l' GitLab utente o il sito Jira di Bitbucket a CodeCatalyst

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/) CodeCatalyst .
2. Accedi al tuo CodeCatalyst spazio.
3. Effettua una delle seguenti operazioni per visualizzare un elenco delle estensioni installate per il tuo spazio:
 - a. Scegli Impostazioni, quindi scegli Estensioni installate.

- b. Scegli l'icona del catalogo



nel menu in alto.

4. Scegli Configura per una delle seguenti estensioni che desideri configurare: GitHub repository, repository Bitbucket, GitLab repository o Jira Software.
5. Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai scelto di configurare:

- GitHub repository: Connect a un GitHub account.
 1. Nella scheda GitHub Account connessi, scegli Connetti GitHub account per accedere al sito esterno di GitHub.
 2. Accedi al tuo GitHub account utilizzando GitHub le tue credenziali, quindi scegli l'account in cui desideri installare Amazon CodeCatalyst.

 Tip

Se in precedenza hai collegato un GitHub account allo spazio, non ti verrà richiesto di autorizzarlo nuovamente. Vedrai invece una finestra di dialogo che ti chiede dove desideri installare l'estensione se sei un membro o un collaboratore in più di uno GitHub spazio, o la pagina di configurazione per l' CodeCatalyst applicazione Amazon se appartieni a un solo GitHub spazio. Configura l'applicazione per l'accesso al repository che desideri consentire, quindi scegli Salva. Se il pulsante Salva non è attivo, apporta una modifica alla configurazione, quindi riprova.

3. Scegli se consentire l'accesso CodeCatalyst a tutti i repository attuali e futuri o scegli i GitHub repository specifici in cui desideri utilizzare. CodeCatalyst L'opzione predefinita prevede l'inclusione di tutti i GitHub repository nell' GitHub account, compresi i repository futuri a cui sarà possibile accedere da. CodeCatalyst
4. Controlla le autorizzazioni concesse a CodeCatalyst, quindi scegli Installa.

Dopo aver collegato il tuo GitHub account a CodeCatalyst, verrai reindirizzato alla pagina dei dettagli dell'estensione dei GitHub repository, dove puoi visualizzare e gestire GitHub gli account connessi e gli archivi collegati GitHub .

- Archivi Bitbucket: Connettiti a uno spazio di lavoro Bitbucket.
 1. Nella scheda Connected Bitbucket workspace, scegli Connect Bitbucket workspace per accedere al sito esterno di Bitbucket.
 2. Accedi al tuo spazio di lavoro Bitbucket utilizzando le tue credenziali Bitbucket e controlla le autorizzazioni concesse a. CodeCatalyst
 3. Dal menu a discesa Autorizza per l'area di lavoro, scegli l'area di lavoro Bitbucket a cui desideri fornire l'accesso, quindi scegli Concedi l'accesso. CodeCatalyst

 Tip

Se in precedenza hai collegato un'area di lavoro Bitbucket allo spazio, non ti verrà richiesto di autorizzare nuovamente. Vedrai invece una finestra di dialogo che ti chiede dove desideri installare l'estensione se sei un membro o un collaboratore in più di uno spazio di lavoro Bitbucket, o la pagina di configurazione per l' CodeCatalyst applicazione Amazon se appartieni a un solo spazio di lavoro Bitbucket. Configura l'applicazione per l'accesso allo spazio di lavoro che desideri consentire, quindi scegli Concedi l'accesso. Se il pulsante Concedi l'accesso non è attivo, apporta una modifica alla configurazione, quindi riprova.

Dopo aver collegato l'area di lavoro Bitbucket a CodeCatalyst, verrai indirizzato alla pagina dei dettagli dell'estensione dei repository Bitbucket, dove puoi visualizzare e gestire le aree di lavoro Bitbucket connesse e gli archivi Bitbucket collegati.

- GitLab repository: Connect a un GitLab utente.
 1. Scegli Connect GitLab user per accedere al sito esterno per GitLab.
 2. Accedi al tuo GitLab utente utilizzando GitLab le tue credenziali e controlla le autorizzazioni concesse a. CodeCatalyst

 Tip

Se in precedenza hai collegato un GitLab utente allo spazio, non ti verrà richiesto di autorizzarlo nuovamente. Verrai invece reindirizzato alla console. CodeCatalyst

3. Scegli Authorize AWS Connector per. GitLab

Dopo aver collegato GitLab l'utente a CodeCatalyst, verrai indirizzato alla pagina dei dettagli GitLab delle estensioni dei repository, dove puoi visualizzare e gestire gli archivi connessi GitLab degli utenti e dei GitLab progetti collegati.

- Jira Software: Connect un sito Jira.
 1. Nella scheda Siti Jira connessi, scegli Connetti sito Jira per accedere al sito esterno per Atlassian Marketplace.
 2. Scegli Scarica subito per iniziare l'installazione CodeCatalyst sul tuo sito Jira.

 Note

Se l'hai già installato CodeCatalyst sul tuo sito Jira, riceverai una notifica. Scegli Inizia per passare alla fase finale.

3. A seconda del ruolo, esegui una delle seguenti operazioni:
 1. Se sei un amministratore del sito Jira, dal menu a discesa del sito, scegli il sito Jira per installare l' CodeCatalyst applicazione, quindi scegli Installa app.

 Note

Se hai un sito Jira, questo passaggio non verrà visualizzato e verrai indirizzato automaticamente al passaggio successivo.

2. a. Se non sei un amministratore di Jira, dal menu a discesa del sito, scegli il sito Jira per installare l' CodeCatalyst applicazione, quindi scegli Richiedi app. Per ulteriori informazioni sull'installazione delle app Jira, consulta [Chi può installare le app?](#) .
 - b. Inserisci il motivo per cui devi effettuare l'installazione CodeCatalyst nel campo di testo di input o mantieni il testo predefinito, quindi scegli Invia richiesta.
4. Controlla le azioni eseguite al CodeCatalyst momento dell'installazione dell'applicazione, quindi scegli Scarica ora.
5. Dopo aver installato l'applicazione, scegli Return CodeCatalyst to per tornare a CodeCatalyst.

Dopo aver collegato il tuo sito Jira a CodeCatalyst, puoi visualizzare il sito connesso nella scheda Siti Jira connessi della pagina dei dettagli dell'estensione di Jira Software.

Se non desideri più utilizzare GitHub repository, repository Bitbucket o repository di GitLab progetti o gestire i problemi di Jira in, puoi disconnettere la fonte di terze parti. CodeCatalyst Quando un GitHub account, un'area di lavoro Bitbucket o un GitLab utente viene disconnesso, gli eventi nei repository di terze parti non avviano l'esecuzione del flusso di lavoro e non potrai utilizzare tali repository con Dev Environments. CodeCatalyst Quando un sito Jira viene disconnesso, i problemi di Jira dei progetti del sito non saranno disponibili nei progetti e Issues sarà nuovamente il fornitore CodeCatalyst dei problemi. CodeCatalyst Per ulteriori informazioni, consulta [Disconnessione di GitHub account, aree di lavoro Bitbucket, GitLab utenti e siti Jira CodeCatalyst](#).

Disconnessione di GitHub account, aree di lavoro Bitbucket, GitLab utenti e siti Jira CodeCatalyst

Se non desideri più utilizzare GitHub repository, repository Bitbucket o repository di GitLab progetti o gestire i problemi di Jira in, puoi disconnettere la fonte di terze parti. CodeCatalyst Una volta disconnesso un GitHub account, un'area di lavoro Bitbucket o un GitLab utente, gli eventi nei repository non avvieranno l'esecuzione del flusso di lavoro e non potrai utilizzare tali repository con Dev Environments. CodeCatalyst Quando un sito Jira viene disconnesso, i problemi di Jira dei progetti del sito non saranno disponibili nei progetti e Issues sarà nuovamente il fornitore CodeCatalyst dei problemi. CodeCatalyst

Note

- Per disconnettere un GitHub account, devi prima scollegare tutti gli archivi collegati GitHub da quell'account.
- Per disconnettere un'area di lavoro Bitbucket, devi prima scollegare tutti i repository Bitbucket collegati da quell'area di lavoro.
- Per disconnettere un GitLab utente, devi prima scollegare tutti gli archivi di progetto collegati da quell'area di lavoro. GitLab
- Per disconnettere un sito Jira, devi prima scollegare tutti i progetti Jira collegati da quell'account.

Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Per disconnettere un GitHub progetto, un'area di lavoro Bitbucket, un utente o un sito Jira GitLab

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Accedi al tuo CodeCatalyst spazio.
3. Effettua una delle seguenti operazioni per visualizzare un elenco delle estensioni installate per il tuo spazio:
 - a. Scegli Impostazioni, quindi scegli Estensioni installate.
 - b. Scegli l'icona del catalogo



nel menu in alto.

4. Scegli Configura per una delle seguenti estensioni che desideri configurare: GitHub repository, repository Bitbucket, GitLab repository o Jira Software.
5. Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai scelto di configurare:

- GitHub archivi: disconnettiti a un account. GitHub

Nella scheda GitHub Account connessi, scegli l' GitHub account che desideri disconnettere, quindi scegli Disconnetti account. GitHub

- Archivi Bitbucket: disconnettiti a uno spazio di lavoro Bitbucket.

Nella scheda Aree di lavoro Bitbucket connesse, scegli l'area di lavoro Bitbucket che desideri disconnettere, quindi scegli Disconnetti l'area di lavoro Bitbucket.

- GitLab archivi: GitLab disconnettiti a un utente.

Nella scheda GitLab Utenti connessi, scegli l' GitLab utente che desideri disconnettere, quindi scegli Disconnetti utente. GitLab

- Jira Software: disconnettiti a un sito Jira.

Nella scheda Siti Jira connessi, scegli il sito Jira che desideri disconnettere, quindi scegli Disconnetti il sito Jira.

6. Nella finestra di dialogo Disconnetti, esamina gli effetti della disconnessione dell'account.
7. Inserisci disconnect nel campo di immissione del testo, quindi scegli Disconnetti.

Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst

Prima di poter utilizzare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto o gestire un progetto Jira, devi connettere la fonte di terze parti a cui appartiene il repository o il progetto con il tuo spazio. CodeCatalyst Per ulteriori informazioni, consulta [Connessione di GitHub account, aree di lavoro Bitbucket, GitLab utenti e siti Jira CodeCatalyst](#).

Puoi utilizzare GitHub repository collegati, repository Bitbucket o repository di GitLab progetto nei flussi di lavoro, in cui gli eventi negli archivi collegati avviano flussi di lavoro che potrebbero creare, testare o distribuire codice, a seconda della configurazione del flusso di lavoro. I file di configurazione dei flussi di lavoro per i flussi di lavoro che utilizzano repository collegati o Bitbucket vengono archiviati negli archivi collegati GitHub . I repository collegati possono essere utilizzati anche con Dev Environments per creare, aggiornare ed eliminare file nei repository collegati. Puoi collegare un GitHub repository, un repository Bitbucket o un repository di progetto a un GitLab CodeCatalyst progetto dalla pagina dei dettagli dei repository, dai repository Bitbucket o dall'estensione dei GitHub repository, oppure dalla visualizzazione dei repository di origine in GitLab Code nel progetto stesso.

Important

Sebbene sia possibile collegare un repository GitHub o Bitbucket come collaboratore, è possibile scollegare un repository di terze parti solo come amministratore dello spazio o amministratore del progetto. Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Important

Dopo aver installato un'estensione del repository, il codice di tutti gli archivi a cui ti colleghi CodeCatalyst verrà indicizzato e archiviato. CodeCatalyst In questo modo il codice sarà ricercabile in. CodeCatalyst Per comprendere meglio la protezione dei dati per il tuo codice quando utilizzi repository collegati in CodeCatalyst, consulta la sezione [Protezione dei dati](#) nella Amazon CodeCatalyst User Guide.

 Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo da CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. È consigliabile assicurarsi sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Puoi utilizzare i progetti Jira collegati per gestire i problemi e collegare le richieste CodeCatalyst pull a un problema di Jira. Lo stato riepilogativo di una pull request e lo stato degli eventi del CodeCatalyst flusso di lavoro associati si riflettono nel tuo problema di Jira.

 Important

Per collegare il tuo progetto Jira al tuo CodeCatalyst progetto, devi essere l'amministratore CodeCatalyst dello spazio o l'amministratore del CodeCatalyst progetto.

 Note

- Un GitHub repository, un repository Bitbucket o un repository di GitLab progetto possono essere collegati a un solo progetto in uno spazio. CodeCatalyst
- Non puoi utilizzare repository vuoti o archiviati, repository Bitbucket o GitHub repository di progetti con progetti. GitLab CodeCatalyst
- Non puoi collegare un GitHub repository, un repository Bitbucket o un repository con lo stesso nome di un GitLab repository in un progetto. CodeCatalyst
- L'estensione GitHub repositories non è compatibile con gli archivi di Enterprise Server. GitHub
- L'estensione Bitbucket repositories non è compatibile con i repository Bitbucket Data Center.
- L'estensione GitLab repositories non è compatibile con gli archivi di progetto autogestiti. GitLab
- Non puoi utilizzare le funzioni Scrivi descrizione per me o Riepiloga commenti con gli archivi collegati. Queste funzionalità sono disponibili solo nelle pull request in. CodeCatalyst

- Un CodeCatalyst progetto può essere collegato a un solo progetto Jira. Un progetto Jira può essere collegato a più CodeCatalyst progetti.

Argomenti

- [Collegamento di risorse da fornitori di terze parti collegati](#)
- [Collegamento di un repository di terze parti durante la creazione del progetto CodeCatalyst](#)

Collegamento di risorse da fornitori di terze parti collegati

Per collegare un GitHub repository, un repository Bitbucket, un repository di progetto o un GitLab progetto Jira a un progetto dalla pagina dei dettagli dell'estensione CodeCatalyst

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). [CodeCatalyst](#)
2. Accedi al tuo CodeCatalyst spazio.
3. Effettua una delle seguenti operazioni per visualizzare un elenco delle estensioni installate per il tuo spazio spaziale:
 - a. Scegliete Impostazioni, quindi scegliete Estensioni installate.
 - b. Scegli l'icona del catalogo

nel menu in alto.
4. Scegli Configura per una delle seguenti estensioni: GitHub repository, repository Bitbucket, GitLab repository o Jira Software.
5. Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai scelto di configurare:
 - GitHub repository: collega un GitHub repository.
 1. Nella scheda GitHub Archivi collegati, scegli Collega repository. GitHub
 2. Dal menu a discesa dell'GitHub account, scegli l' GitHub account che contiene il repository che desideri collegare.
 3. Dal menu a discesa del GitHub repository, scegli il repository che desideri collegare a un progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto nello spazio.

4. (Facoltativo) Se non vedi un GitHub repository nell'elenco dei repository, è possibile che non sia stato configurato per l'accesso al repository nell'applicazione Amazon CodeCatalyst in. GitHub Puoi configurare quali GitHub repository possono essere utilizzati CodeCatalyst nell'account connesso.
 - a. Accedi al tuo [GitHub](#) account, scegli Impostazioni, quindi scegli Applicazioni.
 - b. Nella scheda GitHub App installate, scegli Configura per l' CodeCatalyst applicazione Amazon.
 - c. Effettua una delle seguenti operazioni per configurare l'accesso ai GitHub repository a cui desideri collegarti: CodeCatalyst
 - Per fornire l'accesso a tutti i repository attuali e futuri, scegli Tutti i repository.
 - Per consentire l'accesso a repository specifici, seleziona solo repository, scegli il menu a discesa Seleziona repository, quindi scegli un repository a cui desideri consentire il collegamento. CodeCatalyst
5. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto a cui vuoi collegare il repository. GitHub
6. Scegliere Link (Collegamento).

Se non desideri più utilizzare un GitHub repository CodeCatalyst, puoi scollegarlo da un progetto. CodeCatalyst Quando un repository non è collegato, gli eventi in quel repository non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

- Archivi Bitbucket: collega un repository Bitbucket.
 1. Nella scheda Archivi Bitbucket collegati, scegli Collega repository Bitbucket.
 2. Dal menu a discesa dell'area di lavoro Bitbucket, scegli l'area di lavoro Bitbucket che contiene il repository che desideri collegare.

3. Dal menu a discesa del repository Bitbucket, scegli il repository che desideri collegare a un progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto nello spazio.

4. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto a cui vuoi collegare il repository Bitbucket.
5. Scegliere Link (Collegamento).

Se non desideri più utilizzare un repository Bitbucket in CodeCatalyst, puoi scollegarlo da un progetto. CodeCatalyst Quando un repository non è collegato, gli eventi in quel repository non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

- GitLab repository: collega un repository di progetto. GitLab
 1. Nella scheda Archivi di GitLab progetti collegati, scegli Collega GitLab archivio di progetti.
 2. Dal menu a discesa GitLab utente, scegli l' GitLab utente che contiene l'archivio del progetto che desideri collegare.
 3. Dal menu a discesa GitLab del repository del progetto, scegli il repository che desideri collegare a un progetto. CodeCatalyst

 Tip

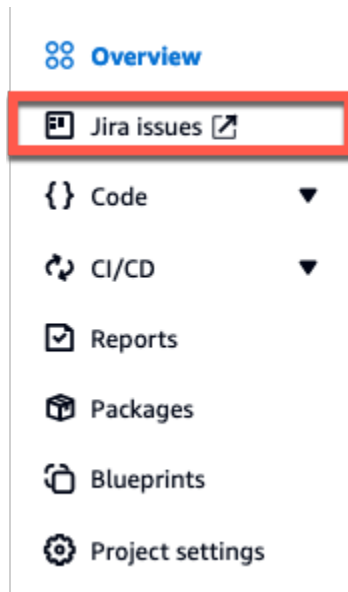
Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto nello spazio.

4. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto a cui vuoi collegare l'archivio del GitLab progetto.
5. Scegliere Link (Collegamento).

Se non desideri più utilizzare un repository di GitLab progetto CodeCatalyst, puoi scollegarlo da un progetto. CodeCatalyst Quando un repository di progetto non è collegato, gli eventi in quel repository di progetto non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository di progetto con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

- Jira Software: collega un progetto Jira.
 1. Nella scheda Progetti Jira collegati, scegli Collega un progetto Jira.
 2. Dal menu a discesa del sito Jira, scegli il sito Jira che contiene il progetto che desideri collegare.
 3. Dal menu a discesa del progetto Jira, scegli il progetto che desideri collegare a un progetto. CodeCatalyst
 4. Dal menu a discesa del CodeCatalyst progetto, scegli il CodeCatalyst progetto che desideri collegare a un progetto Jira.
 5. Scegliere Link (Collegamento).

Una volta collegato un progetto Jira a un CodeCatalyst progetto, l'accesso ai CodeCatalyst problemi viene completamente disabilitato e i problemi nel pannello di CodeCatalyst navigazione verranno sostituiti con un elemento Jira issues che si collega al progetto Jira.



Se non desideri più utilizzare un progetto Jira in CodeCatalyst, puoi scollegarlo dal tuo progetto. CodeCatalyst Quando un progetto Jira non è collegato, i problemi di Jira non saranno disponibili nel CodeCatalyst progetto e Issues sarà nuovamente il CodeCatalyst fornitore dei problemi. Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Per collegare un GitHub repository, un repository Bitbucket o un repository di progetto a un GitLab progetto dalla pagina dei repository di origine di un CodeCatalyst progetto

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Vai al tuo CodeCatalyst progetto.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli Aggiungi repository, quindi scegli Collega repository.
5. Dal menu a discesa Repository provider, scegli uno dei seguenti provider di repository di terze parti: Bitbucket, GitHubGitLab
6. Esegui una delle seguenti operazioni a seconda del provider di repository di terze parti che hai scelto di collegare:
 - GitHub repository: collega un GitHub repository.
 1. Dal menu a discesa dell'GitHub account, scegli l' GitHub account che contiene il repository che desideri collegare.
 2. Dal menu a discesa del GitHub repository, scegli il GitHub repository a cui vuoi collegare il tuo progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto in Amazon. CodeCatalyst

3. (Facoltativo) Se non vedi un GitHub repository nell'elenco dei repository, è possibile che non sia stato configurato per l'accesso al repository nell'applicazione Amazon CodeCatalyst in. GitHub Puoi configurare quali GitHub repository possono essere utilizzati CodeCatalyst nell'account connesso.
 - a. Accedi al tuo [GitHub](#) account, scegli Impostazioni, quindi scegli Applicazioni.

- b. Nella scheda GitHub App installate, scegli Configura per l' CodeCatalyst applicazione Amazon.
- c. Effettua una delle seguenti operazioni per configurare l'accesso ai GitHub repository a cui desideri collegarti: CodeCatalyst
 - Per fornire l'accesso a tutti i repository attuali e futuri, scegli Tutti i repository.
 - Per consentire l'accesso a repository specifici, seleziona solo repository, scegli il menu a discesa Seleziona repository, quindi scegli un repository a cui desideri consentire il collegamento. CodeCatalyst
- Repository Bitbucket: collega un repository Bitbucket.
 1. Dal menu a discesa Bitbucket Workspace, scegli l'area di lavoro Bitbucket che contiene il repository che desideri collegare.
 2. Dal menu a discesa del repository Bitbucket, scegli il repository Bitbucket a cui desideri collegare il tuo progetto. CodeCatalyst

 Tip

Se il nome del repository è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto in Amazon. CodeCatalyst

- GitLab repository: collega un archivio di progetto. GitLab
 1. Dal menu a discesa GitLab utente, scegli l' GitLab utente che contiene l'archivio del progetto che desideri collegare.
 2. Dal menu a discesa GitLab del repository del progetto, scegli l'archivio del GitLab progetto a cui desideri collegare il tuo progetto. CodeCatalyst

 Tip

Se il nome dell'archivio del progetto è disattivato, non puoi collegarlo perché è già stato collegato a un altro progetto in Amazon. CodeCatalyst

7. Scegliere Link (Collegamento).

Se non desideri più utilizzare un GitHub repository, un repository Bitbucket o un repository di progetto in CodeCatalyst, puoi scollegarlo da un GitLab progetto. CodeCatalyst Quando un repository non

È collegato, gli eventi in quel repository non avviano l'esecuzione del flusso di lavoro e non sarà possibile utilizzare tale repository con Dev Environments. CodeCatalyst Per ulteriori informazioni, consulta [Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Dopo aver collegato il tuo GitHub repository, il repository Bitbucket o il repository del progetto al tuo GitLab CodeCatalyst progetto, puoi utilizzarlo nei flussi di lavoro e negli ambienti di sviluppo. CodeCatalyst Puoi anche utilizzare i repository collegati con Amazon Q Developer, blueprint e altro ancora. Per ulteriori informazioni, consultare [Avvio automatico di un flusso di lavoro eseguito dopo eventi di repository di terze parti](#) e [Creazione di un ambiente di sviluppo](#).

Dopo aver collegato il tuo progetto Jira al tuo CodeCatalyst progetto e aver collegato una pull request, gli aggiornamenti provenienti da si CodeCatalyst riflettono nel tuo progetto Jira. Per ulteriori informazioni sul collegamento delle pull request ai problemi di Jira, consulta. [Collegamento dei problemi di Jira alle richieste pull CodeCatalyst](#) Per ulteriori informazioni sulla visualizzazione CodeCatalyst degli eventi in Jira, consulta. [Visualizzazione CodeCatalyst degli eventi nei problemi di Jira](#)

Collegamento di un repository di terze parti durante la creazione del progetto CodeCatalyst

Puoi collegare un GitHub repository, un repository Bitbucket o un repository di progetto a un GitLab nuovo progetto durante la creazione del nuovo progetto. CodeCatalyst CodeCatalyst Per ulteriori informazioni, consulta [Creazione di un progetto con un repository di terze parti collegato](#).

Scollegamento di repository, GitHub repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst

Se non desideri più utilizzare un GitHub repository, un repository Bitbucket o un repository di progetti, o gestire un GitLab progetto Jira in, puoi scollegare il repository o il progetto dal tuo progetto. CodeCatalyst CodeCatalyst

Lo scollegamento di un repository, un GitHub repository Bitbucket o un repository di progetto non elimina il repository né GitLab apporta modifiche ad esso. Non elimina alcun file di configurazione del flusso di lavoro archiviato in quell'archivio collegato. Tuttavia, una volta scollegato un GitHub repository, un repository Bitbucket o un repository di GitLab progetto, gli eventi in quel repository

non avvieranno più l'esecuzione del flusso di lavoro e non potrai utilizzare il repository con Dev Environments. Puoi scollegare un GitHub repository, un repository Bitbucket o un repository di GitLab progetto da un CodeCatalyst progetto dalla pagina dei dettagli dei repository, dei GitHub repository Bitbucket o dall'estensione dei GitLab repository, o dalla vista dei repository di origine in Code nel progetto stesso.

Lo scollegamento di un progetto Jira non elimina il progetto, inclusi gli elementi di pianificazione o le informazioni di sviluppo, né apporta alcuna modifica allo stesso. Tuttavia, una volta scollegato un progetto Jira, i problemi Jira del progetto non saranno più disponibili per il collegamento al CodeCatalyst progetto e Issues sarà nuovamente il fornitore CodeCatalyst dei problemi.

 Important

Per scollegare il tuo GitHub repository, il repository Bitbucket o il repository del progetto Gitlab dal tuo progetto, devi essere l'amministratore dello spazio o l'amministratore CodeCatalyst del progetto.

Per scollegare un GitHub repository, un repository Bitbucket, un repository di progetto o un progetto Jira in un GitLab progetto dalla pagina dei dettagli dell'estensione CodeCatalyst

1. CodeCatalyst [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.
3. Effettua una delle seguenti operazioni per visualizzare un elenco delle estensioni installate per il tuo spazio:
 - a. Scegli Impostazioni, quindi scegli Estensioni installate.
 - b. Scegli l'icona del catalogo  nel menu in alto.
4. Scegli Configura per una delle seguenti estensioni che desideri configurare: GitHub repository, repository Bitbucket, GitLab repository o Jira Software.
5. Esegui una delle seguenti operazioni a seconda dell'estensione di terze parti che hai scelto di configurare:
 - GitHub repository: scollega un GitHub repository.

Nella scheda GitHub repository, scegli il GitHub repository che desideri scollegare, quindi scegli Scollega repository. GitHub

- Archivi Bitbucket: scollega un repository Bitbucket.

Nella scheda Archivi Bitbucket, scegli il repository Bitbucket che desideri scollegare, quindi scegli Scollega l'archivio Bitbucket.

- GitLab repository GitLab : scollega un archivio di progetto.

Nella scheda Archivi del GitLab progetto, scegli l'archivio del GitLab progetto che desideri scollegare, quindi scegli Scollega l'archivio del progetto. GitLab

- Jira Software: scollega un progetto Jira.

Nella scheda Progetti Jira, scegli il progetto Jira che desideri scollegare, quindi scegli Scollega il progetto Jira.

6. Nella finestra di dialogo Scollega, esamina gli effetti dello scollegamento del repository.
7. Immettete unlink nel campo di immissione del testo e scegliete Scollega.

Per scollegare un GitHub repository, un repository Bitbucket o un repository di progetto in un GitLab progetto dalla pagina dei repository di origine CodeCatalyst

1. CodeCatalyst [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo CodeCatalyst progetto.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.
4. Scegli il pulsante di opzione del repository che desideri scollegare, quindi scegli Scollega archivio.
5. Rivedi le informazioni nella finestra di dialogo. Segui le istruzioni, quindi scegli Scollega per scollegare il repository.

Visualizzazione di repository di terze parti e ricerca dei problemi di Jira in CodeCatalyst

Dopo aver collegato repository, GitHub repository Bitbucket o repository di GitLab progetti, puoi visualizzarli per confermare e configurare le risorse. CodeCatalyst Puoi anche cercare problemi Jira collegati in. CodeCatalyst

Argomenti

- [Visualizzazione di repository di terze parti in CodeCatalyst](#)
- [Ricerca dei problemi di Jira in CodeCatalyst](#)

Visualizzazione di repository di terze parti in CodeCatalyst

Puoi visualizzare i GitHub repository collegati, i repository Bitbucket o i repository di GitLab progetto nell'elenco dei repository di origine del tuo progetto o dalla pagina dei dettagli dei repository, dei repository Bitbucket o dell'GitHub estensione dei repository. GitLab Scegliarli dall'elenco dei repository non li apre in. CodeCatalyst Si aprono invece nel provider di repository di terze parti, dove è possibile visualizzare e modificare il codice nell'archivio collegato.

Per visualizzare gli GitHub archivi collegati, gli archivi Bitbucket o gli archivi di progetti in GitLab CodeCatalyst

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Vai al tuo CodeCatalyst progetto.
3. Nel riquadro di navigazione, scegli Codice, quindi scegli Archivi di origine.

Per visualizzare i GitHub repository collegati, i repository Bitbucket o i repository di GitLab progetto dalla pagina dei dettagli dell'estensione

1. [Apri la console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/). CodeCatalyst
2. Accedi al tuo CodeCatalyst spazio, quindi scegli la scheda Estensioni installate.
3. A seconda dei repository di terze parti che desideri visualizzare, esegui una delle seguenti operazioni:
 - Nei GitHub repository, scegli Configura, quindi scegli GitHub Archivi collegati per visualizzare tutti gli GitHub archivi collegati ai CodeCatalyst progetti presenti nel tuo spazio. CodeCatalyst
 - Nei repository Bitbucket, scegli Configura, quindi scegli Archivi Bitbucket collegati per visualizzare tutti gli archivi Bitbucket collegati ai progetti nel tuo spazio. CodeCatalyst CodeCatalyst
 - Nei GitLab repository, scegli Configura, quindi scegli Archivi di progetti collegati per visualizzare tutti gli archivi di GitLab progetto collegati ai progetti presenti nel tuo spazio. GitLab CodeCatalyst CodeCatalyst

I GitHub repository, i repository Bitbucket o gli archivi di GitLab progetto collegati al tuo progetto sono mostrati nell'elenco. CodeCatalyst Scegli il GitHub repository, il repository Bitbucket o il repository del GitLab progetto per visualizzare e modificare i file nel fornitore di repository di terze parti.

Note

Se un flusso di lavoro utilizza un GitHub repository, un repository Bitbucket o un repository di GitLab progetto in un'azione sorgente, le modifiche apportate al flusso di lavoro YAML nell'editor visivo o nell'editor YAML verranno automaticamente salvate e inviate all'archivio di terze parti. CodeCatalyst

Ricerca dei problemi di Jira in CodeCatalyst

Dopo aver collegato un progetto Jira, puoi cercare problemi nel progetto Jira collegato utilizzando la CodeCatalyst barra di ricerca globale. Puoi anche cercare i problemi di Jira CodeCatalyst mentre ti colleghi ai problemi di una pull request. Per ulteriori informazioni sul collegamento dei problemi di Jira a una CodeCatalyst pull request, consulta. [Collegamento dei problemi di Jira alle richieste pull CodeCatalyst](#)

Per cercare i problemi di Jira nei progetti Jira collegati

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Vai al tuo CodeCatalyst progetto.
3. Nella barra di ricerca globale, cerca i problemi relativi a un progetto Jira collegato o i problemi di Jira che desideri collegare a una pull request.

Avvio automatico di un flusso di lavoro eseguito dopo eventi di repository di terze parti

Puoi utilizzare un GitHub repository collegato, un repository Bitbucket o un repository di GitLab progetto come origine per un flusso di lavoro, in cui le modifiche a un ramo specifico in un repository collegato, un repository Bitbucket o un GitHub repository di progetto avviano automaticamente l'esecuzione di un flusso di lavoro. GitLab

Un flusso di lavoro è una procedura automatizzata che descrive come creare, testare e distribuire il codice come parte di un sistema di integrazione e distribuzione continue (CI/CD). Un flusso di lavoro

definisce una serie di passaggi, o azioni, da eseguire durante l'esecuzione di un flusso di lavoro. Un flusso di lavoro definisce anche gli eventi, o trigger, che causano l'avvio del flusso di lavoro. Per configurare un flusso di lavoro, è necessario creare un file di definizione del flusso di lavoro utilizzando l'editor [visivo o YAML](#) della CodeCatalyst console.

Tip

Per una rapida occhiata a come potresti utilizzare i flussi di lavoro in un progetto, [crea un progetto con un blueprint](#). Ogni blueprint implementa un flusso di lavoro funzionante che puoi rivedere, eseguire e sperimentare.

Quando configuri un flusso di lavoro per utilizzare un GitHub repository collegato, un repository Bitbucket o un repository di GitLab progetto, il file di configurazione del flusso di lavoro viene archiviato in tale repository, repository Bitbucket o GitHub repository di progetto. GitLab La configurazione del flusso di lavoro è un file YAML che definisce il nome del flusso di lavoro, i trigger, le risorse, gli artefatti e le azioni. Per ulteriori informazioni sul file di configurazione del flusso di lavoro, vedere. [Definizione YAML del flusso di lavoro](#)

Il file di configurazione del flusso di lavoro deve trovarsi nella `./codecatalyst/workflows/` directory del tuo GitHub repository, del repository Bitbucket o del repository del progetto. GitLab

È possibile utilizzare l'editor di flussi di lavoro per creare e configurare flussi di lavoro. Per ulteriori informazioni, consulta [Guida introduttiva ai flussi di lavoro](#) e [Connessione degli archivi di origine ai flussi di lavoro](#).

L'aggiunta di trigger per avviare le esecuzioni del flusso di lavoro

Puoi configurare un CodeCatalyst flusso di lavoro per avviare automaticamente un'esecuzione quando il codice viene inviato al ramo specificato del tuo repository GitHub o di Bitbucket. Per avviare automaticamente un workflow, aggiungi un trigger alla Triggers sezione del file di configurazione del workflow.

Esempio: un semplice trigger tramite codice

L'esempio seguente mostra un trigger che avvia un flusso di lavoro ogni volta che il codice viene inviato a qualsiasi ramo del repository di origine.

Triggers:

- Type: PUSH

Esempio: un semplice trigger di pull request

L'esempio seguente mostra un trigger che avvia un flusso di lavoro ogni volta che viene creata una richiesta pull su qualsiasi ramo del repository di origine.

```
Triggers:  
- Type: PULLREQUEST  
Events:  
- OPEN
```

Per ulteriori informazioni, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Limitazione dell'accesso IP con provider di repository di terze parti

Puoi limitare l'accesso ai tuoi GitHub repository, agli repository Bitbucket o agli archivi di GitLab progetti in base agli indirizzi IP impostando regole o configurazioni. Puoi farlo tramite le impostazioni o le funzionalità di controllo degli accessi del provider terzo.

A seconda del provider di repository di terze parti che stai utilizzando, vedi una delle seguenti opzioni:

- L'estensione Amazon CodeCatalyst GitHub Repositories è compatibile con le [restrizioni di accesso IP di GitHub Enterprise Cloud](#). Quando configuri un'organizzazione GitHub Enterprise Cloud per limitare l'accesso a indirizzi IP specifici, puoi anche [abilitare GitHub le app a configurare l'elenco degli indirizzi IP consentiti](#), che consentirà di CodeCatalyst registrare automaticamente i propri indirizzi IP con. GitHub In alternativa, è possibile [aggiungere manualmente gli indirizzi CodeCatalyst IP](#).
- L'estensione dei repository Amazon CodeCatalyst Bitbucket è compatibile con le restrizioni di accesso a [Bitbucket Cloud Premium](#). Quando configuri un'area di lavoro Bitbucket Cloud Premium per limitare l'accesso a indirizzi IP specifici, puoi anche [aggiungere indirizzi IP o blocchi di rete per un set di indirizzi IP a una lista di indirizzi IP](#) consentiti.
- L'estensione Amazon CodeCatalyst GitLab Repositories è compatibile con le [restrizioni GitLab degli indirizzi IP](#). Quando configuri un gruppo GitLab Premium o Ultimate per limitare l'accesso a indirizzi IP specifici, puoi anche [aggiungere indirizzi IP o blocchi di rete per un set di indirizzi IP a una](#) lista consentita.

Se gli indirizzi CodeCatalyst IP non sono presenti nell'elenco consentito di un repository di terze parti, CodeCatalyst l'app Amazon non sarà in grado di accedere ai tuoi repository di terze parti. Per ulteriori informazioni, consulta [Indirizzi IP utilizzati dall'estensione degli archivi di terze parti](#).

Indirizzi IP utilizzati dall'estensione degli archivi di terze parti

I seguenti indirizzi IP vengono utilizzati dalle estensioni di terze parti per accedere alle risorse di terze parti:

- GitHub archivi:

```
us-west-2
  52.32.242.246
  54.148.176.49
  35.164.118.94
eu-west-1
  34.241.64.10
  34.246.255.80
  3.248.38.7
```

- Repository e repository Bitbucket: GitLab

```
us-west-2
  35.160.210.199
  54.71.206.108
  54.71.36.205
eu-west-1
  34.242.64.82
  52.18.37.201
  54.77.75.62
```

Bloccare le fusioni di terze parti quando i flussi di lavoro falliscono

Dopo aver collegato un repository GitHub o Bitbucket a, puoi aggiungere flussi di lavoro per le richieste pull CodeCatalyst. CodeCatalyst Allo stesso modo, dopo aver collegato un repository di GitLab progetto a, CodeCatalyst puoi aggiungere flussi di lavoro per le richieste di unione. CodeCatalyst Una o più esecuzioni di workflow possono avvenire su un commit specifico e lo stato di esecuzione di ogni workflow in CodeCatalyst si riflette anche come parte dello stato di commit in GitHub, Bitbucket o. GitLab Quando viene inviato un nuovo commit, i nuovi [stati di esecuzione](#) del flusso di lavoro si riflettono in GitHub Bitbucket o per quel nuovo commit. GitLab Se esegui

nuovamente un flusso di lavoro per un commit, il nuovo stato di esecuzione del flusso di lavoro sostituisce lo stato precedente di quel commit e flusso di lavoro.

Puoi impostare regole di protezione branch in GitHub o Bitbucket per bloccare un'unione di richieste pull oppure in GitLab per bloccare una richiesta di unione, quando l'ultimo commit ha lo stato di esecuzione del workflow non riuscita. Con le regole di protezione delle filiali, lo stato del commit più recente influisce sulla possibilità di unire una pull request in GitHub, Bitbucket o. GitLab Per ulteriori informazioni sui flussi di lavoro, consulta e. [Esecuzione di un workflow](#) [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#)

A seconda del provider di repository di terze parti che stai utilizzando, consulta quanto segue:

- GitHub repositories: GitHub è la documentazione [sui controlli di stato](#) e [sulle filiali protette](#).
- Repository Bitbucket: la documentazione di Bitbucket su [Usare le autorizzazioni delle filiali e Take control with branch permissions](#) in Bitbucket Cloud.
- GitLab repository : [è la documentazione per Auto merge e Protected branch. GitLab](#)

Collegamento dei problemi di Jira alle richieste pull CodeCatalyst

Puoi collegare le richieste pull create in un repository di CodeCatalyst sorgenti ai problemi di Jira. Dopo aver collegato un problema di Jira, il problema viene visualizzato come proprietà della pull request. Di conseguenza, gli eventi di pull request, gli eventi di workflow e gli eventi di distribuzione vengono inviati a Jira e aggiunti al problema di Jira. Le richieste pull possono essere collegate a uno o più problemi di Jira. Puoi collegare solo le richieste pull che si trovano in un repository di CodeCatalyst origine, non quelle in un repository di terze parti. GitHub Prima di poter collegare i problemi di Jira a una pull request, il tuo progetto Jira deve essere collegato al progetto. CodeCatalyst Per ulteriori informazioni sul collegamento di un progetto Jira a un CodeCatalyst progetto, consulta. [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#)

Note

Non puoi creare una pull request senza un repository di sorgenti con due rami nel progetto. CodeCatalyst Per ulteriori informazioni sulle richieste pull, consulta [Working with pull request in CodeCatalyst](#).

Per collegare i problemi di Jira a una CodeCatalyst pull request

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo CodeCatalyst progetto.
3. Nel riquadro di navigazione, scegli Code, quindi scegli Pull request.
4. Scegli Crea richiesta pull per inserire i dettagli della pull request.
5. Dal menu a discesa Source repository, scegli il repository di origine in cui desideri collegare una pull request.
6. Dal menu a discesa Source branch, scegli il ramo che contiene le modifiche che desideri rivedere.
7. Dal menu a discesa del ramo di destinazione, scegli il ramo in cui desideri unire le modifiche riviste.
8. Nel campo di immissione del testo del titolo della Pull request, inserisci il titolo delle tue pull request.
9. Scegli Link issues per i problemi di Jira - campo opzionale, scegli il menu a discesa e cerca i problemi di Jira che desideri aggiungere dal progetto Jira collegato.
10. Seleziona i problemi di Jira che desideri aggiungere alla pull request.
11. Scegli Crea per creare la pull request.

Dopo aver collegato i problemi di Jira a una CodeCatalyst pull request, è disponibile un riepilogo della pull request. Il riepilogo include le esecuzioni del flusso di lavoro, i problemi collegati, i revisori obbligatori, i revisori opzionali e l'autore.

Note

Le informazioni Assegnatario e Creato da associate al problema di Jira non sono disponibili in CodeCatalyst

Dopo aver collegato una pull request, il CodeCatalyst progetto sincronizzato e il progetto Jira consentono agli aggiornamenti di riflettersi nel tuo progetto Jira. CodeCatalyst Lo stato della pull request collegata e tutti gli eventi del flusso di lavoro relativi alla pull request verranno visualizzati nel numero di Jira durante la visualizzazione in Jira. Per ulteriori informazioni sulla visualizzazione CodeCatalyst degli eventi in Jira, consulta [Visualizzazione CodeCatalyst degli eventi nei problemi di Jira](#)

Visualizzazione CodeCatalyst degli eventi nei problemi di Jira

Se i tuoi CodeCatalyst progetti e i progetti Jira sono collegati, lo stato riepilogativo della pull request e lo stato degli eventi del CodeCatalyst flusso di lavoro associati si riflettono nel problema di Jira. Ad esempio, se chiudi o unisci una pull request CodeCatalyst, l'aggiornamento dello stato si riflette nel numero di Jira. CodeCatalyst CI/CD gli eventi del flusso di lavoro relativi a una CodeCatalyst pull request sono sincronizzati, quindi un'esecuzione corretta del flusso di lavoro verrebbe inviata anche al problema di Jira.

Per visualizzare CodeCatalyst gli eventi in un numero di Jira

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Vai al tuo CodeCatalyst progetto.
3. Nel pannello di CodeCatalyst navigazione, scegli Codice, scegli Richieste Pull, quindi scegli la pull request con il problema Jira che desideri visualizzare nel tuo progetto Jira.
4. Nel riquadro Informazioni aggiuntive, scegli il problema di Jira che desideri visualizzare nel tuo progetto Jira.
5. Dal riquadro Dettagli del progetto Jira, seleziona la pull request elencata per Sviluppo per visualizzare i dettagli della pull request.
6. (Facoltativo) Per vedere le build più recenti, scegli la scheda Builds.
7. (Facoltativo) Per vedere lo stato dello sviluppo, scegli la scheda Implementazioni.

Cerca codice, problemi, progetti e utenti in CodeCatalyst

Usa la barra di ricerca o una finestra dedicata CodeCatalyst ai risultati di ricerca per cercare codice, problemi, progetti e utenti CodeCatalyst.

Puoi trovare risorse nel tuo spazio e nei tuoi progetti inserendo domande come nome, descrizione e stato nella barra di ricerca. Puoi anche affinare le query di ricerca utilizzando il linguaggio delle query di ricerca.

Argomenti

- [Perfezionamento della query di ricerca](#)
- [Considerazioni sull'utilizzo della ricerca](#)
- [Riferimento ai campi ricercabili](#)

Per effettuare una ricerca

1. Nella barra di ricerca nella barra di navigazione in alto, inserisci una query di ricerca.
2. (Facoltativo) Perfeziona la query di ricerca utilizzando CodeCatalyst il linguaggio di ricerca utilizzato. Per ulteriori informazioni, consulta [Perfezionamento della query di ricerca](#).
3. Esegui una di queste operazioni:
 - Per cercare risorse all'interno del progetto in cui ti trovi attualmente, scegli Questo progetto.
 - Per cercare risorse all'interno di tutti i progetti nello spazio in cui ti trovi attualmente, scegli Questo spazio.
4. Visualizza i risultati della ricerca in una finestra dedicata dei risultati di ricerca effettuando una delle seguenti operazioni:
 - Nella parte inferiore della finestra dei risultati della ricerca rapida, scegli Visualizza tutti i risultati in project-name | space-name per visualizzare tutti i risultati della ricerca.
 - Premi Invio per visualizzare tutti i risultati della ricerca.

Tip

Menziona gli altri utenti del progetto in un commento o nella descrizione di una pull request, oppure in un commento o descrizione del problema, utilizzando il segno @ seguito dal nome

visualizzato o dal nome utente. Puoi anche creare link a risorse come problemi o file di codice usando il segno @ seguito dal nome del problema o del file di codice.

Perfezionamento della query di ricerca

Se non riesci a trovare ciò che stai cercando dopo la ricerca, puoi affinare la ricerca utilizzando il linguaggio CodeCatalyst di query specializzato. I singoli campi non hanno limiti di caratteri, ma la query complessiva ha un limite di 1.024 caratteri.

Argomenti

- [Perfezionamento per tipo](#)
- [Perfezionamento per campo](#)
- [Perfezionamento con operatori booleani](#)
- [Perfezionamento per progetto](#)

Perfezionamento per tipo

Per restringere l'ambito della ricerca in base a un tipo specifico di informazioni, includi *type:result-type* nella ricerca dove si *result-type* trova `code`, `issueproject`, o `user`

Esempi:

- `type:code AND java`— Mostra i risultati del codice nei campi relativi al codice che contengono «java».

Per ulteriori informazioni, consulta [Campi di codice](#).

- `type:issue AND Bug`— Mostra i risultati dei problemi nei campi relativi al problema che contengono «Bug».

Per ulteriori informazioni, consulta [Campi del problema](#).

- `type:user AND MaryMajor`— Mostra i risultati degli utenti nei campi relativi all'utente che contengono «». MaryMajor

Per ulteriori informazioni, consulta [Campi utente](#).

- `type:project AND Datafeeder`— Mostra i risultati del progetto che contengono «Datafeeder».

Per ulteriori informazioni, consulta [Campi del progetto](#).

Perfezionamento per campo

Per restringere l'ambito della ricerca a un campo specifico, includi *field-name:query* nella ricerca, dove si *field-name* trova `title`, `username`, `project`, `description`, e così via, e *query* indica il testo che stai cercando. Per un elenco di campi, consulta [Riferimento ai campi ricercabili](#). È possibile cercare più interrogazioni utilizzando le parentesi.

Esempi:

- `title:bug`— Mostra i risultati in cui il titolo contiene «bug».
- `username:John`— Mostra i risultati in cui il nome utente contiene «John».
- `project:DataFeeder`— Mostra i risultati del progetto «DataFeeder». La query non distingue tra maiuscole e minuscole.
- `description:overview`— Mostra i risultati in cui la descrizione contiene «panoramica».

Perfezionamento con operatori booleani

Per specificare i vincoli sulle frasi di ricerca, è possibile utilizzare gli operatori booleani `AND` `OR` `NOT`. Se elenchi più frasi, CodeCatalyst per impostazione predefinita le unisce. `OR` Puoi raggruppare le frasi di ricerca usando le parentesi.

- `exception AND type:code`— Mostra solo i risultati del codice per «eccezione».
- `path:README.md AND repo:ServerlessAPI`— Mostra i risultati per i percorsi con «README.md» in cui il repository è denominato «ServerlessAPI».
- `buildspec.yml AND (repo:ServerlessAPI OR ServerlessWebApp)`— Mostra i risultati per «buildspec.yml» dove il repository è «ServerlessAPI» o «». `ServerlessWebApp`
- `path:java NOT (path:py OR path:ts)`— Mostra i risultati in cui il percorso contiene «java» ma non «py» o «ts».

Perfezionamento per progetto

Per restringere l'ambito della ricerca a un progetto specifico, includi *project:name AND query* nella ricerca *name* dov'è il progetto in cui stai cercando e *query* il contenuto che stai cercando.

- `project:name AND query`—Mostra i risultati in cui il percorso contiene la query e il nome del progetto.

Considerazioni sull'utilizzo della ricerca

Aggiornamenti ritardati dei contenuti: possono essere necessari diversi minuti prima che gli aggiornamenti dei contenuti, ad esempio modifiche di nome o riassegnazioni di problemi, si riflettano nei risultati della ricerca. Gli aggiornamenti di grandi dimensioni, come la migrazione di una base di codice, possono richiedere più tempo per essere visualizzati nei risultati di ricerca.

Esclusione dei caratteri speciali: i seguenti caratteri speciali richiedono un'attenzione speciale nelle query di ricerca: `. + - & & | | ! ( ) { } [ ] ^ " ~ * ? : \`. I caratteri speciali non influiranno sulla query, ed è necessario rimuoverli o evitarli. Per sfuggire a un carattere, aggiungete una barra rovesciata (`\`) davanti ad esso. Ad esempio, la query di ricerca `[Feature]` dovrebbe essere `Feature o \ [Feature\]`.

Restringimento della ricerca: la ricerca non fa distinzione tra maiuscole e minuscole. La ricerca esclusivamente in lettere minuscole evita che le query suddividano le parole in caso di modifica tra maiuscole e minuscole. Ad esempio, per eseguire una query solo per `MyService` e `soloMyService`, è consigliabile eseguire una query per `myservice` evitare risultati che contengono solo `o. my service`.

Per impostazione predefinita, la ricerca unisce parole e parti di parole con la congiunzione OR-wise. Ad esempio, `new function` potrebbe restituire risultati contenenti entrambi `new function` e anche risultati con solo `o. new function`. Per evitare quest'ultimo problema, combina più parole con `AND`. Ad esempio, puoi cercare `new AND function`.

Rami predefiniti: la ricerca restituirà solo i risultati del codice dell'ultimo commit sul ramo predefinito di un repository di origine. Per trovare codice su altri branch o commit, prendi in considerazione la possibilità di [clonare il repository localmente](#), [aprire il ramo in un ambiente di sviluppo](#) o [visualizzare i rami e i dettagli](#) nell'interfaccia utente. CodeCatalyst La modifica del ramo predefinito comporta l'aggiornamento dei file individuabili tramite la ricerca. Per ulteriori informazioni, consulta [Gestione del ramo predefinito per un repository](#).

Important

CodeCatalyst non supporta il rilevamento delle modifiche nel ramo predefinito per gli archivi collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima

scollegarlo CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Come procedura consigliata, assicurati sempre di disporre della versione più recente dell'estensione prima di collegare un repository.

Riferimento ai campi ricercabili

CodeCatalyst esegue la ricerca nei seguenti campi quando si inseriscono delle query di ricerca. Gli alias sono un altro nome che è possibile utilizzare per fare riferimento al campo nel linguaggio di interrogazione avanzato.

Campi di codice

Campo	Alias	Descrizione
Nome del ramo	ramo	Nome del ramo in cui si trova il file di codice.
code	N/D	Informazioni sul contenuto del codice sotto forma di frammenti di codice che indicano parti del codice sorgente corrispondenti alla ricerca.
CommitID	N/D	ID di commit del commit in cui il file di codice restituito è stato aggiornato l'ultima volta. Può o meno essere l'ID di commit all'inizio del nome del ramo specificato in <code>branchName</code> .
Messaggio di commit	N/D	Messaggio di commit del commit in cui il file di codice è stato aggiornato l'ultima volta. Può o meno essere il

Campo	Alias	Descrizione
		messaggio di commit all'inizio o del nome del ramo specificato in <code>branchName</code> . Se non è stato fornito alcun messaggio di commit, questo valore sarà una stringa vuota.
<code>filePath</code>	<code>path</code>	Percorso del file di codice.
<code>lastUpdatedBy</code>	N/D	CodeCatalyst utente che ha aggiornato per ultimo il file di codice. Se il nome utente non è disponibile, questo valore sarà l'indirizzo email dell'utente come configurato nel file di configurazione Git.
<code>lastUpdatedById</code>	N/D	ID univoco generato dal sistema dell'utente che ha aggiornato per l'ultima volta il file di codice. Se l>ID utente non è disponibile, questo valore potrebbe essere l'indirizzo e-mail dell'utente.
<code>lastUpdatedTime</code>	N/D	Ora in cui i dati di ricerca sono stati aggiornati l'ultima volta con il commit che conteneva il file di codice (nel timestamp del tempo universale coordinato o (UTC)).
<code>projectId</code>	N/D	ID univoco del progetto generato dal sistema.

Campo	Alias	Descrizione
projectName	ProjectNames, progetto	Visualizza il nome del progetto che contiene l'archivio dei sorgenti in cui è stato eseguito il commit del file di codice.
RepositoryID	Repoid	ID univoco del repository di origine generato dal sistema.
Nome del repository	repository, repo	Visualizza il nome del repository di origine in cui è stato eseguito il commit del file di codice.

Campi del problema

Campo	Alias	Descrizione
ID assegnatari	ID assegnatario	Identificazione univoca generata dal sistema IDs degli utenti assegnati al problema.
assegnatari	cessionario	Nomi utente degli utenti assegnati al problema.
Creato da	N/D	Visualizza il nome dell'utente che ha creato il problema.
createdById	N/D	ID univoco generato dal sistema dell'utente che ha creato il problema.
Ora di creazione	N/D	Ora di creazione del problema (nel timestamp del tempo universale coordinato (UTC)).
description	N/D	Descrizione del problema.

Campo	Alias	Descrizione
è archiviato	archived	Valore booleano che indica se creare il problema in uno stato di archiviazione.
È bloccato	blocked	Valore booleano che indica se il problema è contrassegnato come bloccato.
LabelID	ID etichetta	Etichette uniche generate dal sistema per IDs un numero.
lastUpdatedBy	N/D	Visualizza il nome dell'utente che ha aggiornato per ultimo il problema.
lastUpdatedById	N/D	ID univoco generato dal sistema dell'utente che ha aggiornato per ultimo il problema.
lastUpdatedTime	N/D	Ora dell'ultimo aggiornamento del problema (nel timestamp del tempo universale coordinato o (UTC)).
priority	N/D	Priorità del problema, se ne è stata assegnata una.
projectId	N/D	ID univoco del progetto generato dal sistema.
projectName	ProjectNames, progetto	Progetto in cui è possibile trovare questo problema.
ShortID	N/D	Identificatore abbreviato e ad incremento automatico del problema.

Campo	Alias	Descrizione
status	N/D	Stato del problema che indica se il problema è in arretrato o nella colonna della scheda.
ID di stato	N/D	Identificatore di sistema dello stato.
titolo	N/D	Titolo del problema.

Campi del progetto

Campo	Alias	Description
description	N/D	Descrizione del progetto.
lastUpdatedTime	N/D	Ora dell'ultimo aggiornamento dei metadati del progetto (nel timestamp del tempo universal e coordinato (UTC)).
projectName	project	Nome del progetto nello spazio.
Percorso del progetto	N/D	Nome del progetto indirizzabile tramite URL, definito durante la creazione del progetto. Utilizzato in URLs questo caso richiede il nome del progetto.

Campi utente

Campo	Alias	Descrizione
displayName	N/D	Nome usato per l'utente in CodeCatalyst. I nomi visualizzati non sono univoci.
e-mail	N/D	Indirizzo e-mail dell'utente.
lastUpdatedTime	N/D	Ora dell'ultimo aggiornamento dei metadati dell'utente (nel timestamp dell'ora universale coordinata (UTC)).
userName	username	Nome utente scelto dall'utente al momento della registrazione. CodeCatalyst A differenza dei nomi visualizzati, i nomi utente non possono essere modificati.

Risoluzione dei problemi con Amazon CodeCatalyst

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni in CodeCatalyst. Puoi anche utilizzare il report sullo stato di salute di Amazon per determinare se ci sono problemi di servizio che potrebbero influire sulla tua esperienza.

Argomenti

- [Risoluzione dei problemi generali di accesso](#)
- [Risoluzione dei problemi di supporto](#)
- [Amazon CodeCatalyst non è disponibile in tutto o in parte](#)
- [Non riesco a creare un progetto in CodeCatalyst](#)
- [Non riesco ad accedere al mio spazio BID come nuovo utente o non posso essere aggiunto come nuovo utente SSO a causa del nome utente troncato](#)
- [L'aggiunta di un utente SSO al mio spazio federato come nuovo utente ha creato un utente duplicato](#)
- [Voglio inviare un feedback in CodeCatalyst](#)
- [Risoluzione dei problemi relativi ai repository di origine](#)
- [Risoluzione dei problemi relativi a progetti e blueprint](#)
- [Risoluzione dei problemi con gli ambienti di sviluppo](#)
- [Risoluzione dei problemi relativi ai flussi di lavoro](#)
- [Risoluzione dei problemi relativi ai problemi](#)
- [Risoluzione dei problemi relativi alla ricerca in CodeCatalyst](#)
- [Risoluzione dei problemi relativi alle estensioni](#)
- [Risoluzione dei problemi relativi agli account associati al tuo spazio](#)
- [Risoluzione dei problemi tra Amazon CodeCatalyst e il AWS SDKs AWS CLI](#)

Risoluzione dei problemi generali di accesso

Ho dimenticato la mia password

Problema: ho dimenticato la password che uso per il mio AWS Builder ID e Amazon CodeCatalyst

Possibili correzioni: il modo più semplice per risolvere questo problema è reimpostare la password.

1. Apri [Amazon CodeCatalyst](#) e inserisci il tuo indirizzo e-mail. Quindi, scegli Continue (Continua).
2. Scegli Password dimenticata?
3. Ti invieremo un'email con un link per modificare la password. Se non vedi l'email nella tua casella di posta, controlla la cartella spam.

Amazon CodeCatalyst non è disponibile in tutto o in parte

Problema: ho visitato o seguito un collegamento alla CodeCatalyst console, ma vedo un errore.

Possibili correzioni: i motivi più comuni di questo problema sono il fatto che hai seguito un link a un progetto o a uno spazio a cui non sei stato invitato oppure che c'è un problema di disponibilità generale del servizio. Controlla il [rapporto Health](#) per vedere se ci sono problemi noti con il servizio. In caso contrario, contatta la persona che ti ha invitato al progetto o allo spazio e chiedi un altro invito. Se non sei stato invitato a partecipare a nessun progetto o spazio, puoi registrarti e [creare il tuo spazio e i tuoi progetti](#).

Non riesco a creare un progetto in CodeCatalyst

Problema: voglio creare un progetto, ma il pulsante Crea progetto risulta non disponibile oppure ricevo un messaggio di errore.

Possibili correzioni: i motivi più comuni di questo problema sono che hai effettuato l'accesso alla console con un AWS Builder ID che non ha il ruolo di amministratore dello spazio. È necessario disporre di questo ruolo per creare progetti in uno spazio.

Se hai questo ruolo e il pulsante non appare come disponibile, potrebbe esserci un problema transitorio con il servizio. Aggiorna il browser e riprova.

Risoluzione dei problemi di supporto

Ricevo un errore quando accedo Supporto ad Amazon CodeCatalyst

Problema: quando scelgo l' CodeCatalyst opzione Supporto per Amazon, ricevo il seguente messaggio di errore:

Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the Account AWS that is the billing account for the space.

Possibili correzioni: aggiungi il ruolo richiesto all'account Account AWS di fatturazione per lo spazio. L'account designato come account di fatturazione per lo spazio utilizza il `AWSRoleForCodeCatalystSupport` ruolo e `AmazonCodeCatalystSupportAccess` la politica gestita. Per ulteriori informazioni, consulta [Creazione del `AWSRoleForCodeCatalystSupport` ruolo per il tuo account e il tuo spazio](#).

Note

Un AWS Builder ID può ricevere supporto solo per l'alias con cui è autenticato e solo per le risorse basate sulle autorizzazioni in cui è registrato. CodeCatalyst Il supporto per account e fatturazione è disponibile per tutti gli utenti dello spazio. Tuttavia, i builder possono ottenere supporto solo per le risorse e le informazioni per cui dispongono delle autorizzazioni. CodeCatalyst

Non riesco a creare casi di supporto tecnico per il mio spazio

Problema: non riesco a creare casi di supporto tecnico per il mio spazio.

Correzioni: è necessario aggiungere un piano Business Support o Enterprise Support all'account di fatturazione dello spazio per consentire agli utenti presenti nello spazio di creare richieste di supporto tecnico. Chiedi all'amministratore dello spazio di aggiungere un Supporto piano al tuo account di fatturazione dello spazio o visita <https://repost.aws/> per chiedere alla community. AWS

Il mio account per i casi di assistenza non è più collegato al mio spazio in CodeCatalyst

Problema: il mio account per i casi di assistenza non è più collegato al mio spazio in CodeCatalyst.

Correzioni: se un utente con il ruolo di amministratore dello spazio cambia l'account di fatturazione dello spazio, il Supporto piano e tutti i casi associati verranno disconnessi dallo spazio. I Supporto casi associati al vecchio account di fatturazione spaziale non saranno più visibili in Supporto Amazon CodeCatalyst. L'utente root di quell'account di fatturazione può visualizzare e risolvere vecchi casi da Console di gestione AWS e può configurare le autorizzazioni IAM Supporto per consentire ad

altri utenti di visualizzare e risolvere i vecchi casi. Non potrai continuare a ricevere supporto tecnico CodeCatalyst dal vecchio account di fatturazione Space tramite Console di gestione AWS, ma potrai ricevere supporto tecnico per altri servizi fino all'annullamento del Supporto piano.

Per ulteriori informazioni, consulta [Aggiornamento, risoluzione e riapertura del caso nella Guida per l'utente](#).Supporto

Non riesco ad aprire un caso di supporto per un altro Servizio AWS in Supporto Amazon CodeCatalyst

Problema: non riesco ad aprire una richiesta di supporto per un altro Servizio AWS Supporto utente CodeCatalyst.

Possibili correzioni: puoi aprire i casi di CodeCatalyst supporto solo da Supporto for CodeCatalyst. Se hai bisogno di supporto per servizi o risorse distribuiti CodeCatalyst da un altro AWS, Amazon o altro servizio di terze parti, dovrai creare un caso tramite il canale di assistenza Console di gestione AWS o il canale di assistenza di terze parti. Per ulteriori informazioni, consulta [Creazione di casi di supporto e gestione dei casi](#) nella Guida per l'Supporto utente.

Amazon CodeCatalyst non è disponibile in tutto o in parte

Problema: ho visitato o seguito un collegamento alla CodeCatalyst console, ma vedo un errore.

Possibili correzioni: i motivi più comuni di questo problema sono il fatto che hai seguito un link a un progetto o a uno spazio a cui non sei stato invitato oppure che c'è un problema di disponibilità generale del servizio. Controlla il [rapporto Health](#) per vedere se ci sono problemi noti con il servizio. In caso contrario, contatta la persona che ti ha invitato al progetto o allo spazio e chiedi un altro invito. Se non sei stato invitato a partecipare a nessun progetto o spazio, puoi registrarti e [creare il tuo spazio e i tuoi progetti](#).

Non riesco a creare un progetto in CodeCatalyst

Problema: voglio creare un progetto, ma il pulsante Crea progetto risulta non disponibile oppure ricevo un messaggio di errore.

Possibili correzioni: i motivi più comuni di questo problema sono che hai effettuato l'accesso alla console con un AWS Builder ID che non ha il ruolo di amministratore dello spazio. È necessario disporre di questo ruolo per creare progetti in uno spazio.

Se hai questo ruolo e il pulsante non appare come disponibile, potrebbe esserci un problema transitorio con il servizio. Aggiorna il browser e riprova.

Non riesco ad accedere al mio spazio BID come nuovo utente o non posso essere aggiunto come nuovo utente SSO a causa del nome utente troncato

Problema: CodeCatalyst tronca i nomi utente dopo 100 caratteri, il che può far apparire alcuni nomi utente identici. Come nuovo utente che accede a uno CodeCatalyst spazio, riscontro questo problema a seconda del tipo di spazio, come segue:

- Ho un AWS Builder ID che voglio usare per accedere. CodeCatalyst Quando tento di accedere allo spazio, ricevo un messaggio che indica che il mio nome utente non è valido.
- Sono un amministratore di identità federato per uno CodeCatalyst spazio che supporta la federazione delle identità. Quando aggiungo un nuovo utente agli utenti e ai gruppi SSO in IAM Identity Center, ricevo un messaggio che indica che l'utente non è valido.

Possibili correzioni: il primo utente che accede CodeCatalyst o viene aggiunto allo spazio come utente SSO con un determinato nome utente troncato avrà successo. Tutti gli utenti che si registrano con un ID AWS Builder o che vengono aggiunti successivamente a IAM Identity Center non potranno accedere perché il nome sembrerà un duplicato. A seconda del tipo di spazio, esegui una delle seguenti operazioni:

- Per poter accedere allo spazio AWS Builder ID, registrati con un nome utente diverso.
- Per poter aggiungere il nuovo utente in IAM Identity Center, aggiungi l'utente con un nome utente diverso.

Note

Anche se il nome utente sembra troncato, viene CodeCatalyst mappato all'identità in un modo che non è influenzato dai nomi utente con nomi troncati. Tuttavia, se viene creato un nome utente uguale a un nome utente troncato, tale nome utente non sarà disponibile se un altro utente associato (con lo stesso spazio o applicazione IAM Identity Center) si è già unito con quel nome utente troncato. CodeCatalyst

L'aggiunta di un utente SSO al mio spazio federato come nuovo utente ha creato un utente duplicato

Problema: gli utenti CodeCatalyst SSO aggiunti a uno CodeCatalyst spazio e poi rimossi potrebbero riscontrare un problema a causa del tentativo di riutilizzo del nome utente. Ciò potrebbe causare un errore simile al seguente, in cui non è consentito duplicare l'utente anche se è stato ricreato.

Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the Account AWS that is the billing account for the space.

Possibili correzioni: se un utente IDC esistente viene eliminato e quindi viene ricreato un nuovo utente con lo stesso nome utente, il nuovo utente non è in grado di accedere a causa del conflitto tra il nome utente e quello precedente. Dopo aver aggiunto un nome utente a uno spazio come utente SSO, il nome non può essere riutilizzato. Tutti gli utenti che si registrano con un ID AWS Builder o vengono aggiunti successivamente in IAM Identity Center non potranno accedere perché il nome sembrerà un duplicato.

A seconda del tipo di spazio, esegui una delle seguenti operazioni:

- Per poter accedere allo spazio AWS Builder ID, registrati con un nome utente diverso.
- Per poter aggiungere il nuovo utente in IAM Identity Center, aggiungi l'utente con un nome utente diverso.

Voglio inviare un feedback in CodeCatalyst

Problema: ho trovato un bug nel CodeCatalyst file e voglio inviare un feedback.

Possibili correzioni: puoi inviare feedback direttamente in CodeCatalyst.

1. Apri la CodeCatalyst console all'indirizzo <https://codecatalyst.aws/>.
2. Nel riquadro di navigazione, scegli Invia feedback.
3. Scegli il tipo di feedback dal menu a discesa e inserisci il tuo feedback.

Risoluzione dei problemi relativi ai repository di origine

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni relativi ai repository di origine in CodeCatalyst.

Argomenti

- [Ho raggiunto lo spazio di archiviazione massimo disponibile e visualizzo avvisi o errori](#)
- [Ricevo un errore durante il tentativo di clonare o eseguire il push su un repository di CodeCatalyst sorgenti Amazon](#)
- [Ricevo un errore durante il tentativo di eseguire il commit o il push su un repository di CodeCatalyst sorgenti Amazon](#)
- [Ho bisogno di un repository di sorgenti per il mio progetto](#)
- [Il mio repository di origine è nuovo di zecca ma contiene un commit](#)
- [Voglio un ramo diverso come ramo predefinito](#)
- [Ricevo e-mail sull'attività nelle pull request](#)
- [Ho dimenticato il mio token di accesso personale \(PAT\)](#)
- [Una pull request non mostra le modifiche che mi aspetto](#)
- [Una pull request mostra lo stato Non unibile](#)

Ho raggiunto lo spazio di archiviazione massimo disponibile e visualizzo avvisi o errori

Problema: voglio salvare il codice in uno o più repository di sorgenti in CodeCatalyst, ma vedo un errore. Nella console, nella pagina del repository di origine viene visualizzato un messaggio che indica che ho raggiunto il limite di spazio di archiviazione per lo spazio.

Possibili correzioni: a seconda del ruolo che ricopri nel progetto o nello spazio, puoi ridurre le dimensioni di uno o più repository di origine, eliminare gli archivi di origine non utilizzati o modificare il livello di fatturazione con uno che abbia più spazio di archiviazione.

- Per ridurre le dimensioni di un archivio di origine in un progetto, puoi eliminare i rami inutilizzati. Per ulteriori informazioni, consulta [Eliminazione di un ramo](#) e [Ruolo di collaboratore](#).
- Per ridurre lo spazio di archiviazione complessivo, è possibile eliminare gli archivi di origine non utilizzati. Per ulteriori informazioni, consulta [Eliminazione di un archivio di origine](#) e [Ruolo di amministratore del progetto](#).

- Per aumentare la quantità di spazio di archiviazione disponibile, puoi modificare il livello di fatturazione impostando uno con più spazio di archiviazione. Per ulteriori informazioni, consulta [Modifica del livello di CodeCatalyst fatturazione](#) nella Amazon CodeCatalyst Administrator Guide.

Ricevo un errore durante il tentativo di clonare o eseguire il push su un repository di CodeCatalyst sorgenti Amazon

Problema: quando provo a clonare un archivio di sorgenti su un computer locale o in un ambiente di sviluppo integrato (IDE), ricevo un errore di autorizzazione.

Possibili correzioni: potresti non avere un token di accesso personale (PAT) per il tuo AWS Builder ID, potresti non aver configurato il tuo sistema di gestione delle credenziali con il tuo PAT o il PAT potrebbe essere scaduto. Prova una o più delle seguenti soluzioni:

- Crea un token di accesso personale (PAT). Per ulteriori informazioni, consulta [Concedi agli utenti l'accesso al repository con token di accesso personali](#).
- Assicurati di aver accettato un invito al progetto che contiene il repository dei sorgenti e di essere ancora un membro di quel progetto. Non puoi clonare un repository di origine se non sei un membro attivo di quel progetto. Accedi alla console e prova a navigare nello spazio e nel progetto in cui stai tentando di clonare un repository di origine. Se non riesci a visualizzare il progetto nell'elenco dei progetti dello spazio, significa che non sei membro di quel progetto o non hai accettato un invito a partecipare a quel progetto. Per ulteriori informazioni, consulta [Accettazione di un invito e creazione di un AWS Builder ID](#).
- Assicurati che il comando clone sia formattato correttamente e includa il tuo ID AWS Builder. Per esempio:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

- Usa il AWS CLI per assicurarti di avere un PAT associato al tuo AWS Builder ID e che non sia scaduto. Se non ne hai uno o il PAT è scaduto, creane uno. Per ulteriori informazioni, consulta [Concedi agli utenti l'accesso al repository con token di accesso personali](#).
- Prova a creare un ambiente di sviluppo per lavorare con il codice nel repository di origine invece di clonarlo in un repository o IDE locale. Per ulteriori informazioni, consulta [Creazione di un ambiente di sviluppo](#).

Ricevo un errore durante il tentativo di eseguire il commit o il push su un repository di CodeCatalyst sorgenti Amazon

Problema: quando provo a inviare un push a un repository di origine, ricevo un errore di autorizzazione.

Possibili correzioni: potresti non avere un ruolo nel progetto che ti consenta di eseguire il commit e inviare modifiche al codice del progetto. Visualizza il tuo ruolo nel progetto in cui stai cercando di inviare le modifiche a un archivio di origine. Per ulteriori informazioni, consulta [Ottenere un elenco dei membri e dei relativi ruoli nel progetto](#) e [Concessione dell'accesso con ruoli utente](#).

Se hai un ruolo che consente di eseguire e inviare modifiche, nel ramo in cui stai tentando di apportare modifiche potrebbe essere configurata una regola di branch che ti impedisce di inviare modifiche al codice in quel ramo. Prova invece a creare un ramo e a inviare il codice a quel ramo. Per ulteriori informazioni, consulta [Gestisci le azioni consentite per una filiale con le regole della filiale](#).

Ho bisogno di un repository di sorgenti per il mio progetto

Problema: il mio progetto non ha un repository di sorgenti o ho bisogno di un altro repository di sorgenti per il mio progetto.

Possibili correzioni: alcuni progetti vengono creati senza risorse. Se sei un membro del progetto, puoi creare repository di sorgenti per quel progetto in CodeCatalyst. Se qualcuno con il ruolo di amministratore dello spazio installa i GitHub repository e li collega a un GitHub account, puoi collegarti ai GitHub repository disponibili per aggiungerli al tuo progetto, se hai il ruolo di amministratore del progetto. Per ulteriori informazioni, vedere [Creazione di un repository di origine e Collegamento di un repository](#) di origine.

Il mio repository di origine è nuovo di zecca ma contiene un commit

Problema: ho appena creato un repository di sorgenti. Dovrebbe essere vuoto, ma contiene un commit, un ramo e un README.md file.

Possibili correzioni: si tratta di un comportamento previsto. Tutti i repository di origine CodeCatalyst includono un commit iniziale che imposta il ramo predefinito su `main` e include codice di esempio (se il repository è stato creato per un progetto utilizzando un blueprint che includeva codice di esempio) o un file markdown modello per un file README del repository. Puoi creare rami aggiuntivi nella console e nei client Git. È possibile creare e modificare file nella console ed eliminare file negli ambienti di sviluppo e nei client Git.

Voglio un ramo diverso come ramo predefinito

Problema: il mio repository di origine è stato fornito con un ramo predefinito denominato `main`, ma voglio un ramo diverso come ramo predefinito.

Possibili correzioni: non è possibile modificare o eliminare il ramo predefinito nei repository di origine in CodeCatalyst. È possibile creare rami aggiuntivi e utilizzarli nelle azioni di origine dei flussi di lavoro. Puoi anche scegliere di collegare i GitHub repository e usarli come repository per il tuo progetto.

Ricevo e-mail sull'attività nelle pull request

Problema: non mi sono registrato né configurato le notifiche e-mail sull'attività delle pull request, ma le ricevo comunque.

Possibili correzioni: le notifiche e-mail vengono inviate automaticamente sulle attività di pull request. Per ulteriori informazioni, consulta [Revisione del codice con richieste pull in Amazon CodeCatalyst](#).

Ho dimenticato il mio token di accesso personale (PAT)

Problema: ho usato un PAT per clonare, inviare ed estrarre codice per i repository dei sorgenti, ma ho perso il valore del mio token e non riesco a trovarlo nella console. CodeCatalyst

Possibili correzioni: il modo più rapido per risolvere questo problema consiste nel creare un altro PAT e configurare il gestore delle credenziali o l'IDE per utilizzare questo nuovo PAT. Visualizziamo il valore di un PAT solo quando lo crei. Se perdi questo valore, non può essere recuperato. Per ulteriori informazioni, consulta [Concedi agli utenti l'accesso al repository con token di accesso personali](#).

Una pull request non mostra le modifiche che mi aspetto

Problema: ho creato una richiesta pull, ma non vedo le modifiche che mi aspetto di vedere tra i rami di origine e di destinazione.

Possibili correzioni: ciò potrebbe essere causato da una serie di problemi. Prova una o più delle seguenti soluzioni:

- È possibile che stiate esaminando le modifiche tra le revisioni precedenti o che non stiate visualizzando le modifiche più recenti. Aggiorna il browser e assicurati di aver scelto il confronto tra le revisioni che desideri visualizzare.

- Non tutte le modifiche in una pull request possono essere visualizzate nella console. Ad esempio, non è possibile visualizzare i sottomoduli Git nella console, quindi non è possibile visualizzare le differenze in un sottomodulo in una pull request. Alcune differenze potrebbero essere troppo grandi per essere visualizzate. Per ulteriori informazioni, consulta [Quote per i repository di sorgenti in CodeCatalyst](#) e [Visualizzazione di un file](#).
- Le richieste pull mostrano le differenze tra la base di unione e la revisione scelta. Quando create una richiesta pull, la differenza visualizzata per voi è la differenza tra la punta del ramo di origine e la punta del ramo di destinazione. Una volta creata la pull request, la differenza visualizzata è tra la revisione e la relativa base di unione. La base di unione è il commit che costituiva la punta del ramo di destinazione al momento della creazione della revisione. La base di unione può cambiare tra le revisioni. Per ulteriori informazioni sulle differenze e sulle basi di unione in Git, consulta [git-merge-base](#) la documentazione di Git.

Una pull request mostra lo stato Non unibile

Problema: voglio unire una pull request, ma il relativo stato è indicato come Non unificabile.

Possibili correzioni: ciò può essere causato da uno o più problemi:

- Tutti i revisori necessari per la richiesta pull devono approvare una pull request prima che possa essere unita. Controlla l'elenco dei revisori richiesti per tutti i revisori con un'icona a forma di orologio accanto al nome. L'icona a forma di orologio indica che il revisore non ha approvato la pull request.

Note

Se un revisore richiesto è stato rimosso dal progetto prima di approvare la pull request, non è possibile unire la pull request. Chiudi la pull request e crea una nuova pull request.

- Potrebbe esserci un conflitto di fusione tra il ramo di origine e il ramo di destinazione. CodeCatalyst non supporta tutte le possibili strategie e opzioni di fusione Git. Puoi valutare i rami per i conflitti di unione in un ambiente di sviluppo o clonare il repository e utilizzare uno strumento IDE o Git per trovare e risolvere i conflitti di unione. Per ulteriori informazioni, consulta [Unione di una richiesta di pull](#).

Risoluzione dei problemi relativi a progetti e blueprint

Questa sezione può aiutarti a risolvere alcuni problemi comuni che potresti riscontrare mentre lavori con progetti e blueprint in Amazon. CodeCatalyst

API Java con dipendenze del AWS Fargate blueprint mancanti per apache-maven-3.8.6

Problema: per un progetto creato dall'API Java con AWS Fargate blueprint, il flusso di lavoro ha esito negativo e viene visualizzato un errore relativo alle dipendenze mancanti. apache-maven-3.8.6 Il flusso di lavoro non riesce con un output simile al seguente esempio:

```
Step 8/25 : RUN wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-
maven-3.8.6-bin.tar.gz -P /tmp
---> Running in 1851ce6f4d1b
[91m--2023-03-10 01:24:55-- https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz
[0m[91mResolving dlcdn.apache.org (dlcdn.apache.org)...
[0m[91m151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443...
[0m[91mconnected.
[0m[91mHTTP request sent, awaiting response... [0m[91m404 Not Found
2023-03-10 01:24:55 ERROR 404: Not Found.
[0mThe command '/bin/sh -c wget https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz -P /tmp' returned a non-zero code: 8
[Container] 2023/03/10 01:24:55 Command failed with exit status 8
```

Soluzione: aggiorna il file Dockerfile del blueprint utilizzando i seguenti passaggi.

1. Nella barra di ricerca, inserisci apache-maven-3.8.6 per individuare il dockerfile all'interno del progetto creato con l'API Java con blueprint. AWS Fargate
2. Aggiorna il Dockerfile (/static-assets/app/Dockerfile) da utilizzare maven:3.9.0-amazoncorretto-11 come immagine di base e rimuovi la dipendenza dal pacchetto. apache-maven-3.8.6
3. (Consigliato) Consigliamo inoltre di aggiornare la dimensione dell'heap di Maven a 6 GB.

Di seguito è riportato un esempio di Dockerfile.

```
FROM maven:3.9.0-amazoncorretto-11 AS builder

COPY ./pom.xml ./pom.xml
COPY src ./src/

ENV MAVEN_OPTS='-Xmx6g'

RUN mvn -Dmaven.test.skip=true clean package

FROM amazoncorretto:11-alpine

COPY --from=builder target/Customerservice-0.0.1.jar Customerservice-0.0.1.jar
EXPOSE 80
CMD ["java","-jar","-Dspring.profiles.active=prod","/Customerservice-0.0.1.jar", "-server.port=80"]
```

Il moderno flusso di lavoro del blueprint di applicazioni Web a tre livelli OnPullRequest non riesce a causa di un errore di autorizzazione per Amazon CodeGuru

Problema: quando tento di eseguire un flusso di lavoro per il mio progetto, il flusso di lavoro non viene eseguito e viene visualizzato il seguente messaggio:

```
Failed at codeguru_codereview: The action failed during runtime. View the action's logs for more details.
```

Soluzione: una possibile causa di questo errore di azione potrebbe essere dovuta alla mancanza di autorizzazioni nella policy del ruolo IAM, in quanto alla versione del ruolo di servizio utilizzata da CodeCatalyst in the connected mancano le autorizzazioni necessarie per l' Account AWS esecuzione corretta dell'azione codeguru_codereview. Per risolvere questo problema, è necessario aggiornare il ruolo di servizio con le autorizzazioni richieste oppure è necessario modificare il ruolo di servizio utilizzato per il flusso di lavoro con uno con le autorizzazioni richieste per Amazon e CodeGuru Amazon CodeGuru Reviewer. Segui i passaggi seguenti, trova il tuo ruolo e aggiorna le autorizzazioni relative alla policy del ruolo per consentire al flusso di lavoro di funzionare correttamente.

Note

Questi passaggi si applicano ai seguenti flussi di lavoro in: CodeCatalyst

- Il OnPullRequest flusso di lavoro fornito per i progetti creati con il modello di applicazione Web Modern a tre livelli in. CodeCatalyst
- Flussi di lavoro aggiunti ai progetti CodeCatalyst con azioni che accedono ad Amazon CodeGuru o Amazon CodeGuru Reviewer.

Ogni progetto contiene flussi di lavoro con azioni che utilizzano un ruolo e un ambiente forniti da chi è Account AWS connesso al progetto. CodeCatalyst Il flusso di lavoro con le azioni e la relativa politica designata viene archiviato nell'archivio dei sorgenti nella directory `/.codecatalyst/workflows`. La modifica del flusso di lavoro YAML non è richiesta a meno che non si aggiunga un nuovo ID di ruolo al flusso di lavoro esistente. Per informazioni sugli elementi e sulla formattazione del modello YAML, vedere. [Definizione YAML del flusso di lavoro](#)

Questi sono i passaggi principali da seguire per modificare la politica relativa ai ruoli e verificare il flusso di lavoro YAML.

Per fare riferimento al nome del tuo ruolo nel flusso di lavoro YAML e aggiornare la policy

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio. Vai al tuo progetto.
3. Scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il flusso di lavoro intitolato. OnPullRequest Scegliere la scheda Definizione .
5. Nel workflow YAML, nel `Role` : campo sotto l'azione `codeguru_codereview`, prendi nota del nome del ruolo. Questo è il ruolo con la politica che modificherai in IAM. L'esempio seguente mostra il nome del ruolo.

CI/CD Workflows Environments Compute Secrets Change tracking

OnPullRequest mysftscvb63 test

Latest state Definition Runs

Latest run Run-9ab13 Run mode Queued Latest commit 14035d52 Workflow definition Valid

WorkflowSource mysftscvb63 test

codeguru_codereview
aws/build@v1
Environment: development

```

1 Name: OnPullRequest
2 SchemaVersion: "1.0"
3 Triggers:
4   - Type: PULLREQUEST
5     Branches:
6       - main
7     Events:
8       - OPEN
9       - REVISION
10 Actions:
11   codeguru_codereview:
12     Identifier: aws/build@v1
13     Inputs:
14       Sources:
15         - WorkflowSource
16     Variables:
17       - Name: AWS_DEFAULT_REGION
18         Value: us-west-2
19     Outputs:
20       Artifacts:
21         - Name: codereview
22         Files:
23           - ./code-guru/*
24     Configuration:
25       Steps:
26         - Run: curl -OL https://github.com/aws/aws-codeguru-cl
27         - Run: unzip aws-codeguru-cli.zip
28         - Run: export PATH=$PATH:./aws-codeguru-cli/bin
29         - Run: aws-codeguru-cli --root-dir ./src --no-prompt
30     Environment:
31       Name: development
32     Connections:
33       - Name: connection-11-30
34       Role: CodeCatalystPreviewDevelopmentAdministrator-j
  
```

Copy

6. Esegui una delle seguenti operazioni:

- (Consigliato) Aggiorna il ruolo di servizio collegato al tuo progetto con le autorizzazioni richieste per Amazon CodeGuru e Amazon CodeGuru Reviewer. Il ruolo avrà un nome `CodeCatalystWorkflowDevelopmentRole-spaceName` con un identificatore univoco aggiunto. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere [Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName* del servizio](#). Procedi ai passaggi successivi per aggiornare la policy in IAM.

Note

È necessario disporre AWS dell'accesso di amministratore a Account AWS con il ruolo e la policy.

- Cambia il ruolo di servizio utilizzato per il flusso di lavoro con uno con le autorizzazioni richieste per Amazon CodeGuru e Amazon CodeGuru Reviewer o crea un nuovo ruolo con le autorizzazioni richieste.
7. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>

Nella console IAM, trova il ruolo indicato nella fase 5, ad esempio `CodeCatalystPreviewDevelopmentRole`.
 8. Nel ruolo del passaggio 5, modifica la politica di autorizzazione per includere le `codeguru:*` autorizzazioni `codeguru-reviewer:*` e. Dopo aver aggiunto queste autorizzazioni, la politica di autorizzazione dovrebbe essere simile alla seguente:
 9. Dopo aver apportato le correzioni ai criteri, torna CodeCatalyst e riavvia l'esecuzione del flusso di lavoro.

Stai ancora cercando di risolvere il tuo problema?

Puoi andare [su Amazon CodeCatalyst](#) o compilare un [modulo Support Feedback](#). Nella sezione Richiedi informazioni, sotto Come possiamo aiutarti, inserisci che sei un CodeCatalyst cliente Amazon. Fornisci il maggior numero di dettagli possibile in modo che possiamo risolvere il problema nel modo più efficiente.

Risoluzione dei problemi con gli ambienti di sviluppo

Consulta le seguenti sezioni per risolvere i problemi relativi agli ambienti di sviluppo. Per ulteriori informazioni sugli ambienti di sviluppo, vedere. [Scrivi e modifica il codice con Dev Environments in CodeCatalyst](#)

Argomenti

- [La creazione del mio ambiente di sviluppo non è riuscita a causa di un problema con le quote](#)
- [Non riesco a trasferire le modifiche dal mio ambiente di sviluppo a un ramo specifico di un repository](#)
- [Il mio ambiente di sviluppo non è stato ripreso](#)
- [Il mio ambiente di sviluppo si è disconnesso](#)
- [Il mio ambiente di sviluppo connesso al VPC non è riuscito](#)

- [Non riesco a trovare la cartella in cui si trova il mio progetto](#)
- [Non riesco a connettermi al mio ambiente di sviluppo tramite SSH](#)
- [Non riesco a connettermi al mio ambiente di sviluppo tramite SSH perché manca la mia configurazione SSH locale](#)
- [Non riesco a connettermi al mio ambiente di sviluppo tramite SSH perché ho problemi con il mio profilo for the AWS Configcodecatalyst](#)
- [Non riesco a creare un ambiente di sviluppo quando ho effettuato l'accesso CodeCatalyst utilizzando un account Single Sign-on](#)
- [Risoluzione dei problemi con IDEs](#)
- [Risoluzione dei problemi con devfiles](#)

La creazione del mio ambiente di sviluppo non è riuscita a causa di un problema con le quote

Problema: voglio creare un ambiente di sviluppo in CodeCatalyst, ma vedo un errore. Nella console, nella pagina Dev Environments vedo un messaggio che indica che ho raggiunto il limite di spazio di archiviazione per lo spazio.

Possibili correzioni: a seconda del ruolo che ricopri nel progetto o nello spazio, puoi eliminare uno o più ambienti di sviluppo personali oppure, se hai il ruolo di amministratore dello spazio, puoi eliminare gli ambienti di sviluppo non utilizzati creati da altri utenti. Puoi anche decidere di modificare il livello di fatturazione con uno che includa più spazio di archiviazione.

- Per visualizzare i limiti di archiviazione, visualizza la scheda Fatturazione CodeCatalyst dello spazio Amazon per verificare se le quote di utilizzo hanno raggiunto il massimo consentito. Se le quote hanno raggiunto il massimo, contatta qualcuno con il ruolo di amministratore di Space per rimuovere gli ambienti di sviluppo non necessari o per prendere in considerazione la modifica del livello di fatturazione.
- Per rimuovere gli ambienti di sviluppo che hai creato e che non ti servono più, consulta [Eliminazione di un ambiente di sviluppo](#)

Se il problema persiste e ricevi un errore nell'IDE, verifica di avere un CodeCatalyst ruolo che ti consenta di creare un ambiente di sviluppo. Il ruolo di amministratore dello spazio, il ruolo di amministratore del progetto e il ruolo di collaboratore sono tutti autorizzati a creare ambienti di sviluppo. Per ulteriori informazioni, consulta [Concessione dell'accesso con ruoli utente](#).

Non riesco a trasferire le modifiche dal mio ambiente di sviluppo a un ramo specifico di un repository

Problema: voglio eseguire il commit e inviare le modifiche al codice nel mio ambiente di sviluppo a un ramo in un repository di sorgenti, ma vedo un errore.

Possibili correzioni: a seconda del ruolo che ricopri nel progetto o nello spazio, potresti non avere le autorizzazioni per inviare il codice ai repository di sorgenti del progetto. Il ruolo di amministratore dello spazio, il ruolo di amministratore del progetto e il ruolo di collaboratore sono tutti autorizzati a inviare codice ai repository del progetto.

Se hai il ruolo di Collaboratore ma non puoi inviare codice a un ramo specifico, è possibile che sia configurata una regola di branch per il ramo specifico che impedisce agli utenti con quel ruolo di inviare codice a quel particolare ramo. Prova a inviare le modifiche a un altro ramo oppure crea un ramo e poi invia il codice a quel ramo. Per ulteriori informazioni, consulta [Gestisci le azioni consentite per una filiale con le regole della filiale](#).

Il mio ambiente di sviluppo non è stato ripreso

Problema: il mio ambiente di sviluppo non è ripreso dopo averlo interrotto.

Possibili correzioni: per risolvere il problema, visualizza la scheda Fatturazione CodeCatalyst dello spazio Amazon per vedere se le quote di utilizzo hanno raggiunto i limiti massimi. Se le quote hanno raggiunto il limite massimo, contatta l'amministratore di Space per aumentare il livello di fatturazione.

Il mio ambiente di sviluppo si è disconnesso

Problema: il mio ambiente di sviluppo si è disconnesso mentre lo stavo usando.

Possibili soluzioni: per risolvere il problema, controlla la tua connessione Internet. Se non sei connesso a Internet, connettiti e riprendi a lavorare nel tuo ambiente di sviluppo.

Il mio ambiente di sviluppo connesso al VPC non è riuscito

Problema: ho associato una connessione VPC al mio ambiente di sviluppo e si verificano degli errori.

Possibili correzioni: Docker utilizza un dispositivo a livello di collegamento chiamato rete bridge che consente ai contenitori collegati alla stessa rete di bridge di comunicare. Di norma, il bridge di default utilizza la sottorete 172.17.0.0/16 per le reti dei container. Se la sottorete VPC per l'istanza del tuo ambiente utilizza lo stesso intervallo di indirizzi già utilizzato da Docker, potrebbe

verificarsi un conflitto di indirizzi IP. Per risolvere un conflitto di indirizzi IP causato da Amazon VPC e Docker utilizzando lo stesso blocco di indirizzi IPv4 CIDR, configura un blocco CIDR diverso da 172.17.0.0/16

Note

Non è possibile modificare l'intervallo di indirizzi IP di un VPC o di una sottorete esistente.

Non riesco a trovare la cartella in cui si trova il mio progetto

Problema: non riesco a trovare la cartella in cui si trova il mio progetto.

Possibili correzioni: per localizzare il progetto, cambia cartella in cui /projects si trova. Questa è la cartella in cui puoi trovare i tuoi progetti.

Non riesco a connettermi al mio ambiente di sviluppo tramite SSH

Per risolvere i problemi di connessione al tuo ambiente di sviluppo tramite SSH, puoi eseguire il ssh comando con l'-vvv opzione per mostrare ulteriori informazioni su come risolvere il problema:

```
ssh -vvv codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

Non riesco a connettermi al mio ambiente di sviluppo tramite SSH perché manca la mia configurazione SSH locale

Se manca la configurazione SSH locale (~/.ssh/config) o il contenuto della Host codecatalyst-dev-env* sezione non è aggiornato, non potrai connetterti al tuo ambiente di sviluppo tramite SSH. Per risolvere questo problema, elimina la **Host codecatalyst-dev-env*** sezione ed esegui nuovamente il primo comando dal modale SSH Access. Per ulteriori informazioni, consulta [Connessione a un ambiente di sviluppo tramite SSH](#).

Non riesco a connettermi al mio ambiente di sviluppo tramite SSH perché ho problemi con il mio profilo for the AWS Configcodecatalyst

Assicurati che il tuo AWS Config (~/.aws/config) per il codecatalyst profilo corrisponda a quello descritto in [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#). In caso contrario, elimina il profilo codecatalyst ed esegui nuovamente il primo comando dal modale SSH Access. Per ulteriori informazioni, consulta [Connessione a un ambiente di sviluppo tramite SSH](#).

Non riesco a creare un ambiente di sviluppo quando ho effettuato l'accesso CodeCatalyst utilizzando un account Single Sign-on

Problema: quando accedo alla CodeCatalyst console come utente SSO, ricevo un errore di eccezione sconosciuto quando scelgo di creare un ambiente di sviluppo nello spazio. Quando scelgo di creare un ambiente di sviluppo e scelgo l'IDE per l'accesso, ad esempio AWS Cloud9, riscontro problemi simili ai seguenti:

- La pagina Dev Environments nella CodeCatalyst console mostra l'ambiente di sviluppo nell'elenco con uno FAILED stato.
- Viene visualizzato un messaggio di errore simile al seguente:

An unknown exception happened

We encountered an unknown exception when launching your Dev Environment. Mention your Dev Environment id `error_message_ID` if you want to report or need any help.

Possibili soluzioni.

Gli ambienti di sviluppo non sono disponibili per gli utenti negli spazi in cui Active Directory viene utilizzato come provider di identità. Gli amministratori dello spazio possono utilizzare un provider di identità alternativo per accedere agli ambienti di sviluppo, come IAM Identity Center. Per ulteriori informazioni sulla pianificazione di uno spazio che supporti la federazione delle identità, consulta [Pianificazione dello spazio che supporti la federazione delle identità](#) nella Guida per l'CodeCatalyst amministratore.

Risoluzione dei problemi con IDEs

Consulta le seguenti sezioni per risolvere i problemi relativi a in. IDEs CodeCatalyst Per ulteriori informazioni su IDEs, vedere. [Creazione di un ambiente di sviluppo in un IDE](#)

Argomenti

- [Ho versioni di immagini di runtime non corrispondenti in AWS Cloud9](#)
- [Non riesco ad accedere ai miei file in /projects/projects in AWS Cloud9](#)
- [Non riesco ad avviare il mio ambiente di sviluppo AWS Cloud9 utilizzando un devfile personalizzato](#)
- [Sto riscontrando problemi in AWS Cloud9](#)

- [Nel JetBrains, non riesco a connettermi ai miei ambienti di sviluppo tramite CodeCatalyst](#)
- [Non riesco a installarlo Kit di strumenti AWS per il mio IDE](#)
- [Nel mio IDE, non riesco ad avviare i miei ambienti di sviluppo](#)

Ho versioni di immagini di runtime non corrispondenti in AWS Cloud9

AWS Cloud9 utilizza versioni diverse della risorsa frontend e dell'immagine di runtime del backend. L'utilizzo di versioni diverse potrebbe causare un funzionamento errato dell'estensione Git. Kit di strumenti AWS Per risolvere il problema, accedi alla dashboard di Dev Environment, interrompi Dev Environment e riavvialo. Per risolvere il problema di utilizzo APIs, utilizza l'UpdateDevEnvironmentAPI per aggiornare il runtime. Per ulteriori informazioni, [UpdateDevEnvironment](#) consulta il riferimento alle CodeCatalyst API di Amazon.

Non riesco ad accedere ai miei file in **/projects/projects** in AWS Cloud9

L' AWS Cloud9 editor non è in grado di accedere ai file nella directory/projects/projects. Per risolvere il problema, utilizzate il AWS Cloud9 terminale per accedere ai file o spostateli in un'altra directory.

Non riesco ad avviare il mio ambiente di sviluppo AWS Cloud9 utilizzando un devfile personalizzato

L'immagine del tuo devfile potrebbe non essere compatibile con. AWS Cloud9 Per risolvere il problema, esamina il devfile dal tuo repository e dall'ambiente di sviluppo corrispondente e creane uno nuovo per continuare.

Sto riscontrando problemi in AWS Cloud9

Per altri problemi, consulta la sezione sulla risoluzione dei problemi nella Guida per l'[AWS Cloud9 utente](#).

Nel JetBrains, non riesco a connettermi ai miei ambienti di sviluppo tramite CodeCatalyst

Per risolvere il problema, verifica di avere JetBrains installato solo la versione più recente di. Se disponi di più versioni, disinstalla le versioni precedenti e registra nuovamente il gestore di protocollo chiudendo l'IDE e il browser. Quindi apri JetBrains e registra nuovamente il gestore di protocollo.

Non riesco a installarlo Kit di strumenti AWS per il mio IDE

Per risolvere questo problema per VS Code, installa manualmente AWS Toolkit for Visual Studio Code da [GitHub](#).

Per risolvere questo problema JetBrains, installa manualmente AWS Toolkit for JetBrains da [GitHub](#).

Nel mio IDE, non riesco ad avviare i miei ambienti di sviluppo

Per risolvere questo problema per VS Code, verifica di avere l'ultima versione di VS Code e di aver AWS Toolkit for Visual Studio Code installato l'ultima versione di VS Code. Se non disponi della versione più recente, aggiorna e avvia il tuo Dev Environment. Per ulteriori informazioni, consulta [Amazon CodeCatalyst for VS Code](#).

Per risolvere questo problema JetBrains, verifica di avere AWS Toolkit for JetBrains installato la versione più recente JetBrains di. Se non disponi della versione più recente, aggiorna e avvia il tuo Dev Environment. Per ulteriori informazioni, consulta [Amazon CodeCatalyst for JetBrains](#).

Risoluzione dei problemi con devfiles

Consulta le seguenti sezioni per risolvere i problemi relativi a devfiles in. CodeCatalyst Per ulteriori informazioni su devfiles, vedere. [Configurazione di un devfile per un ambiente di sviluppo](#)

Argomenti

- [Il mio ambiente di sviluppo utilizza il devfile universale predefinito anche se ho implementato un'immagine personalizzata in un devfile personalizzato](#)
- [Il mio progetto non è stato creato nel mio ambiente di sviluppo con il devfile universale predefinito](#)
- [Voglio spostare un repository devfile per un ambiente di sviluppo](#)
- [Ho problemi ad avviare il mio devfile](#)
- [Non so come controllare lo stato del mio devfile](#)
- [Il mio devfile non è compatibile con gli strumenti forniti nell'ultima immagine](#)

Il mio ambiente di sviluppo utilizza il devfile universale predefinito anche se ho implementato un'immagine personalizzata in un devfile personalizzato

Se si CodeCatalyst verificano errori durante l'avvio di un ambiente di sviluppo che utilizza un devfile personalizzato, l'ambiente di sviluppo utilizza per impostazione predefinita il devfile universale predefinito. Per risolvere il problema, puoi controllare l'errore esatto nei log sottostanti. /aws/mde/

logs/devfile.log Puoi anche verificare se postStart l'esecuzione è andata a buon fine nei tuoi log: `/aws/mde/logs/devfileCommand.log`

Il mio progetto non è stato creato nel mio ambiente di sviluppo con il devfile universale predefinito

Per risolvere il problema, verifica di non utilizzare un devfile personalizzato. Se non state utilizzando un devfile personalizzato, visualizzate il `devfile.yaml` file nell'archivio di origine del progetto per individuare e correggere eventuali errori.

Voglio spostare un repository devfile per un ambiente di sviluppo

Puoi spostare il devfile predefinito nel `/projects/devfile.yaml` tuo repository del codice sorgente. Per aggiornare la posizione del devfile, usa il seguente comando: `/aws/mde/mde start --location repository-name/devfile.yaml`

Ho problemi ad avviare il mio devfile

Se c'è un problema durante l'avvio del tuo devfile, entrerà in modalità di ripristino in modo che tu possa comunque connetterti al tuo ambiente e correggere il tuo devfile. In modalità di ripristino, l'esecuzione `/aws/mde/mde status` non conterrà la posizione del tuo devfile.

```
{
  "status": "STABLE"
}
```

Puoi controllare l'errore nei log sotto `/aws/mde/logs`, correggere il devfile e riprovare a eseguire. `/aws/mde/mde start`

Non so come controllare lo stato del mio devfile

Puoi controllare lo stato del tuo devfile eseguendo `/aws/mde/mde status` Dopo aver eseguito questo comando, è possibile che venga visualizzato uno dei seguenti:

- `{"status": "STABLE", "location": "devfile.yaml" }`

Ciò indica che il file dev è corretto.

- `{"status": "STABLE" }`

Ciò indica che il tuo devfile non può essere avviato ed è entrato in modalità di ripristino.

Puoi controllare l'errore esatto nei log sotto. `/aws/mde/logs/devfile.log`

Puoi anche verificare se `postStart` l'esecuzione è andata a buon fine nei tuoi log: `/aws/mde/logs/devfileCommand.log`

Per ulteriori informazioni, consulta [Specificare immagini devfile universali per un ambiente di sviluppo](#).

Il mio devfile non è compatibile con gli strumenti forniti nell'ultima immagine

Nel tuo ambiente di sviluppo, `devfile` o `devfile postStart` potrebbe fallire se `latest` gli strumenti non dispongono degli strumenti necessari per un progetto specifico. Per risolvere il problema, procedi come segue:

1. Vai al tuo devfile.
2. Nel tuo devfile, esegui l'aggiornamento a una versione con immagine granulare anziché. `latest` Potrebbe essere simile al seguente:

```
components:
  - container:
      image: public.ecr.aws/amazonlinux/universal-image:1.0
```

3. Crea un nuovo ambiente di sviluppo utilizzando il devfile aggiornato.

Risoluzione dei problemi relativi ai flussi di lavoro

Consulta le seguenti sezioni per risolvere i problemi relativi ai flussi di lavoro in Amazon.

CodeCatalyst Per ulteriori informazioni sui flussi di lavoro, consulta [Crea, testa e distribuisce con i flussi di lavoro](#).

Argomenti

- [Come posso correggere i messaggi «Workflow is inactive»?](#)
- [Come posso correggere gli errori «La definizione del flusso di lavoro presenta n errori»?](#)
- [Come posso correggere gli errori «Impossibile individuare le credenziali» e "ExpiredToken"?](#)
- [Come posso correggere gli errori «Impossibile connettersi al server»?](#)
- [Perché mancano CodeDeploy dei campi nell'editor visivo?](#)
- [Come posso correggere gli errori relativi alle funzionalità IAM?](#)
- [Come posso correggere gli errori di «npm install»?](#)

- [Perché più flussi di lavoro hanno lo stesso nome?](#)
- [Posso archiviare i miei file di definizione del flusso di lavoro in un'altra cartella?](#)
- [Come faccio ad aggiungere azioni in sequenza al mio flusso di lavoro?](#)
- [Perché il mio flusso di lavoro viene convalidato correttamente ma fallisce in fase di esecuzione?](#)
- [L'individuazione automatica non rileva alcun rapporto relativo alla mia azione](#)
- [La mia azione non riesce sui report rilevati automaticamente dopo aver configurato i criteri di successo](#)
- [L'individuazione automatica genera report che non voglio](#)
- [L'individuazione automatica genera molti report di piccole dimensioni per un singolo framework di test](#)
- [I flussi di lavoro elencati in CI/CD non corrispondono a quelli del repository di origine](#)
- [Non riesco a creare o aggiornare i flussi di lavoro](#)

Come posso correggere i messaggi «Workflow is inactive»?

Problema: nella CodeCatalyst console, in CI/CD, Workflows, il flusso di lavoro viene visualizzato con il seguente messaggio:

`Workflow is inactive.`

Questo messaggio indica che il file di definizione del flusso di lavoro contiene un trigger che non si applica al ramo in cui ti trovi attualmente. Ad esempio, il file di definizione del flusso di lavoro potrebbe contenere un PUSH trigger che fa riferimento al tuo main ramo, ma ti trovi su un feature branch. Poiché le modifiche che stai apportando al feature branch non si applicano a main, e non avvieranno, il flusso di lavoro viene eseguito in main, CodeCatalyst disattiva il flusso di lavoro sul branch e lo contrassegna come `Inactive`.

Possibili soluzioni.

Se desideri avviare un flusso di lavoro sul tuo feature branch, puoi fare quanto segue:

- Nel tuo feature branch, nel file di definizione del flusso di lavoro, rimuovi la Branches proprietà dalla Triggers sezione in modo che abbia il seguente aspetto:

```
Triggers:
- Type: PUSH
```

Questa configurazione fa sì che il trigger si attivi quando viene inviato un push a qualsiasi branch, incluso il feature branch. Se il trigger è attivato, CodeCatalyst avvierà l'esecuzione di un flusso di lavoro utilizzando il file di definizione del flusso di lavoro e i file sorgente presenti nel ramo verso cui si sta eseguendo il push.

- Nel tuo feature branch, nel file di definizione del flusso di lavoro, rimuovi la `Triggers` sezione ed esegui il flusso di lavoro manualmente.
- Nel tuo feature branch, nel file di definizione del flusso di lavoro, modifica la `PUSH` sezione in modo che faccia riferimento al tuo feature branch anziché a un altro ramo (come `main`, ad esempio).

Important

Fai attenzione a non confermare queste modifiche se non intendi unirle per tornare alla tua `main` filiale.

Per ulteriori informazioni sulla modifica del file di definizione del flusso di lavoro, consulta [Creazione di un flusso di lavoro](#).

Per ulteriori informazioni sui trigger, consulta [L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#).

Come posso correggere gli errori «La definizione del flusso di lavoro presenta *n* errori»?

Problema: viene visualizzato uno dei seguenti messaggi di errore:

Errore 1:

Nella pagina CI/CD, Flussi di lavoro, sotto il nome del flusso di lavoro, vedi:

```
Workflow definition has n errors
```

Errore 2:

Durante la modifica di un flusso di lavoro, scegli il pulsante `Convalida` e nella parte superiore della CodeCatalyst console viene visualizzato il seguente messaggio:

```
The workflow definition has errors. Fix the errors and choose Validate to verify your changes.
```

Errore 3:

Dopo aver visitato la pagina dei dettagli del flusso di lavoro, viene visualizzato il seguente errore nel campo Definizione del flusso di lavoro:

n errors

Possibili soluzioni.

- Scegli CI/CD, scegli Flussi di lavoro e scegli il nome del flusso di lavoro che presenta l'errore. Nel campo Definizione del flusso di lavoro in alto, scegli il link all'errore. I dettagli sull'errore vengono visualizzati nella parte inferiore della pagina. Segui i suggerimenti per la risoluzione dei problemi riportati nell'errore per risolvere il problema.
- Assicurati che il file di definizione del flusso di lavoro sia un file YAML.
- Assicurati che le proprietà YAML nel file di definizione del flusso di lavoro siano annidate al livello corretto. Per vedere come devono essere annidate le proprietà nel file di definizione del flusso di lavoro, consultate o consultate la [Definizione YAML del flusso di lavoro](#) documentazione dell'azione, che è collegata a from. [Aggiungere un'azione a un flusso di lavoro](#)
- Accertatevi che gli asterischi (*) e gli altri caratteri speciali non compaiano correttamente. Per evitarli, aggiungete virgolette singole o doppie. Per esempio:

```
Outputs:  
  Artifacts:  
    - Name: myartifact  
      Files:  
        - "**/*"
```

Per ulteriori informazioni sui caratteri speciali nel file di definizione del flusso di lavoro, consulta [Linee guida e convenzioni sulla sintassi](#).

- Assicurati che le proprietà YAML nel file di definizione del flusso di lavoro utilizzino le maiuscole corrette. Per ulteriori informazioni sulle regole di classificazione delle maiuscole, consulta. [Linee guida e convenzioni sulla sintassi](#) Per determinare la classificazione delle maiuscole e minuscole corretta per ogni proprietà [Definizione YAML del flusso di lavoro](#), consulta o consulta la documentazione relativa all'azione, che è collegata a from. [Aggiungere un'azione a un flusso di lavoro](#)
- Assicurati che la SchemaVersion proprietà sia presente e impostata sulla versione corretta nel file di definizione del flusso di lavoro. Per ulteriori informazioni, consulta [SchemaVersion](#).

- Assicuratevi che la `Triggers` sezione del file di definizione del flusso di lavoro includa tutte le proprietà richieste. Per determinare le proprietà richieste, scegli il trigger nell'[editor visivo](#) e cerca i campi in cui mancano informazioni, oppure consulta la documentazione di riferimento del trigger all'indirizzo [Triggers](#).
- Assicuratevi che la `DependsOn` proprietà nel file di definizione del flusso di lavoro sia configurata correttamente e non introduca dipendenze circolari. Per ulteriori informazioni, consulta [Azioni di sequenziamento](#).
- Assicuratevi che la `Actions` sezione del file di definizione del flusso di lavoro includa almeno un'azione. Per ulteriori informazioni, consulta [Azioni](#).
- Assicuratevi che ogni azione includa tutte le proprietà richieste. Per determinare le proprietà richieste, scegli l'azione nell'[editor visivo](#) e cerca i campi in cui mancano informazioni, oppure consulta la documentazione dell'azione, a cui è collegato il link da [Aggiungere un'azione a un flusso di lavoro](#).
- Assicuratevi che tutti gli artefatti di input abbiano artefatti di output corrispondenti. Per ulteriori informazioni, consulta [Definizione di un artefatto di output](#).
- Assicuratevi che le variabili definite in un'azione vengano esportate in modo che possano essere utilizzate in altre azioni. Per ulteriori informazioni, consulta [Esportazione di una variabile in modo che altre azioni possano utilizzarla](#).

Come posso correggere gli errori «Impossibile individuare le credenziali» e "ExpiredToken"?

Problema: durante l'elaborazione [Tutorial: distribuzione di un'applicazione su Amazon EKS](#), nella finestra del terminale della macchina di sviluppo vengono visualizzati uno o entrambi i seguenti messaggi di errore:

```
Unable to locate credentials. You can configure credentials by running "aws configure".
```

```
ExpiredToken: The security token included in the request is expired
```

Possibili soluzioni.

Questi errori indicano che le credenziali che stai utilizzando per accedere ai AWS servizi sono scadute. In questo caso, non eseguire il `aws configure` comando. Utilizzate invece le seguenti istruzioni per aggiornare la chiave di AWS accesso e il token di sessione.

Per aggiornare la chiave di AWS accesso e il token di sessione

1. Assicurati di avere l'URL del portale di AWS accesso, il nome utente e la password dell'utente che stai utilizzando per completare il tutorial Amazon EKS (codecatalyst-eks-user). Dovresti aver configurato questi elementi al termine [Passaggio 1: configura la tua macchina di sviluppo](#) del tutorial.

 Note

Se non disponi di queste informazioni, vai alla pagina dei codecatalyst-eks-user dettagli in IAM Identity Center, scegli Reimposta password, Genera una password monouso [...] e Reimposta nuovamente la password per visualizzare le informazioni sullo schermo.

2. Esegui una di queste operazioni:
 - Incolla l'URL del portale di AWS accesso nella barra degli indirizzi del browser.

Or

 - Aggiorna la pagina del portale di AWS accesso se è già stata caricata.
3. Accedi con il nome codecatalyst-eks-user utente e la password, se non hai già effettuato l'accesso.
4. Scegli Account AWS, quindi scegli il nome Account AWS a cui hai assegnato l'codecatalyst-eks-userutente e il set di autorizzazioni.
5. Accanto al nome del set di autorizzazioni (codecatalyst-eks-permission-set), scegliete Accesso da riga di comando o accesso programmatico.
6. Copia i comandi al centro della pagina. Sono simili ai seguenti:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
export AWS_SESSION_TOKEN="session-token"
```

... *session-token* dov'è una lunga stringa casuale.

7. Incolla i comandi nel prompt del terminale sulla tua macchina di sviluppo e premi Invio.

Le nuove chiavi e il token di sessione vengono caricati.

Ora hai aggiornato le tue credenziali. I `kubectl` comandi `AWS CLI` `eksctl`, e ora dovrebbero funzionare.

Come posso correggere gli errori «Impossibile connettersi al server»?

Problema: durante l'esecuzione del tutorial descritto in [Tutorial: distribuzione di un'applicazione su Amazon EKS](#), viene visualizzato un messaggio di errore simile al seguente nella finestra del terminale della macchina di sviluppo:

```
Unable to connect to the server: dial tcp: lookup long-string.gr7.us-west-2.eks.amazonaws.com on 1.2.3.4:5: no such host
```

Possibili soluzioni.

Questo errore indica in genere che le credenziali utilizzate dall'`kubectl` utilità per connettersi al cluster Amazon EKS sono scadute. Per risolvere il problema, aggiorna le credenziali inserendo il seguente comando al prompt del terminale:

```
aws eks update-kubeconfig --name codecatalyst-eks-cluster --region us-west-2
```

Dove:

- *codecatalyst-eks-cluster* viene sostituito dal nome del tuo cluster Amazon EKS.
- *us-west-2* viene sostituito dalla AWS regione in cui è distribuito il cluster.

Perché mancano CodeDeploy dei campi nell'editor visivo?

Problema: stai utilizzando un'azione [Deploy to Amazon ECS](#) e non vedi CodeDeploy i campi come CodeDeploy AppSpec nell'editor visivo del flusso di lavoro. Questo problema può verificarsi perché il servizio Amazon ECS specificato nel campo Servizio non è configurato per eseguire distribuzioni blu/verdi.

Possibili soluzioni.

- Scegli un servizio Amazon ECS diverso nella scheda Configurazione dell'azione Deploy to Amazon ECS. Per ulteriori informazioni, consulta [Distribuzione su Amazon ECS con un flusso di lavoro](#).

- Configura il servizio Amazon ECS selezionato per eseguire le distribuzioni, consulta la sezione [Blue/Green blue/green deployments. For more information about configuring blue/green deployment with nella](#) Amazon Elastic Container Service CodeDeploy Developer Guide.

Come posso correggere gli errori relativi alle funzionalità IAM?

Problema: stai utilizzando un'azione [Deploy CloudFormation stack](#) e lo vedi **##[error] requires capabilities: [*capability-name*]** nei log dell'azione Deploy stack. CloudFormation

Possibili correzioni: completa la seguente procedura per aggiungere la funzionalità al file di definizione del flusso di lavoro. Per ulteriori informazioni sulle funzionalità IAM, consulta la sezione [Riconoscimento delle risorse IAM nei CloudFormation modelli nella Guida per l'utente IAM](#).

Visual

Per aggiungere una funzionalità IAM utilizzando l'editor visivo

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli Visual.
7. Nel diagramma del flusso di lavoro, scegli l'azione Deploy CloudFormation stack.
8. Scegli la scheda Configurazione.
9. In basso, scegli Avanzato - opzionale.
10. Nell'elenco a discesa Funzionalità, seleziona la casella di controllo accanto alla funzionalità menzionata nel messaggio di errore. Se la funzionalità non è disponibile nell'elenco, usa l'editor YAML per aggiungerla.
11. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
12. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.

13. Se l'esecuzione di un nuovo flusso di lavoro non si avvia automaticamente, esegui il flusso di lavoro manualmente per vedere se le modifiche correggono l'errore. Per ulteriori informazioni sull'esecuzione manuale di un flusso di lavoro, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

YAML

Per aggiungere una funzionalità IAM utilizzando l'editor YAML

1. [Apri la CodeCatalyst console all'indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Scegliere il progetto.
3. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
4. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
5. Scegli Modifica.
6. Scegli YAML.
7. Nell'azione Deploy CloudFormation stack, aggiungi una capabilities proprietà, come questa:

```
DeployCloudFormationStack:
  Configuration:
    capabilities: capability-name
```

Sostituisci *capability-name* con il nome della funzionalità IAM mostrata nel messaggio di errore. Usa virgole e nessun spazio per elencare più funzionalità. Per ulteriori informazioni, vedere la descrizione della capabilities proprietà nel [Azione 'Implementa CloudFormation stack' YAML](#).

8. (Facoltativo) Scegliete Convalida per convalidare il codice YAML del flusso di lavoro prima di eseguire il commit.
9. Scegliete Commit, inserite un messaggio di commit e scegliete nuovamente Commit.
10. Se l'esecuzione di un nuovo flusso di lavoro non si avvia automaticamente, esegui il flusso di lavoro manualmente per vedere se le modifiche correggono l'errore. Per ulteriori informazioni sull'esecuzione manuale di un flusso di lavoro, consulta [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Come posso correggere gli errori di «npm install»?

Problema: l'azione [AWS CDK deploy](#) o l'azione [AWS CDK bootstrap](#) falliscono con un errore. `npm install` Questo errore può verificarsi perché stai archiviando le dipendenze AWS CDK dell'app in un registro di gestione dei pacchetti del nodo privato (npm) a cui l'azione non può accedere.

Possibili correzioni: utilizza le seguenti istruzioni per aggiornare il `cdk.json` file dell' AWS CDK app con informazioni aggiuntive sul registro e sull'autenticazione.

Prima di iniziare

1. Crea segreti per le tue informazioni di autenticazione. Farai riferimento a questi segreti nel `cdk.json` file invece di fornire gli equivalenti in chiaro. Per creare i segreti:
 - a. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
 - b. Scegliere il progetto.
 - c. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Segreti.
 - d. Crea due segreti con le seguenti proprietà:

Primo segreto	Secondo segreto
Nome: <code>npmUsername</code>	Nome: <code>npmAuthToken</code>
Valore: <code>npm-username</code> , <code>npm-username</code> dov'è il nome utente utilizzato per l'autenticazione nel registro npm privato.	Valore: <code>npm-auth-token</code> , <code>npm-auth-token</code> dov'è il token di accesso utilizzato o per l'autenticazione nel registro npm privato. Per ulteriori informazioni sui token di accesso npm, vedere Informazioni sui token di accesso nella documentazione di npm .
(Facoltativo) Descrizione: The username used to authenticate to the private npm registry.	(Facoltativo) Descrizione: The access token used to authenticate to the private npm registry.

Per ulteriori informazioni sui segreti, vedere [Mascherare i dati utilizzando segreti](#).

2. Aggiungi i segreti come variabili di ambiente alla tua AWS CDK azione. L'azione sostituirà le variabili con valori reali durante l'esecuzione. Per aggiungere i segreti:
 - a. Nel riquadro di navigazione, scegli CI/CD, quindi scegli Flussi di lavoro.
 - b. Scegli il nome del tuo flusso di lavoro. Puoi filtrare in base al nome del repository o del ramo di origine in cui è definito il flusso di lavoro oppure filtrare in base al nome o allo stato del flusso di lavoro.
 - c. Scegli Modifica.
 - d. Scegli Visual.
 - e. Nel diagramma del flusso di lavoro, scegli la tua AWS CDK azione.
 - f. Scegli la scheda Input.
 - g. Aggiungi due variabili con le seguenti proprietà:

Prima variabile	Seconda variabile
Nome: NPMUSER	Nome: NPMTOKEN
Value (Valore): \${Secrets.npmUsername}	Value (Valore): \${Secrets.npmAuthToken}

Ora hai due variabili contenenti riferimenti a segreti.

Il codice YAML del file di definizione del flusso di lavoro dovrebbe essere simile al seguente:

Note

Il seguente esempio di codice proviene da un'azione di AWS CDK bootstrap; un'azione di AWS CDK distribuzione avrà un aspetto simile.

```
Name: CDK_Bootstrap_Action
SchemaVersion: 1.0
Actions:
  CDKBootstrapAction:
    Identifier: aws/cdk-bootstrap@v2
    Inputs:
```

```

Variables:
  - Name: NPMUSER
    Value: ${Secrets.npmUsername}
  - Name: NPMTOKEN
    Value: ${Secrets.npmAuthToken}
Sources:
  - WorkflowSource
Environment:
  Name: Dev2
Connections:
  - Name: account-connection
    Role: codecatalystAdmin
Configuration:
  Parameters:
    Region: "us-east-2"

```

Ora sei pronto per utilizzare le NPMTOKEN variabili NPMUSER and nel tuo `cdk.json` file. Passate alla procedura successiva.

Per aggiornare il file `cdk.json`

1. Passa alla directory principale del AWS CDK progetto e apri il file `cdk.json`
2. Trova la `"app"`: proprietà e modificala per includere il codice mostrato in *red italics*:

Note

Il codice di esempio seguente proviene da un TypeScript progetto. Se stai usando un JavaScript progetto, il codice avrà un aspetto simile ma non identico.

```

{
  "app": "npm set registry=https://your-registry/folder/CDK-package/ --
userconfig .npmrc && npm set //your-registry/folder/CDK-package/:always-auth=true
--userconfig .npmrc && npm set //your-registry/folder/CDK-package/:_authToken=
\"${NPMUSER}\"\":\"${NPMTOKEN}\" && npm install && npx ts-node --prefer-ts-exts bin/
hello-cdk.ts/js",
  "watch": {
    "include": [
      "***"
    ],
  },

```



```
"exclude": [  
  "README.md",  
  "cdk*.json",  
  "**/*.d.ts",  
  "**/*.js",  
  "tsconfig.json",  
  "package*.json",  
  ...  
]
```

3. Nel codice evidenziato in *red italics*, sostituisci:

- *your-registry/folder/CDK-package/* con il percorso delle dipendenze AWS CDK del progetto nel registro privato.
- *hello-cdk.ts|.js* con il nome del file entrypoint. Questo può essere un file `.ts` (TypeScript) o `.js` (JavaScript) a seconda della lingua che stai usando.

 Note

L'azione sostituirà *NPM_TOKEN* le variabili *NPM_USER* and con il nome utente e il token di accesso npm che hai specificato in Secrets.

4. Salva il file `cdk.json`.
5. Esegui nuovamente l'azione manualmente per vedere se le modifiche correggono l'errore. Per ulteriori informazioni sull'esecuzione manuale delle azioni, vedere [Avvio manuale dell'esecuzione di un flusso di lavoro](#).

Perché più flussi di lavoro hanno lo stesso nome?

I flussi di lavoro vengono archiviati per ramo per repository. Due flussi di lavoro diversi possono avere lo stesso nome se esistono in rami diversi. Nella pagina [Flussi di lavoro](#), puoi differenziare i flussi di lavoro con lo stesso nome osservando il nome del ramo. Per ulteriori informazioni, consulta [Organizzazione del codice sorgente, lavoro con le filiali di Amazon CodeCatalyst](#).

Workflows (6) [Create workflow](#)

Filter workflows

BuildToProd [Actions](#)

cfn-source-repository **branch-A**

No recent runs.

[View all runs](#)

BuildToProd [Actions](#)

cfn-source-repository **main**

► **Recent runs** ✓ ✗ ✓ ✓

[View all runs](#)

Posso archiviare i miei file di definizione del flusso di lavoro in un'altra cartella?

No, è necessario archiviare tutti i file di definizione del flusso di lavoro nella `.codecatalyst/workflows` cartella o nelle sottocartelle di tale cartella. Se utilizzi un repository mono con più progetti logici, posiziona tutti i file di definizione del flusso di lavoro `.codecatalyst/workflows` nella cartella o in una delle relative sottocartelle, quindi utilizza il campo `File modificati` (editor visivo) o la `FilesChanged` proprietà (editor YAML) all'interno di un trigger per attivare automaticamente il flusso di lavoro in un percorso di progetto specificato. Per ulteriori informazioni, consulta [Aggiungere trigger ai flussi di lavoro](#) e [Esempio: un trigger con push, branch e file](#).

Come faccio ad aggiungere azioni in sequenza al mio flusso di lavoro?

Per impostazione predefinita, quando aggiungi un'azione al tuo flusso di lavoro, questa non avrà dipendenze e verrà eseguita in parallelo con altre azioni.

Se desideri disporre le azioni in sequenza, puoi impostare una dipendenza da un'altra azione impostando il `DependsOn` campo. Puoi anche configurare un'azione per consumare artefatti o variabili che sono risultati di altre azioni. Per ulteriori informazioni, consulta [Azioni di sequenziamento](#).

Perché il mio flusso di lavoro viene convalidato correttamente ma fallisce in fase di esecuzione?

Se hai convalidato il flusso di lavoro utilizzando il `Validate` pulsante, ma il flusso di lavoro non è riuscito comunque, potrebbe trattarsi di una limitazione nel validatore.

Eventuali errori relativi a una CodeCatalyst risorsa come segreti, ambienti o flotte nella configurazione del flusso di lavoro non verranno registrati durante un commit. Se vengono utilizzati riferimenti non validi, l'errore verrà identificato solo quando viene eseguito un flusso di lavoro. Analogamente, se sono presenti errori nella configurazione delle azioni, ad esempio la mancanza di un campo obbligatorio o errori di battitura negli attributi delle azioni, tali errori verranno identificati solo durante l'esecuzione del flusso di lavoro. Per ulteriori informazioni, consulta [Creazione di un flusso di lavoro](#).

L'individuazione automatica non rileva alcun rapporto relativo alla mia azione

Problema: ho configurato l'individuazione automatica per un'azione che esegue test, ma non viene rilevato alcun report. CodeCatalyst

Possibili correzioni: il problema potrebbe essere causato da diversi problemi. Prova una o più delle seguenti soluzioni:

- Assicurati che lo strumento utilizzato per eseguire i test produca output in uno dei formati compatibili CodeCatalyst . Ad esempio, se desideri consentire CodeCatalyst di scoprire pytest i report sui test e sulla copertura del codice, includi i seguenti argomenti:

```
--junitxml=test_results.xml --cov-report xml:test_coverage.xml
```

Per ulteriori informazioni, consulta [Tipi di report sulla qualità](#).

- Assicurati che l'estensione del file per gli output sia coerente con il formato scelto. Ad esempio, quando configuri pytest per produrre risultati in JUnitXML formato, verifica che l'estensione del file sia. `.xml` Per ulteriori informazioni, consulta [Tipi di report sulla qualità](#).
- Assicuratevi che IncludePaths sia configurato per includere l'intero file system (`**/*`) a meno che non stiate escludendo alcune cartelle di proposito. Allo stesso modo, assicurati di ExcludePaths non escludere le directory in cui prevedi che si trovino i report.
- Se hai configurato manualmente un report per utilizzare un file di output specifico, verrà escluso dal rilevamento automatico. Per ulteriori informazioni, consulta [Rapporti sulla qualità \(esempio YAML\)](#).
- L'individuazione automatica potrebbe non trovare i report perché l'azione non è riuscita prima della generazione degli output. Ad esempio, la build potrebbe aver avuto esito negativo prima dell'esecuzione di qualsiasi test unitario.

La mia azione non riesce sui report rilevati automaticamente dopo aver configurato i criteri di successo

Problema: quando abilito l'individuazione automatica e configuro i criteri di successo, alcuni report non soddisfano i criteri di successo e di conseguenza l'azione fallisce.

Possibili correzioni: per risolvere il problema, prova una o più delle seguenti soluzioni:

- Modifica `IncludePaths` o `ExcludePaths` escludi i report che non ti interessano.
- Aggiorna i criteri di successo per consentire l'esecuzione di tutti i report. Ad esempio, se vengono rilevati due report con una copertura di linea del 50% e un altro del 70%, regola la copertura minima della linea al 50%. Per ulteriori informazioni, consulta [Criteri di successo](#)
- Trasforma il rapporto di errore in un rapporto configurato manualmente. Ciò consente di configurare diversi criteri di successo per quel rapporto specifico. Per ulteriori informazioni, consulta [Configurazione dei criteri di successo per i report](#).

L'individuazione automatica genera report che non voglio

Problema: quando abilito l'individuazione automatica, genera report che non voglio. Ad esempio, CodeCatalyst genera report sulla copertura del codice per i file inclusi nelle dipendenze della mia applicazione archiviate in `node_modules`

Possibili correzioni: è possibile regolare la `ExcludePaths` configurazione per escludere i file indesiderati. Ad esempio, per escludere `node_modules`, aggiungi `node_modules/**/*.*`. Per ulteriori informazioni, consulta [Includi/escludi percorsi](#).

L'individuazione automatica genera molti report di piccole dimensioni per un singolo framework di test

Problema: quando utilizzo determinati framework di reporting di test e copertura del codice, ho notato che l'individuazione automatica genera un gran numero di report. Ad esempio, quando si utilizza il [plug-in Maven Surefire](#), l'individuazione automatica produce un report diverso per ogni classe di test.

Possibili correzioni: il framework potrebbe essere in grado di aggregare gli output in un unico file. Ad esempio, se utilizzi il plug-in Maven Surefire, puoi utilizzarlo per aggregare i file manualmente. `npj junit-merge` L'espressione completa può essere simile a questa:

```
mvn test; cd test-package-path/surefire-reports && npx junit-merge -d ./ && rm  
*Test.xml
```

I flussi di lavoro elencati in CI/CD non corrispondono a quelli del repository di origine

Problema: [i flussi di lavoro visualizzati nella pagina CI/CD, Flussi di lavoro non corrispondono a quelli nella cartella del repository di origine. ~/.codecatalyst/workflows/](#) Potresti riscontrare le seguenti discrepanze:

- Nella pagina Flussi di lavoro viene visualizzato un flusso di lavoro, ma nel repository di origine non esiste un file di definizione del flusso di lavoro corrispondente.
- Nel repository di origine esiste un file di definizione del flusso di lavoro, ma un flusso di lavoro corrispondente non viene visualizzato nella pagina Flussi di lavoro.
- Esiste un flusso di lavoro sia nel repository di origine che nella pagina Flussi di lavoro, ma i due sono diversi.

Questo problema può verificarsi se la pagina Flussi di lavoro non ha avuto il tempo di aggiornarsi o se è stata superata la quota del flusso di lavoro.

Possibili soluzioni.

- Attesa. In genere è necessario attendere due o tre secondi dopo il commit al codice sorgente prima di visualizzare la modifica nella pagina Flussi di lavoro.
- Se hai superato la quota del flusso di lavoro, esegui una delle seguenti operazioni:

Note

Per determinare se una quota di flusso di lavoro è stata superata [Quote per i flussi di lavoro in CodeCatalyst](#), rivedi e confronta le quote documentate con i flussi di lavoro nel tuo repository di origine o nella pagina Flussi di lavoro. Non viene visualizzato alcun messaggio di errore che indichi che una quota è stata superata, quindi dovrai indagare da solo.

- Se hai superato il numero massimo di flussi di lavoro per quota di spazio, elimina alcuni flussi di lavoro e quindi esegui un commit di test sul file di definizione del flusso di lavoro. Un esempio di commit di test potrebbe essere l'aggiunta di uno spazio al file.

- Se hai superato la quota di dimensione massima del file di definizione del flusso di lavoro, modifica il file di definizione del flusso di lavoro per ridurne la lunghezza.
- Se hai superato la quota massima di file di workflow elaborati in un'unica quota di eventi di origine, esegui diversi commit di test. Modifica un numero di flussi di lavoro inferiore al massimo in ogni commit.

Non riesco a creare o aggiornare i flussi di lavoro

Problema: voglio creare o aggiornare un flusso di lavoro, ma visualizzo un errore quando tento di eseguire la modifica.

Possibili correzioni: a seconda del ruolo che ricopri nel progetto o nello spazio, potresti non avere le autorizzazioni per inviare codice agli archivi di origine del progetto. I file YAML per i flussi di lavoro sono archiviati nei repository. Per ulteriori informazioni, consulta [File di definizione del flusso di lavoro](#). Il ruolo di amministratore dello spazio, il ruolo di amministratore del progetto e il ruolo di collaboratore sono tutti autorizzati a eseguire il commit e il trasferimento del codice nei repository di un progetto.

Se hai il ruolo di Collaboratore ma non puoi creare o confermare modifiche al flusso di lavoro YAML in un ramo specifico, è possibile che per quel ramo sia configurata una regola di branch che impedisce agli utenti con quel ruolo di inviare codice a quel particolare ramo. Prova a creare un flusso di lavoro in un ramo diverso o a trasferire le modifiche in un ramo diverso. Per ulteriori informazioni, consulta [Gestisci le azioni consentite per una filiale con le regole della filiale](#).

Risoluzione dei problemi relativi ai problemi

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni relativi a CodeCatalyst

Argomenti

- [Non riesco a scegliere un assegnatario per il mio problema](#)

Non riesco a scegliere un assegnatario per il mio problema

Problema: quando si crea un problema, l'elenco degli assegnatari è vuoto.

Possibili correzioni: l'elenco degli assegnatari è collegato direttamente CodeCatalyst agli utenti elencati come membri del progetto. Per verificare che l'accesso al profilo utente funzioni

correttamente, scegli l'icona del profilo, quindi scegli Profilo utente. Se le informazioni del profilo utente non vengono inserite, consulta il rapporto sullo stato di salute per eventuali incidenti. Se viene compilato, invia un ticket di assistenza.

Risoluzione dei problemi relativi alla ricerca in CodeCatalyst

Consulta le seguenti sezioni per risolvere i problemi relativi alla ricerca in CodeCatalyst. Per ulteriori informazioni sui flussi di lavoro, consulta [Cerca codice, problemi, progetti e utenti in CodeCatalyst](#).

Argomenti

- [Non riesco a trovare un utente nel mio progetto](#)
- [Non vedo cosa sto cercando nel mio progetto o nel mio spazio](#)
- [Il numero di risultati di ricerca continua a cambiare quando navigo tra le pagine](#)
- [La mia query di ricerca non è stata completata](#)

Non riesco a trovare un utente nel mio progetto

Problema: quando cerco di visualizzare i dettagli di un utente, non vedo le sue informazioni nel progetto.

Possibili correzioni: attualmente la ricerca non supporta la ricerca di utenti all'interno di un progetto. Per cercare utenti con accesso al tuo spazio, passa a Questo spazio in QuickSearch o rimuovi tutti i filtri di progetto che potresti aver specificato utilizzando il linguaggio di interrogazione avanzato.

Non vedo cosa sto cercando nel mio progetto o nel mio spazio

Problema: i risultati non vengono visualizzati quando cerco di cercare informazioni particolari.

Possibili correzioni: è probabile che gli aggiornamenti dei contenuti richiedano alcuni secondi per essere aggiornati nei risultati di ricerca. Gli aggiornamenti di grandi dimensioni possono richiedere diversi minuti.

Per le risorse che non sono state aggiornate di recente, potrebbe essere necessario affinare la ricerca. Puoi perfezionare la ricerca aggiungendo altre parole chiave o utilizzando il linguaggio di interrogazione avanzato. Per ulteriori informazioni sul perfezionamento delle query, consulta [Perfezionamento della query di ricerca](#)

Il numero di risultati di ricerca continua a cambiare quando navigo tra le pagine

Problema: il numero di risultati di ricerca sembra cambiare quando passo alla pagina successiva, quindi non è chiaro quanti risultati totali ci siano.

Possibili correzioni: quando navighi tra le pagine dei risultati di ricerca, potresti notare una modifica nel numero di risultati di ricerca che corrispondono alla tua query. Il numero di risultati potrebbe essere aggiornato per riflettere un numero più accurato di corrispondenze scoperte mentre navighi tra le pagine.

Mentre navighi tra i risultati, potresti visualizzare il seguente messaggio: Nessun risultato per «test». Riceverai il messaggio se non avrai accesso ai risultati rimanenti.

La mia query di ricerca non è stata completata

Problema: i risultati della mia query di ricerca non vengono visualizzati e sembra che stia impiegando troppo tempo.

Possibili correzioni: la ricerca potrebbe non essere completata quando nello spazio vengono eseguite molte ricerche contemporaneamente, a livello di programmazione o a causa dell'elevata attività del team. Se esegui ricerche programmatiche, mettile in pausa o riducile. Altrimenti, riprova tra qualche secondo.

Risoluzione dei problemi relativi alle estensioni

Consulta le seguenti sezioni per risolvere i problemi relativi alle estensioni in. CodeCatalyst Per ulteriori informazioni sulle estensioni, consulta. [Aggiungi funzionalità ai progetti con estensioni in CodeCatalyst](#)

Argomenti

- [Non riesco a visualizzare le modifiche a un archivio di terze parti collegato o a cercare i risultati di tali modifiche](#)

Non riesco a visualizzare le modifiche a un archivio di terze parti collegato o a cercare i risultati di tali modifiche

Problema: le modifiche nel mio repository di terze parti non vengono visualizzate in. CodeCatalyst

Possibili correzioni: CodeCatalyst attualmente non supporta il rilevamento delle modifiche nel ramo predefinito per i repository collegati. Per modificare il ramo predefinito per un repository collegato, è necessario prima scollegarlo da CodeCatalyst, modificare il ramo predefinito e quindi collegarlo nuovamente. Per ulteriori informazioni, consulta [Collegamento di GitHub repository, repository Bitbucket, repository di progetti e GitLab progetti Jira in CodeCatalyst](#).

Risoluzione dei problemi relativi agli account associati al tuo spazio

In CodeCatalyst, puoi aggiungerne uno Account AWS al tuo spazio per concedere autorizzazioni alle risorse e per scopi di fatturazione. Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni relativi agli account associati in CodeCatalyst

Argomenti

- [La mia richiesta di Account AWS connessione riceve un errore relativo al token non valido](#)
- [Il flusso di lavoro CodeCatalyst del mio progetto Amazon non funziona con un errore per l'account configurato, l'ambiente o il ruolo IAM](#)
- [Ho bisogno di un account, un ruolo e un ambiente associati per creare un progetto](#)
- [Non riesco ad accedere alla pagina Amazon CodeCatalyst Spaces nel Console di gestione AWS](#)
- [Voglio un account diverso come account di fatturazione](#)
- [Il flusso di lavoro del mio progetto fallisce a causa di un errore nel nome di connessione](#)

La mia richiesta di Account AWS connessione riceve un errore relativo al token non valido

Problema: quando si crea una richiesta di connessione con un token di connessione, la pagina non accetta il token e mostra un errore che indica che il token non è valido.

Possibili correzioni: assicurati di fornire l'ID dell'account che desideri aggiungere al tuo spazio. Per aggiungere l'account devi disporre delle autorizzazioni amministrative Account AWS o poter collaborare con l'amministratore.

Quando scegli di verificare l'account, si aprirà una nuova finestra del browser in Console di gestione AWS. È necessario accedere allo stesso account sul lato della console. Riprova dopo aver verificato quanto segue:

- Hai effettuato l'accesso Console di gestione AWS con lo stesso nome Account AWS che desideri aggiungere al tuo spazio.

- Hai effettuato l'accesso Console di gestione AWS con la regione impostata sulla regione corretta per il tuo spazio.
- Se sei arrivato dalla pagina di fatturazione e desideri aggiungerlo Account AWS come account di fatturazione specifico per il tuo spazio, assicurati che l'account non abbia raggiunto la quota di account di fatturazione per uno o più spazi.

Il flusso di lavoro CodeCatalyst del mio progetto Amazon non funziona con un errore per l'account configurato, l'ambiente o il ruolo IAM

Problema: quando il flusso di lavoro viene eseguito e non trova un account configurato o ruoli IAM associati al tuo spazio, devi compilare manualmente i campi ruolo, connessione e ambiente nel flusso di lavoro YAML. Visualizza l'azione del flusso di lavoro non riuscita e nota se i messaggi di errore sono i seguenti:

- Il ruolo non è disponibile per l'uso con la connessione associata all'ambiente.
- L'azione non è riuscita. Stato: NON RIUSCITO; Il valore fornito per la connessione o l'ambiente dell'account non è valido. Verifica che la connessione sia associata al tuo spazio e che l'ambiente sia associato al tuo progetto.
- L'azione non è riuscita. Stato: NON RIUSCITO; Il valore fornito per il ruolo IAM non è valido. Verifica che il nome esista, che il ruolo IAM sia stato aggiunto alla connessione del tuo account e che la connessione sia già associata al tuo CodeCatalyst spazio Amazon

Possibili correzioni: [assicurati che i campi YAML del flusso di lavoro abbiano valori accurati per Environment, Connections e Role](#). Le azioni del CodeCatalyst flusso di lavoro che richiedono un ambiente sono azioni di creazione o distribuzione che eseguono AWS risorse o generano pile di risorse. AWS

Scegli il blocco di azioni del flusso di lavoro non riuscito, quindi scegli Visual. Scegli la scheda Configurazione. Se i campi Ambiente, Nome connessione e Nome ruolo non sono compilati, dovrai aggiornare manualmente il flusso di lavoro. Utilizza i seguenti passaggi per modificare il tuo flusso di lavoro YAML:

- Espandi la `/.codecatalyst` directory, quindi espandi la `/workflows` directory. Aprire il file YAML del flusso di lavoro. Assicurati che i ruoli IAM e le informazioni sull'account siano specificati nello YAML che hai configurato per il tuo flusso di lavoro. Esempio:

```
Actions:
  cdk_bootstrap:
    Identifier: action-@v1
    Inputs:
      Sources:
        - WorkflowSource
    Environment:
      Name: Staging
    Connections:
      - Name: account-connection
        Role: build-role
```

Le proprietà Environment, Connection e Role sono necessarie per eseguire azioni di creazione e distribuzione CodeCatalyst del flusso di lavoro con risorse. AWS [Per un esempio, vedi l'azione di CodeCatalyst compilazione che fa riferimento ai parametri YAML per Environment, Connections e Role.](#)

- Assicurati che al tuo spazio sia stato aggiunto un account e assicurati che all'account siano stati aggiunti il ruolo o i ruoli IAM appropriati. Puoi modificare o aggiungere account se hai il ruolo di amministratore di Space. Per ulteriori informazioni, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS.](#)

Ho bisogno di un account, un ruolo e un ambiente associati per creare un progetto

Problema: nelle opzioni di creazione del progetto, al mio progetto non è disponibile un account aggiuntivo nel mio spazio oppure è necessario aggiungere un altro account al mio spazio da utilizzare.

Possibili correzioni: per il tuo spazio, puoi aggiungere l'autorizzazione Account AWS ad aggiungerli al tuo progetto se hai il ruolo di amministratore dello spazio. È inoltre necessario disporre di un account Account AWS in cui disporre delle autorizzazioni amministrative o poter lavorare con il proprio AWS amministratore.

Per assicurarti che un account e un ruolo siano disponibili nella schermata di creazione del progetto, devi prima aggiungere l'account e i ruoli. Per ulteriori informazioni, consulta [Consentire l'accesso alle AWS risorse con connessione Account AWS.](#)

Hai la possibilità di scegliere di creare un ruolo di servizio con una politica di ruolo chiamata `CodeCatalystWorkflowDevelopmentRole-spaceName` politica del ruolo. Il ruolo avrà un nome `CodeCatalystWorkflowDevelopmentRole-spaceName` con un identificatore univoco aggiunto. Per ulteriori informazioni sul ruolo e sulla politica relativa ai ruoli, vedere. [Comprensione del ruolo `CodeCatalystWorkflowDevelopmentRole-spaceName` del servizio](#) Per i passaggi per creare il ruolo, vedere [Creazione del `CodeCatalystWorkflowDevelopmentRole-spaceName` ruolo per il tuo account e il tuo spazio](#). Il ruolo viene aggiunto al tuo account e disponibile nelle pagine di creazione del progetto in CodeCatalyst.

Non riesco ad accedere alla pagina Amazon CodeCatalyst Spaces nel Console di gestione AWS

Problema: quando provo ad accedere alla CodeCatalyst pagina Amazon in Console di gestione AWS per aggiungere un account al mio CodeCatalyst spazio o aggiungere ruoli a un account in AWS, ricevo un errore di autorizzazione.

Possibili soluzioni.

Per il tuo spazio, puoi aggiungere l'autorizzazione Account AWS ad aggiungerli al tuo progetto se hai il ruolo di amministratore dello spazio. È inoltre necessario disporre di un account Account AWS in cui si dispone delle autorizzazioni amministrative o è possibile collaborare con AWS l'amministratore. Devi prima assicurarti di aver effettuato l'accesso Console di gestione AWS con lo stesso account che desideri gestire. Dopo aver effettuato l'accesso a Console di gestione AWS, puoi aprire la console e riprovare.

Aprire la CodeCatalyst pagina Amazon Console di gestione AWS nella [https://us-west-2.console.aws.amazon.com/codecatalyst/home_page?region=us-west-2#/.](https://us-west-2.console.aws.amazon.com/codecatalyst/home_page?region=us-west-2#/)

Voglio un account diverso come account di fatturazione

Problema: quando ho configurato il mio CodeCatalyst accesso, ho completato diversi passaggi per configurare il mio spazio e associare un account autorizzato Account AWS. Ora voglio autorizzare un altro account per la fatturazione.

Possibili correzioni: per il tuo spazio, puoi autorizzare gli account di fatturazione se disponi del ruolo di amministratore dello spazio. È inoltre necessario disporre di un account Account AWS in cui disporre delle autorizzazioni amministrative o poter collaborare con il proprio amministratore. AWS

Per ulteriori informazioni, consulta [Managing billing](#) nella Amazon CodeCatalyst Administrator Guide.

Il flusso di lavoro del mio progetto fallisce a causa di un errore nel nome di connessione

Problema: durante la creazione di un progetto e quindi l'esecuzione del flusso di lavoro del progetto, il flusso di lavoro non riesce e mostra un errore che indica che il nome della connessione non è valido, come segue:

Errore in: il nome della connessione non è valido. <action_name>

Possibili correzioni: assicurati di fornire l'ID dell'account che desideri aggiungere al tuo spazio e assicurati che l'account non sia abilitato per connessioni con account con restrizioni del progetto. Se l'account è abilitato per connessioni con account limitati al progetto, potrebbe essere necessario aggiornare la connessione all'account abilitando l'accesso al nuovo progetto. Per ulteriori informazioni, consulta [Configurazione delle connessioni di account con restrizioni del progetto](#).

Risoluzione dei problemi tra Amazon CodeCatalyst e il AWS SDKs AWS CLI

Le seguenti informazioni possono aiutarti a risolvere i problemi più comuni quando lavori con CodeCatalyst and the o il. AWS CLI AWS SDKs

Argomenti

- [Ricevo un errore quando aws codecatalyst accedo da una riga di comando o da un terminale che dice che si tratta di una scelta non valida](#)
- [Ricevo un errore relativo alle credenziali quando eseguo i comandi aws codecatalyst](#)

Ricevo un errore quando aws codecatalyst accedo da una riga di comando o da un terminale che dice che si tratta di una scelta non valida

Problema: quando provo a usare AWS CLI with CodeCatalyst, uno o più aws codecatalyst comandi non vengono riconosciuti come validi.

Soluzione: la causa più comune di questo problema è l'utilizzo di una versione di AWS CLI che non contiene gli aggiornamenti più recenti per i servizi e i comandi più recenti. Aggiorna l'installazione di AWS CLI e riprova. Per ulteriori informazioni, consulta [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#).

Ricevo un errore relativo alle credenziali quando eseguo i comandi aws codecatalyst

Problema: quando provo a utilizzare il comando AWS CLI with CodeCatalyst, ricevo un messaggio che indica `You can configure credentials by running "aws configure".` o. `Unable to locate authorization token`

Soluzione: è necessario configurare un AWS CLI profilo per utilizzare i CodeCatalyst comandi. Per ulteriori informazioni, consulta . [Configurazione per l'utilizzo AWS CLI di CodeCatalyst](#).

Comprendere lo stato attuale del servizio con il rapporto sullo stato di CodeCatalyst salute

Il rapporto sullo CodeCatalyst stato di salute di Amazon è una dashboard pubblica che fornisce agli utenti un elenco aggregato di up-to-the-minute notifiche riguardanti le prestazioni delle risorse e la disponibilità dei servizi CodeCatalyst che hanno un impatto diffuso. Puoi vedere quali risorse presentano problemi e possono influire sulle applicazioni in CodeCatalyst. Ciò consente di tenere traccia delle interruzioni e di altri tempi di inattività delle risorse a livello di sistema. Quando si verifica un incidente, sull'icona del rapporto sanitario appare un indicatore blu. Inoltre, invia CodeCatalyst automaticamente un avviso e una notifica via e-mail a tutti gli utenti con il ruolo di amministratore dello spazio nel progetto, fornendo dettagli e una cronologia dell'incidente quasi in tempo reale.

La dashboard fornisce un elenco di tutti gli eventi attivi e un registro di un massimo di 100 incidenti precedenti verificatisi negli ultimi 30 giorni. È possibile organizzare l'elenco degli incidenti in base alla data di aggiornamento dell'incidente. Puoi anche aggiornare l'elenco degli incidenti per aggiornamenti aggiornati al minuto.

Ecco un possibile flusso di lavoro per l'utilizzo del rapporto sullo CodeCatalyst stato di salute:

Mateo Jackson è uno sviluppatore di Budding Space con autorizzazioni di amministratore di Space. Durante il tentativo di creare una pull request, continua a ricevere un messaggio di errore. Controlla la sua posta elettronica e scopre di aver ricevuto un'e-mail generata automaticamente relativa a un incidente di sistema che CodeCatalyst fornisce una cronologia dettagliata del problema di sistema che ha interessato il suo spazio. Sceglie Visualizza aggiornamento e viene indirizzato al rapporto sullo stato di CodeCatalyst salute, dove può visualizzare tutti gli incidenti segnalati dal sistema. Sceglie l'incidente dall'elenco per ottenere ulteriori informazioni. Si apre una schermata divisa che fornisce un timestamp dell'ultimo aggiornamento, della cronologia, delle funzionalità interessate, dell'ora di inizio e dello stato attuale dell'incidente. Può anche vedere che il problema è in corso, ma il team di assistenza ha iniziato a lavorarci. Ogni volta che viene aggiornato la cronologia o lo stato dell'incidente, riceve un'e-mail. Nel caso in cui non abbia accesso alla sua e-mail, può scegliere l'icona a forma di campana nel pannello superiore per accedere al referto CodeCatalyst sanitario.

CodeCatalyst concetti del rapporto sullo stato di salute

L'apprendimento dei seguenti concetti ti aiuterà a comprendere il rapporto sullo CodeCatalyst stato di salute e in che modo ti consentono di tenere traccia dello stato di applicazioni, servizi e risorse.

Incidente

L'incidente è l'evento di sistema che influisce sulle applicazioni e sulle risorse interne CodeCatalyst. È possibile scegliere l'incidente per visualizzare una cronologia dettagliata dell'evento, inclusa l'ora in cui è iniziato e se il team di assistenza sta lavorando per risolverlo.

Stato

Lo stato è lo stato in tempo reale dell'incidente. Verrà visualizzato come In corso o Risolto.

Funzionalità impattate

Le funzionalità interessate sono le risorse o le applicazioni interessate dall'incidente. Un singolo incidente può interessare più aree del sistema, tra cui richieste pull, problemi, flussi di lavoro, test, implementazione e origine.

Aggiornato il

Aggiornato il fornisce un timestamp dell'ultimo aggiornamento relativo all'incidente.

Supporto per Amazon CodeCatalyst

Quando crei uno spazio, devi connetterne un Account AWS e designarlo come account di fatturazione per il tuo spazio. L'account Account AWS che hai designato come account di fatturazione è anche il luogo in cui accedi al tuo Supporto piano per Amazon. CodeCatalyst Se hai bisogno di assistenza, puoi creare casi di supporto da questa sezione designata. Account AWS

CodeCatalyst gli utenti in uno spazio utilizzano la CodeCatalyst pagina Supporto per Amazon per CodeCatalyst gestire i casi di supporto. È possibile passare a un Supporto piano come Business Support o Enterprise Support per creare e gestire casi di supporto CodeCatalyst tecnico in CodeCatalyst. L'assistenza è disponibile tramite telefono, web o chat per i casi di supporto.

Supporto Tramite Amazon è possibile supportare solo casi specifici del CodeCatalyst servizio e delle risorse CodeCatalyst. CodeCatalyst le risorse includono le risorse distribuite all'interno CodeCatalyst e dagli utenti in CodeCatalyst, ma non includono le risorse distribuite per altri servizi AWS o di terze parti. Se hai bisogno di supporto per qualsiasi altro AWS servizio, devi aprirlo tramite. Console di gestione AWS

Per modificare il piano di supporto, consulta [Modifica dei piani di supporto](#).

Note

I piani Developer Support non sono progettati per ambienti di produzione. Se un account di fatturazione dello spazio dispone di un piano Developer Support, questo piano non è disponibile a cascata per tutti gli amministratori dello spazio e i membri dello spazio all'interno di in. Supporto CodeCatalyst

Fatturazione Supporto per Amazon CodeCatalyst

Quando crei uno spazio in CodeCatalyst, gli utenti dello spazio possono creare e gestire casi di supporto Supporto per Amazon CodeCatalyst. Puoi creare due tipi di casi relativi ai clienti:

- I casi di assistenza relativi all'account e alla fatturazione sono disponibili per tutti CodeCatalyst gli utenti dello spazio. Puoi ricevere assistenza per domande sulla fatturazione e sull'account in base alle tue autorizzazioni in. CodeCatalyst

- Nei casi di supporto tecnico, puoi rivolgerti a un tecnico del supporto tecnico che ti aiuterà a risolvere problemi tecnici relativi al servizio e alle estensioni delle applicazioni di terze parti. Se hai sottoscritto un piano di supporto Basic, non puoi creare un caso di supporto tecnico.

L'account di fatturazione Account AWS designato per lo spazio deve disporre di un piano Business Support o Enterprise Support per lo spazio da utilizzare Supporto CodeCatalyst per i casi tecnici.

Note

Se il tuo spazio è utilizzato Supporto per Amazon CodeCatalyst da un account che non dispone di un piano Business Support o Enterprise Support, puoi comunque utilizzare Amazon Supporto CodeCatalyst per i casi relativi all'account e alla fatturazione.

Per il supporto tecnico, devi aprire tutti i casi tramite la CodeCatalyst console. Non è possibile creare casi CodeCatalyst di [Supporto](#) supporto tecnico per Console di gestione AWS.

Note

Le richieste di aumento del limite di servizio non sono disponibili Supporto per Amazon CodeCatalyst. Queste richieste possono essere inviate solo dall'utente root per l'account di fatturazione dello spazio in. AWS Support Center Console

Supporto per Amazon CodeCatalyst prevede gli stessi contratti di assistenza Supporto, con le seguenti considerazioni:

- Elenchi di gravità, tempi di risposta e SLAs casi di Supporto richiesta di assistenza CodeCatalyst, come descritto in Supporto dettaglio in [Scelta della gravità](#).
- Gli amministratori dello spazio e i membri dello spazio non possono utilizzare l' Supporto APIs AWS SDK o Supporto l'app di Slack per creare casi per. CodeCatalyst CodeCatalyst i casi di supporto possono essere inviati solo da. CodeCatalyst

 Note

CodeCatalyst non è completamente integrato con AWS Trusted Advisor il nostro AWS Incident Detection and Response. Verifica la modalità CodeCatalyst di integrazione per garantire che le tue pratiche aziendali siano allineate all'integrazione attuale.

Devi essere un utente dello spazio in cui desideri richiedere assistenza.

 Note

Se hai più di un costruttore nel tuo spazio, ti consigliamo di acquistare un piano Business Support o Enterprise Support. Questi piani forniscono supporto tecnico per lo spazio a un massimo di 5.000 costruttori.

L'account Account AWS designato come account di fatturazione per lo spazio utilizza il `AWSRoleForCodeCatalystSupport` ruolo e la politica [AmazonCodeCatalystSupportAccess](#) gestita. Ciò consente CodeCatalyst agli utenti che si trovano in uno spazio di accedere alla CodeCatalyst pagina Supporto per Amazon. Per ulteriori informazioni su questo ruolo e questa politica, consulta [AmazonCodeCatalystSupportAccess](#). Per altre considerazioni sulla fatturazione, consulta [Managing billing](#) nella Amazon CodeCatalyst Administrator Guide.

Ecco un possibile flusso per un builder che crea un caso di supporto in: CodeCatalyst

Mateo Jackson è uno sviluppatore di un progetto in. CodeCatalyst Dopo aver sottoscritto il servizio Account AWS che gestisce la fatturazione Supporto per Amazon CodeCatalyst e aver effettuato l'aggiornamento a un piano Business Support, tutti gli sviluppatori del settore possono creare casi di supporto tecnico. Mateo invia una richiesta di supporto tecnico per un flusso di lavoro non riuscito nel proprio progetto. Mateo utilizza la CodeCatalyst pagina Supporto per Amazon per compilare il modulo e creare un caso, fornendo l'ID del flusso di lavoro e altri dettagli nella richiesta. Il caso viene creato con un case ID e include l'ID account dell'account Account AWS designato come account di fatturazione e associato al piano di supporto per lo spazio.

Sebbene tutti i builder possano creare casi di assistenza in Supporto for CodeCatalyst, non ti viene addebitato alcun costo per ogni caso creato. Puoi aprire praticamente un numero illimitato di casi e contatti in base al piano Supporto Premium acquistato sul tuo account Space Billing.

 Note

L'account Space Billing è l'account Account AWS che ti viene addebitato per CodeCatalyst gli utenti e le risorse. Se hai utilizzato altri servizi Account AWS, contatta Supporto il Console di gestione AWS per ricevere assistenza sulle risorse distribuite ad altri servizi.

Puoi identificare quello su cui Account AWS sei stato distribuito dal flusso di lavoro.

Configurazione dello spazio Supporto per Amazon CodeCatalyst

Supporto for Amazon CodeCatalyst gestisce i casi di supporto come parte dell'integrazione delle Supporto API con CodeCatalyst.

Il `AWSRoleForCodeCatalystSupport` ruolo è un ruolo di servizio che viene utilizzato per i casi di supporto nel tuo settore. Il ruolo deve essere aggiunto all'account di fatturazione designato per lo spazio. Per ulteriori informazioni o per creare il ruolo, consulta [Creazione del `AWSRoleForCodeCatalystSupport` ruolo per il tuo account e il tuo spazio](#).

 Note

Per uno spazio creato prima del 20 aprile 2023, devi creare il ruolo in modo che il supporto possa lavorare CodeCatalyst per il tuo spazio. Se crei uno spazio dopo il 20 aprile 2023, puoi creare il ruolo durante la creazione dello spazio CodeCatalyst, nella pagina dei dettagli di fatturazione in o facendo clic sul link del banner di supporto in CodeCatalyst.

Per configurare il supporto per il tuo spazio

1. Quando crei uno CodeCatalyst spazio, ti viene richiesto di collegare un account di fatturazione. L'account di fatturazione designato per lo spazio verrà fatturato da. AWS Per ulteriori informazioni sulla creazione di uno spazio, consulta. [Creazione di un nuovo ruolo spaziale e di sviluppo \(a partire senza invito\)](#)
2. Quando si crea uno CodeCatalyst spazio, è disponibile l'opzione per creare il ruolo di `AWSRoleForCodeCatalystSupport` servizio che consente CodeCatalyst agli utenti di accedere al supporto. Il ruolo utilizza la politica gestita `AmazonCodeCatalystSupportAccess`. Il ruolo deve essere aggiunto all'account di fatturazione Account AWS designato per lo spazio. Per ulteriori informazioni sulla creazione di questo ruolo, consulta [Creazione del `AWSRoleForCodeCatalystSupport` ruolo per il tuo account e il tuo spazio](#).

3. Per l'account di fatturazione designato per lo spazio, si consiglia all'amministratore dello spazio di acquistare un piano Business Support o Enterprise Support per Account AWS. Tutti i membri dello spazio saranno in grado di gestire i casi di supporto per Amazon CodeCatalyst e i canali di supporto saranno allineati al piano di supporto acquistato in base al quale le integrazioni verranno completate.
4. Per creare e gestire i casi di supporto in CodeCatalyst, consulta [Creazione di una CodeCatalyst richiesta di supporto in CodeCatalyst](#)

Accesso al supporto per CodeCatalyst in Console di gestione AWS

Se l'account di fatturazione di uno spazio abilitato al supporto viene disconnesso, i casi di supporto associati all'account di fatturazione spaziale precedente e al piano di supporto associato non saranno più visibili in Amazon. L'utente root di quell'account di fatturazione può visualizzare e risolvere i vecchi casi in Console di gestione AWS e può impostare le autorizzazioni IAM di supporto per consentire ad altri utenti di visualizzare e risolvere i vecchi casi. Potrai comunque usufruire dei vantaggi del tuo piano di supporto rispetto a tutti gli altri Servizi AWS e completare eventuali casi di CodeCatalyst di supporto non risolti in precedenza. Console di gestione AWS

Per ulteriori informazioni, consulta [Aggiornamento, risoluzione e riapertura del caso nella Guida per l'utente di supporto](#).

I casi di assistenza per informazioni generali sulle procedure CodeCatalyst possono essere aperti anche in Console di gestione AWS, ma non è possibile ricevere supporto tecnico tramite questo canale. Per ulteriori informazioni, consulta [Creazione dei casi di supporto e gestione dei casi nella Guida](#) per l'utente di supporto.

Ecco un possibile flusso per un utente che risolve un caso di supporto CodeCatalyst in Console di gestione AWS:

Sebbene tutti i builder possano creare casi di assistenza con Supporto Amazon CodeCatalyst, le richieste di assistenza vengono fatturate dall'account designato come account di fatturazione per lo spazio. Mateo Jackson è uno sviluppatore di un progetto CodeCatalyst che ha aperto una richiesta di assistenza tecnica per un flusso di lavoro non riuscito nel proprio progetto. Tuttavia, l'account di fatturazione per lo spazio che era stato registrato ad Amazon CodeCatalyst e che aveva acquistato un piano Business Support è stato disconnesso dallo spazio. L'unico modo per Mateo di visualizzare le ultime comunicazioni e risolvere i casi aperti CodeCatalyst è gestire l'ID del caso dal Centro di supporto. A tale scopo, Mateo riceve le autorizzazioni IAM

dall'utente root del precedente account Space Billing associato alla richiesta di assistenza e risolve il caso tramite la console. Supporto

Important

Se modifichi l'account di fatturazione designato per il tuo spazio, il tuo Supporto piano sarà comunque accessibile fino alla fine del mese tramite Only. Console di gestione AWS Dovrai riacquistare Supporto sull'account di fatturazione aggiornato per continuare ad accedere ai casi di assistenza creati in precedenza in. CodeCatalyst Ti consigliamo di attendere di aver risolto tutti i casi di assistenza per modificare gli account di fatturazione degli spazi per evitare qualsiasi impatto sull'accesso ai casi di supporto tramite Supporto Amazon CodeCatalyst.

Creazione di una CodeCatalyst richiesta di supporto in CodeCatalyst

Puoi creare una richiesta di supporto nella CodeCatalyst pagina Supporto per Amazon.

Un AWS Builder ID può ricevere supporto solo per l'alias con cui è autenticato e solo per le risorse in base alle relative autorizzazioni. Le opzioni di account e fatturazione sono disponibili per tutti gli amministratori e i membri dello spazio. Tuttavia, gli utenti possono ricevere assistenza solo per le risorse a cui hanno accesso CodeCatalyst e non in relazione alla gestione della fatturazione dell'account.

Puoi creare un account e una richiesta di fatturazione o una richiesta di supporto tecnico per CodeCatalyst le tue risorse utilizzando la CodeCatalyst pagina Supporto per il tuo spazio.

Note

Solo i casi specifici del CodeCatalyst servizio e delle risorse possono essere supportati tramite Supporto Amazon CodeCatalyst. CodeCatalyst le risorse includono le risorse distribuite all'interno CodeCatalyst e dagli utenti in CodeCatalyst, ma non includono le risorse distribuite per altri servizi AWS o di terze parti. Se hai bisogno di supporto per qualsiasi altro AWS servizio, devi aprirlo tramite. Console di gestione AWS

Per creare una richiesta di supporto in CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Nella parte superiore della pagina, scegli il ? icona, quindi scegli Support.
4. Scegli Crea caso.
5. Selezionare una delle seguenti opzioni:
 - Account e fatturazione
 - Tecnico

Note

Supporto Per quanto riguarda Amazon CodeCatalyst, se viene aggiunto un piano Business Support o Enterprise Support all'account di fatturazione dello spazio, il supporto CodeCatalyst tecnico sarà disponibile per tutti gli amministratori dello spazio e i membri dello spazio. Per informazioni sulla risoluzione dei problemi, consulta [Non riesco a creare casi di supporto tecnico per il mio spazio](#).

Supporto i piani non si estendono su più spazi. Se sei membro di più spazi, l'amministratore dello spazio dovrà acquistare un piano Supporto Premium per ogni spazio per ricevere supporto tecnico in tutti gli spazi.

6. Scegli Service (Servizio), Category (Categoria), e Severity (Gravità). Per informazioni sulla scelta di una gravità, vedi [Scelta di una gravità](#).
 - Informazioni generali
 - Sistema impattato
 - Sistema di produzione impattato
 - Sistema di produzione spento
 - Sistema business-critical spento
7. Scegli Fase successiva: informazioni aggiuntive.

8. Nella pagina Informazioni aggiuntive, per Oggetto, inserisci un titolo relativo al problema.
9. Per Descrizione, segui le istruzioni per descrivere il caso, ad esempio:
 - Informazioni sulla risoluzione dei problemi specifiche CodeCatalyst, come l'ID del flusso di lavoro, i registri o le schermate
 - Ricezione di messaggi di errore
 - Procedure di risoluzione dei problemi seguite

Note

Non condividete informazioni sensibili relative alla corrispondenza dei casi, come credenziali, carte di credito, informazioni URLs firmate o di identificazione personale.

10. (Facoltativo) Scegli Attach files (Allega file) per aggiungere qualsiasi file pertinente al tuo caso, come registri di errori o screenshot. Puoi allegare fino a tre file. Ogni file può essere fino a 5 MB.
11. In Space name, viene visualizzato il nome dello spazio.
12. Nel nome Builder, il nome completo associato al tuo AWS Builder ID viene compilato automaticamente.
13. (Facoltativo) Scegliete il progetto in Nome progetto (se applicabile).

Note

Ti verranno mostrati solo i progetti per i quali hai le autorizzazioni. Se hai bisogno di accedere a un altro progetto, chiedi all'amministratore del progetto di fornirti l'accesso prima di creare una richiesta di supporto.

14. Scegli il passaggio successivo: Contattaci.
15. In Lingua di contatto preferita, scegli quella predefinita. Al momento è disponibile solo l'inglese.
16. Scegli l'opzione Web, Telefono o Chat per il metodo di contatto.
17. Controlla i dettagli del caso, quindi scegli Invia. Vengono visualizzati il numero di ID caso e il riepilogo.

La richiesta di assistenza viene creata a livello di spazio ed è visualizzabile da tutti i membri con accesso allo spazio e al progetto (se selezionati) definiti nella richiesta di supporto. Al momento non è possibile omettere una richiesta di supporto da parte dei singoli utenti.

Risoluzione di un caso di supporto in CodeCatalyst

Puoi risolvere i casi di supporto aperti dalla CodeCatalyst pagina Supporto per Amazon.

Devi avere un ruolo di amministratore o membro dello Space nello spazio in cui desideri risolvere un caso di supporto. Se non disponi del ruolo di amministratore dello spazio o se al momento della creazione del caso è stato selezionato un progetto, dovrai anche appartenere al progetto per visualizzare e risolvere il caso.

Per risolvere un caso di supporto aperto in CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accedi al tuo CodeCatalyst spazio.

Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Nella parte superiore della pagina, scegli il? icona, quindi scegli Supporto Amazon CodeCatalyst.
4. Scegli il link per il caso di supporto che desideri gestire. Seleziona Risolvi caso.

Riapertura di una richiesta di supporto in CodeCatalyst

Puoi utilizzare la funzione Riaprire un caso di supporto risolto dalla CodeCatalyst pagina Supporto per Amazon.

Note

Puoi riaprire il caso di supporto fino a 14 giorni dalla risoluzione del problema. Tuttavia, non è possibile riaprire un caso inattivo da più di 14 giorni. Se non riesci a riaprire il caso, apri un nuovo caso e includi l'ID del caso precedente come riferimento.

Se viene riaperto un caso esistente con informazioni diverse da quelle del problema attuale, l'operatore di supporto potrebbe chiederti di creare un nuovo caso.

Per riaprire una richiesta di assistenza in CodeCatalyst

1. Apri la CodeCatalyst console all'[indirizzo https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accedi al tuo CodeCatalyst spazio.

Tip

Se appartieni a più di uno spazio, scegli uno spazio nella barra di navigazione in alto.

3. Nella parte superiore della pagina, scegli il? icona, quindi scegli AWS Support for CodeCatalyst.
4. Scegli il link per il caso di supporto che desideri gestire. Scegli Reopen (Riapri). Scegli OK nella schermata di conferma, quindi scegli Invia.
5. Inserisci la descrizione con le informazioni più recenti sullo stesso problema. Non condividere informazioni sensibili nelle corrispondenze tra i casi, come credenziali, carte di credito, informazioni URLs firmate o di identificazione personale.

Quote per CodeCatalyst

La tabella seguente descrive le quote e i limiti per Amazon CodeCatalyst. Puoi trovare informazioni aggiuntive su aspetti specifici CodeCatalyst nei seguenti argomenti:

- [Quote per i repository di sorgenti in CodeCatalyst](#)
- [Quote per identità, autorizzazione e accesso in CodeCatalyst](#)
- [Quote per i flussi di lavoro in CodeCatalyst](#)
- [Quote per ambienti di sviluppo in CodeCatalyst](#)
- [Quote per progetti](#)
- [Quote per i progetti in CodeCatalyst](#)
- [Quote per gli spazi](#)
- [Quote per le emissioni in CodeCatalyst](#)

Numero massimo di spazi in un account	5
Numero massimo di spazi che un utente può creare in un mese di calendario	5
Numero minimo di Account AWS per spazio	1
Numero massimo di connessioni all'account per uno spazio	5.000
Numero massimo di account Account AWS di fatturazione per uno spazio	1
Numero massimo di spazi nel livello Gratuito per i quali è Account AWS possibile specificarne uno come account di fatturazione	3
Numero massimo di spazi in un livello a pagamento per i quali è Account AWS possibile specificarne uno come account di fatturazione	15

Numero massimo di connessioni VPC per uno spazio	100
Numero massimo di progetti in uno spazio	100
Numero massimo di progetti a cui un utente può appartenere	1.000
Descrizioni degli spazi	Le descrizioni degli spazi sono facoltative. Se specificate, devono avere una lunghezza compresa tra 0 e 200 caratteri. Possono contenere qualsiasi combinazione di lettere, numeri, spazi, punti, caratteri di sottolineatura, virgole, trattini e i seguenti caratteri speciali: ? & \$ % + = / \ ; : \n \t \r
Nomi degli spazi	I nomi degli spazi devono essere univoci CodeCatalyst. Non è possibile riutilizzare i nomi degli spazi eliminati. I nomi degli spazi devono avere una lunghezza compresa tra 3 e 63 caratteri. Devono inoltre iniziare con un carattere alfanumerico. I nomi degli spazi possono contenere qualsiasi combinazione di lettere, numeri, punti, trattini bassi e trattini. Non possono contenere nessuno dei seguenti caratteri: ! ? @ # \$ % ^ & * () + = { } [] / \ > < ~ ` ' " ; :

Cronologia dei documenti per Amazon CodeCatalyst

La tabella seguente descrive la cronologia della documentazione e gli aggiornamenti per la documentazione generale di CodeCatalyst.

Modifica	Descrizione	Data
Amazon non CodeCatalyst è più aperta a nuovi clienti	Amazon non CodeCatalyst è più aperta a nuovi clienti. Per ulteriori informazioni, consulta Come migrare da Amazon CodeCatalyst .	7 novembre 2025
Nuovo contenuto: nuova immagine universale di devfile	È stata aggiunta la documentazione per la nuova immagine universale di devfile ()Universal image 5.0.	30 luglio 2025
Contenuto aggiornato: privacy dei dati	È stata aggiornata la documentazione per includere informazioni aggiuntive sulla privacy dei dati relative a IAM Identity Center. Vedi Privacy dei dati in CodeCatalyst	28 luglio 2025
Contenuto aggiornato: deprecazione dei CodeCatalyst blueprint	Contenuto AWS del Project Development Kit (AWS PDK) rimosso dalla tabella Blueprint disponibili .	6 dicembre 2024
Nuovi contenuti: nuova immagine universale di devfile	È stata aggiunta la documentazione per la nuova immagine universale di devfile ()Universal image 4.0.	26 agosto 2024
Nuovi contenuti: Immagine di runtime utilizzata dall'azione 'deploy'AWS CDK, Immagine	Aggiunti due argomenti: Immagine di runtime utilizzata dall'azione 'deploy'AWS CDK	20 agosto 2024

[di runtime utilizzata dall'azione 'bootstrap'AWS CDK](#)

[eImmagine di runtime utilizzat a dall'azione 'bootstrap'AWS CDK.](#)

[Nuovo contenuto: immagini devfile universali obsolete](#)

È stata aggiunta documenta zione per le immagini devfile universali obsolete (e).
Universal image 1.0
Universal image 2.0

16 agosto 2024

[Contenuto aggiornato: deprecazione dei CodeCatalyst blueprint](#)

[Il Serverless Application Model \(SAM\) e il contenuto del servizio Video-on-demand Web sono stati rimossi dalla tabella Blueprint disponibili.](#)

14 agosto 2024

[Contenuto aggiornato: Immagini attive](#)

È stato aggiornato l'[Immagini attive](#) argomento per indicare che alcune azioni potrebbero o utilizzare un'immagine di novembre 2022 o marzo 2024. È stata inoltre aggiornata la documentazione di ciascuna azione per indicare quale immagine viene utilizzata.

9 agosto 2024

[Contenuto aggiornato: risoluzione dei problemi relativi al nome utente troncato](#)

Sono stati aggiornati gli argomenti sulla risoluzione dei problemi per includere informazioni sulla risoluzione dei nomi utente che potrebbero o essere troncati in alcuni casi.

31 luglio 2024

[Contenuti aggiornati: Creazione di un ambiente](#)

È stata aggiornata la [Creazione di un ambiente](#) sezione per menzionare CodeCatalyst i ruoli.

25 luglio 2024

Contenuto aggiornato: Condivisione di artefatti e file tra azioni	È stato aggiornato il Riferimen to ai file in un artefatto sottoargomento per includere informazioni sui gruppi di azioni. È stato inoltre aggiunto il Esempio: fare riferimento a un file in un artefatto quando è presente un gruppo di azioni sottoargomento.	18 luglio 2024
Contenuto aggiornato: Crea, testa e distribuisce con i flussi di lavoro	È stata aggiornata la documentazione per indicare che la definizione del flusso di lavoro .yaml o l'estensione del .yam1 file devono essere in lettere minuscole.	15 luglio 2024
Contenuto aggiornato: è stata aggiunta la risoluzione dei problemi per gli utenti SSO durante la creazione di un ambiente di sviluppo in uno spazio CodeCatalyst	È stato aggiunto un argomento sulla risoluzione dei problemi che include informazioni sui problemi riscontrati dagli utenti SSO che hanno effettuato l'accesso a uno spazio con federazione delle identità durante la creazione di ambienti di sviluppo.	12 luglio 2024
Contenuto aggiornato: Creazione di un flusso di lavoro	È stato aggiornato l'argomen to per indicare che il file di definizione del flusso di lavoro può essere archiviato in una sottodirectory di ~/.codeca talyt/workflows/ .	12 luglio 2024

Contenuto aggiornato: Stati del flusso di lavoro in CodeCatalyst	Sono state aggiunte istruzioni su come nascondere i flussi di lavoro inattivi nella CodeCatalyst console.	12 luglio 2024
Nuovi contenuti: Configura zione dei trigger solo manuali	È stato aggiunto un Configurazione dei trigger solo manuali argomento.	26 giugno 2024
Contenuto aggiornato: Creazione di un ambiente	Sono state aggiornate la Creazione di un ambiente sezione e molte altre per riflettere una nuova funzionalità che consente di specificare un ruolo IAM predefinito durante la creazione di un ambiente CI/CD. Questo ruolo verrà assegnato a tutte le azioni del flusso di lavoro associate all'ambiente. Non è più necessario assegnare i ruoli IAM direttamente alle azioni.	21 giugno 2024
Nuovi contenuti: Tutorial: Estrai da un archivio di pacchetti	È stato aggiunto un tutorial che descrive come configurare un flusso di lavoro per estrarre le dipendenze da un archivio di CodeCatalyst pacchetti.	20 giugno 2024

[Contenuto aggiornato: utilizza gli archivi dei GitLab progetti con i blueprint](#)

Documentazione aggiornata per la possibilità di creare un repository di GitLab progetto quando [Creare un progetto con un blueprint](#)[Aggiungere un modello in un progetto per integrare le risorse](#), o [creare un](#) blueprint personalizzato.

19 giugno 2024

[Contenuto aggiornato: Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa](#)

È stato aggiornato il tutorial per mostrare la possibilità di utilizzare Amazon Q per scegliere un blueprint durante la creazione di un progetto o l'aggiunta di un blueprint a un progetto esistente.

18 giugno 2024

[Contenuto aggiornato: creazione di un archivio di sorgenti](#)

È stata aggiornata la documentazione per includere informazioni sulla creazione di repository vuoti.

18 giugno 2024

[Nuovi contenuti: creazione di un progetto CodeCatalyst con Amazon Q](#)

Sono state aggiunte nuove sezioni [Creare un progetto](#) intitolate [Le migliori pratiche per l'utilizzo di Amazon Q per creare progetti o aggiungere funzionalità con blueprint](#) e [Le migliori pratiche per utilizzare i blueprint con i progetti](#) con istruzioni per creare progetti e aggiungere progetti con Amazon Q.

18 giugno 2024

[Nuovi contenuti: clona un repository Git esistente in un repository sorgente](#)

È stato aggiunto un nuovo argomento su come usare Git per inviare un clone mirror o un repository locale a un repository sorgente vuoto in. CodeCatalyst

18 giugno 2024

[Nuovi contenuti: supporto per i tipi di pacchetto Maven NuGet e Python](#)

È stata aggiunta la documentazione per l'utilizzo dei pacchetti Maven NuGet e Python in. CodeCatalyst

17 giugno 2024

[Aggiornamenti agli spazi e alle connessioni degli account](#)

Ora puoi connettere l'account Account AWS specificato come account di fatturazione a più CodeCatalyst spazi. Per gli spazi configurati per la federazione delle identità, un'istanza di Identity Center può essere connessa a più spazi. Consulta le sezioni [Consentire l'accesso alle AWS risorse con connessione Account AWS](#), [Risoluzione dei problemi relativi agli account associati al tuo spazio](#) e [Quote per CodeCatalyst](#).

13 giugno 2024

[Contenuto aggiornato: utilizzo dei ruoli](#)

È stata aggiornata la documentazione per i ruoli per includere le autorizzazioni per l'utilizzo di Amazon Q per consigliare e creare attività per un problema.

13 giugno 2024

Contenuto aggiornato: utilizzo dei ruoli	È stata aggiornata la documentazione relativa ai ruoli per includere le autorizzazioni per la creazione, l'aggiornamento e la rimozione di collegamenti tra i problemi.	13 giugno 2024
Contenuto aggiornato: Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa	È stato aggiornato il tutorial per includere informazioni sull'uso di Amazon Q per consigliare attività per un problema.	13 giugno 2024
Contenuto aggiornato: gestione delle attività relative ai problemi	Sono state aggiunte informazioni sull'utilizzo di Amazon Q per consigliare attività per un problema in CodeCatalyst.	13 giugno 2024
Nuovi contenuti: collega un problema a un altro	È stato aggiunto un nuovo argomento sul collegamento di problemi ad altri problemi in CodeCatalyst.	13 giugno 2024
Contenuto aggiornato: AWS CDK azione 'distribuisci' YAML	È stata corretta la descrizione della Region proprietà.	12 giugno 2024
Contenuto aggiornato: utilizza repository di terze parti con progetti	Documentazione aggiornata per la possibilità di creare un GitHub repository quando Creare un progetto con un blueprint Aggiungere un modello in un progetto per integrare le risorse , o di creare un blueprint personalizzato.	6 giugno 2024

Nuovi contenuti: sono stati aggiunti passaggi per lavorare con le connessioni personali	Sono stati aggiunti passaggi per la creazione e l'eliminazione delle connessioni personali. Le connessioni personali ti consentono di gestire GitHub le risorse per progetti e blueprint in Amazon CodeCatalyst.	6 giugno 2024
Nuovi contenuti: estensione dei repository Bitbucket	Aggiunti nuovi contenuti per l'utilizzo dell'estensione dei repository Bitbucket in CodeCatalyst	5 giugno 2024
Nuovi contenuti: Tipi di operazione	È stato aggiornato l' Operazioni di terze parti argomento per menzionare l'azione SonarCloud Scan.	29 maggio 2024
Contenuto aggiornato: AWS CDK azione 'distribuisci' YAML	È stato corretto l'CdkRootRootPath esempio.	28 maggio 2024
Contenuti aggiornati	Titoli degli argomenti aggiornati e contenuto riorganizzato per migliorare la leggibilità e la scoperta. Se vuoi fornire feedback su queste modifiche , utilizza questo link Fornisci feedback .	17 maggio 2024
Nuovi contenuti: visualizzazione della cronologia delle modifiche a un file	È stata aggiornata la documentazione per riflettere le nuove funzionalità di visualizzazione della cronologia delle modifiche a un file in un archivio di origine.	1° maggio 2024

Contenuto aggiornato: Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa	È stato aggiornato il tutorial per riflettere l'integrazione con Amazon Q Developer.	29 aprile 2024
Contenuto aggiornato: Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa	Il tutorial è stato aggiornato in modo da consentire ad Amazon Q di analizzare e i problemi in base alla complessità, suggerire e creare attività e lavorare sulle attività relative a un problema.	22 aprile 2024
Nuovi contenuti: Garantire l'esecuzione di un flusso di lavoro	Aggiunti Garantire l'esecuzione di un flusso di lavoro e diversi altri argomenti relativi alle approvazioni dei flussi di lavoro. Richiesta di approvazioni per l'esecuzione del flusso di lavoro	22 aprile 2024
Nuovi contenuti: utilizzo delle attività per suddividere i problemi in obiettivi più piccoli	Contenuti aggiunti per supportare l'avvio delle attività nei numeri. È possibile aggiungere attività ai numeri per suddividere ulteriormente, organizzare e tenere traccia del lavoro relativo a quel problema.	4 aprile 2024

Contenuti aggiornati: Condivisione di artefatti e file tra azioni	L' Condivisione di artefatti e file tra azioni argomento è stato aggiornato per includere due nuovi argomenti secondari: Posso condividere gli artefatti senza specificarli come output e input? e Posso condividere artefatti tra flussi di lavoro?	2 aprile 2024
Contenuto aggiornato: Limitazioni delle GitHub azioni in CodeCatalyst	È stato aggiornato l' Limitazioni delle GitHub azioni in CodeCatalyst argomento per indicare che GitHub le azioni vengono eseguite su un'immagine Docker di un ambiente di runtime precedente.	2 aprile 2024
Nuovi contenuti: AWS CDK azione 'distribuisci' YAML	È stata aggiunta una nuova <code>CloudAssemblyRootPath</code> proprietà a AWS CDK azione 'distribuisci' YAML .	1 aprile 2024
Contenuti aggiornati: Specificazione delle immagini dell'ambiente di runtime	L' Specificazione delle immagini dell'ambiente di runtime argomento è stato aggiornato per includere informazioni sulla nuova immagine dell'ambiente di runtime di marzo 2024.	26 marzo 2024

[Contenuto aggiornato: utilizzo dei ruoli](#)

Informazioni consolidate sulle autorizzazioni dei ruoli in un'unica tabella. La tabella si trova in un nuovo [Visualizzazione delle autorizzazioni disponibili per ogni ruolo](#) argomento.

18 marzo 2024

[Nuovi contenuti: Visualizzazione di tutti gli spazi e i progetti per un utente](#)

Sono state aggiunte informazioni sulla visualizzazione di un elenco nella home page dell'utente che mostra ogni CodeCatalyst spazio o progetto per l'utente che ha effettuato l'accesso. CodeCatalyst Per informazioni, consulta [Visualizzazione di tutti gli spazi e i progetti per un utente](#).

18 marzo 2024

[Nuovi contenuti: Esempio: un grilletto con trazione e ramificazioni](#)

Aggiunto un esempio di trigger di pull request. Sono state apportate piccole correzioni in tutto l'[L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#) argomento.

11 marzo 2024

[Contenuto aggiornato: utilizzo dei ruoli](#)

È stata aggiornata la documentazione relativa ai ruoli per includere le autorizzazioni per la creazione, l'eliminazione e la visualizzazione degli ambienti.

4 marzo 2024

Contenuto aggiornato: Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa	È stato aggiornato il tutorial per riflettere le modifiche durante la creazione e l'assegnazione di problemi ad Amazon Q.	4 marzo 2024
Nuovi contenuti: componenti Issues	Sono stati aggiunti nuovi contenuti su come lavorare con i componenti dei problemi come sviluppatore di blueprint personalizzati.	27 febbraio 2024
Contenuti aggiornati: Tipi di operazione	È stato aggiornato l' CodeCatalyst Azioni di laboratorio argomento per includere un elenco di azioni di CodeCatalyst Labs.	21 febbraio 2024
Contenuto aggiornato: utilizzo delle richieste pull	La documentazione è stata aggiornata per riflettere le nuove funzionalità con regole di approvazione e requisiti prioritari per unire una pull request.	15 febbraio 2024
Contenuto aggiornato: unione delle richieste pull	È stata aggiunta la documentazione per le richieste pull che include informazioni sull'annullamento dei requisiti di unione per unire una richiesta pull che non ha ancora ricevuto l'approvazione dei revisori obbligatori o non ha ancora soddisfatto le regole di approvazione.	15 febbraio 2024

Nuovi contenuti: gestione delle regole di approvazione	È stata aggiunta la documentazione per le pull request che include informazioni sulla creazione e la gestione delle regole di approvazione.	15 febbraio 2024
Contenuto aggiornato: utilizzo dei ruoli	È stata aggiornata la documentazione per i ruoli in modo da includere le autorizzazioni per l'utilizzo delle regole di approvazione e delle richieste pull.	14 febbraio 2024
Contenuti aggiornati: Come posso correggere gli errori «La definizione del flusso di lavoro presenta <i>n</i> errori»?	È stata aggiornata la Come posso correggere gli errori «La definizione del flusso di lavoro presenta <i>n</i> errori»? sezione per includere ulteriori suggerimenti per la risoluzione dei problemi.	9 febbraio 2024
Nuovi contenuti: Visualizzazione dello stato di un flusso di lavoro	È stata aggiunta una sezione che descrive gli stati del flusso di lavoro.	9 febbraio 2024
Nuovi contenuti: Visualizzazione dello stato di un flusso di lavoro	È stata aggiunta una sezione che descrive gli stati del flusso di lavoro.	9 febbraio 2024
Aggiorna il contenuto: Quote per i flussi di lavoro in CodeCatalyst	È stato aggiornato l' Quote per i flussi di lavoro in CodeCatalyst argomento con il Numero massimo di azioni per flusso di lavoro e il Numero massimo di ambienti associati a quote Account AWS per spazio.	7 febbraio 2024

Contenuto aggiornato: Creazione di un ambiente	È stata aggiornata la Creazione di un ambiente sezione per indicare che è possibile utilizzare al massimo una connessione a un account per ambiente.	31 gennaio 2024
Nuovo contenuto: archivio di blueprint GitHub personalizzati	Aggiunto nuovo contenuto per il GitHub repository che viene reso disponibile pubblicamente.	10 gennaio 2024
Contenuto aggiornato: configurazione di npm con CodeCatalyst	Istruzioni generali di configurazione aggiornate per l'utilizzo di npm con CodeCatalyst e aggiunta chiarezza sull'opzione. <code>always-auth=true</code>	5 gennaio 2024
Contenuto aggiornato: utilizzo delle richieste pull	La documentazione è stata aggiornata per riflettere le nuove funzionalità con le funzionalità di intelligenza artificiale generativa di CodeCatalyst.	28 novembre 2023
Contenuto aggiornato: creazione di un problema	La documentazione è stata aggiornata per riflettere le nuove funzionalità con le funzionalità di intelligenza artificiale generativa di CodeCatalyst.	28 novembre 2023
Nuovi contenuti: Tutorial: Utilizzo delle funzionalità di intelligenza artificiale generativa	È stato aggiunto un tutorial per l'utilizzo delle funzionalità di intelligenza artificiale generativa in Amazon CodeCatalyst.	28 novembre 2023

Nuovi contenuti: progetti personalizzati e gestione del ciclo di vita	Sono stati aggiunti nuovi contenuti per l'utilizzo di blueprint personalizzati e funzionalità di gestione del ciclo di vita in Amazon. CodeCatalyst	27 novembre 2023
Contenuti aggiornati: Tutorial: creazione di un progetto con il moderno modello di applicazioni web a tre livelli	È stato aggiornato il tutorial con correzioni e informazioni sulla risoluzione dei problemi.	22 novembre 2023
Contenuto aggiornato: L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger	Sono stati corretti alcuni esempi e descrizioni relativi ai trigger delle pull request. È stata aggiunta una Linee guida per l'utilizzo di trigger e filiali sezione.	22 novembre 2023
Nuovi contenuti: Accesso con SSO	Sono state aggiunte informazioni sull'accesso con Single Sign-On (SSO) e collegamenti a informazioni sulla configurazione e la gestione di uno CodeCatalyst spazio che supporti la federazione delle identità. Consulta Configura e accedi a CodeCatalyst e Accesso con SSO .	17 novembre 2023
Contenuto aggiornato: utilizzo dei ruoli	È stata aggiornata la documentazione per i ruoli per includere le autorizzazioni per lavorare con team, connessioni VPC, Single Sign-On e risorse della macchina.	16 novembre 2023

Contenuto aggiornato: utilizzo delle richieste pull	La documentazione è stata aggiornata per riflettere i cambiamenti nella modalità di visualizzazione delle modifiche relative a una pull request.	16 novembre 2023
Contenuto aggiornato: quote per CodeCatalyst	È stato aggiornato l' Quote per CodeCatalyst argomento con il numero massimo di connessioni VPC per una quota di spazio.	16 novembre 2023
Nuovo argomento sulla migrazione per CodeCatalyst	Nuovo argomento sulla migrazione.	16 novembre 2023
Nuovi contenuti: gestione dei team per uno spazio e per CodeCatalyst i progetti	Sono state aggiunte informazioni sull'utilizzo dei team con spazi. Consulta Consentire l'accesso allo spazio tramite team e Consentire l'accesso al progetto tramite team .	16 novembre 2023
Nuovi contenuti: gestione delle risorse delle macchine per progetti e flussi di lavoro in uno spazio	Sono state aggiunte informazioni sull'utilizzo delle risorse della macchina con gli spazi. Per informazioni, consulta Consentire l'accesso allo spazio per le risorse della macchina .	16 novembre 2023
Nuovi contenuti: gestione delle risorse delle macchine per progetti e flussi di lavoro in un progetto CodeCatalyst	Sono state aggiunte informazioni sull'utilizzo delle risorse della macchina con CodeCatalyst i progetti. Per informazioni, consulta Consentire l'accesso al progetto per le risorse della macchina .	16 novembre 2023

Nuovi contenuti: associazione di una connessione VPC a un ambiente	È stata aggiunta la documentazione per l'associazione di una connessione VPC a un ambiente, che può essere utilizzata in un flusso di lavoro.	16 novembre 2023
Nuovi contenuti: associazione di una connessione VPC a un ambiente di sviluppo	È stata aggiunta documentazione per l'utilizzo degli ambienti di sviluppo con una connessione VPC.	16 novembre 2023
Nuovo contenuto	Pubblicazione iniziale dell' Amazon CodeCatalyst Administrator Guide .	16 novembre 2023
Nuovi contenuti: AWS CDK azione 'distribuisci' YAML	È stata aggiunta una nuova <code>CdkCliVersion</code> proprietà al AWS CDK azione 'distribuisci' YAML e al azione 'AWS CDK bootstrap' YAML .	14 novembre 2023
Contenuto aggiornato: Utilizzo dei ruoli	È stata aggiornata la documentazione per i ruoli in modo da includere le autorizzazioni per l'utilizzo delle regole delle filiali.	13 novembre 2023
Contenuto aggiornato: risoluzione dei problemi relativi agli archivi di origine, ai flussi di lavoro e agli ambienti di sviluppo	Sono stati aggiornati gli argomenti sulla risoluzione dei problemi per includere informazioni sull'utilizzo delle regole delle filiali.	13 novembre 2023
Contenuto aggiornato: Crea e testa azioni YAML	È stata aggiornata la documentazione relativa alla <code>Environment</code> proprietà. Ora è un campo opzionale per le azioni di compilazione e test.	13 novembre 2023

Nuovo contenuto: gestione delle regole delle filiali	È stata aggiunta la documentazione per le filiali, che include informazioni sulla visualizzazione di eventuali regole per le filiali in un archivio di origine e sulla creazione e gestione delle regole delle filiali.	13 novembre 2023
Contenuto aggiornato: utilizzo delle richieste pull	La documentazione è stata aggiornata per riflettere i cambiamenti nella modalità di visualizzazione delle informazioni su una pull request.	10 novembre 2023
Contenuto aggiornato: Memorizzazione nella cache dei file tra le esecuzioni del workflow	È stata aggiornata la documentazione per includere le limitazioni della memorizzazione nella cache dei file.	10 novembre 2023
Contenuto aggiornato: Tutorial: distribuzione di un'applicazione su Amazon EKS	È stata aggiornata la documentazione per menzionare il progetto EKS App Deployment.	9 novembre 2023
Nuovo contenuto: pacchetti in CodeCatalyst	È stata aggiunta documentazione per l'utilizzo dei pacchetti in CodeCatalyst.	1 novembre 2023
Contenuti nuovi e aggiornati: utilizzo dei ruoli	È stata aggiornata la documentazione per quattro nuovi ruoli in CodeCatalyst: Power user, Limited access, Reviewer e Read only.	1 novembre 2023

Contenuto aggiornato: Esportazione dei parametri GitHub di output	Aggiornati gli esempi per utilizzare il file di GITHUB_OUTPUT ambiente anziché il set-output comando. L'uso dei file di ambiente GitHub è il metodo consigliato per impostare i parametri di output.	24 ottobre 2023
Nuovi contenuti: L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger	Documentazione aggiunta per i trigger di pianificazione.	16 ottobre 2023
Contenuto aggiornato: Azione «Distribuisci su cluster Kubernetes» YAML	Sono state aggiunte informazioni sull'utilizzo del CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> ruolo negli Tutorial: distribuzione di un'applicazione su Amazon EKS argomenti Azione «Distribuisci su cluster Kubernetes» YAML e.	22 settembre 2023

[Contenuto aggiornato: nuovo nome del ruolo e nuova politica per il CodeCatalystWorkflowDevelopmentRole-*spaceName* ruolo](#)

Sono stati aggiornati i passaggi e le descrizioni dei ruoli per la modifica del nome del ruolo di sviluppatore in CodeCatalystWorkflowDevelopmentRole-*spaceName* . Il ruolo di sviluppatore ora utilizza la politica AdministratorAccess AWS gestita. Consulta [Comprensione del ruolo CodeCatalystWorkflowDevelopmentRole-*spaceName* del servizio e Creazione di un nuovo ruolo spaziale e di sviluppo \(a partire senza invito\)](#).

20 settembre 2023

[Contenuti aggiornati: Utilizzo delle variabili nei flussi di lavoro](#)

Sono stati introdotti due nuovi concetti: variabili definite dall'utente e variabili predefinite. Questi concetti dovrebbero rendere la [Utilizzo delle variabili nei flussi di lavoro](#) sezione più facile da leggere e comprendere.

19 settembre 2023

[Contenuto aggiornato: utilizzo dei commit](#)

La documentazione è stata aggiornata per riflettere la modifica delle informazioni visualizzate e fornire dettagli sulla visualizzazione dei commit con più genitori.

7 settembre 2023

Nuovi contenuti: L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger	È stato aggiunto il seguente esempio all' L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger argomento: Esempio: un semplice trigger «push to main»	6 settembre 2023
Contenuto aggiornato: utilizzo delle richieste pull	La documentazione è stata aggiornata per riflettere la modifica dell'ordine di visualizzazione del ramo di origine e del ramo di destinazione durante la creazione di una richiesta pull.	30 agosto 2023
Nuovo contenuto: visualizza e modifica il ramo predefinito	È stata aggiunta la documentazione per le filiali che include informazioni sulla visualizzazione e la modifica del ramo predefinito per un repository di origine.	30 agosto 2023
Contenuto aggiornato: Azione «Distribuisci su cluster Kubernetes» YAML	È stata aggiunta una nota su Helm e Kustomize alla descrizione della Manifests proprietà in. Azione «Distribuisci su cluster Kubernetes» YAML	15 agosto 2023
Nuovo contenuto: gestione degli allegati dei problemi	È stata aggiunta documentazione per l'utilizzo e la gestione degli allegati relativi ai problemi.	15 agosto 2023

[Contenuti aggiornati: L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#)

Migliorata e ampliata la documentazione relativa ai trigger del flusso di lavoro.

11 agosto 2023

[Nuovi contenuti: risoluzione dei problemi relativi alle autorizzazioni dei ruoli](#)

Sono state aggiunte informazioni sull'aggiornamento delle autorizzazioni dei ruoli per eseguire un flusso di lavoro che richiede l'accesso ad Amazon CodeGuru. Per informazioni, consulta [Il moderno flusso di lavoro del blueprint di applicazioni Web a tre livelli OnPullRequest non riesce a causa di un errore di autorizzazione per Amazon CodeGuru](#).

11 agosto 2023

[Nuovi contenuti: Come posso correggere i messaggi «Workflow is inactive»?](#)

È stato aggiunto il seguente argomento di risoluzione dei problemi: [Come posso correggere i messaggi «Workflow is inactive»?](#)

11 agosto 2023

[Nuovi contenuti: Implementazione su Amazon EKS con un flusso di lavoro](#)

Documentazione aggiunta per l'azione del cluster Deploy to Kubernetes. Per ulteriori informazioni, consultare [Implementazione su Amazon EKS con un flusso di lavoro](#) e [Tutorial: distribuzione di un'applicazione su Amazon EKS](#).

27 luglio 2023

[Aggiornamenti sulla modalità di registrazione degli eventi di gestione per uno spazio CodeCatalyst](#)

Sono state aggiunte informazioni su come vengono registrati gli eventi di gestione per azioni specifiche in uno CodeCatalyst spazio con. AWS CloudTrail Sono state aggiunte informazioni su come tutti gli eventi in uno spazio possono essere visualizzati con il list-event-logs comando. Per informazioni, consulta [Monitoraggio degli eventi e delle chiamate API tramite la registrazione.](#)

20 luglio 2023

[Contenuto aggiornato: L'avvio di un flusso di lavoro viene eseguito automaticamente utilizzando i trigger](#)

È stata aggiornata la documentazione per indicare che i trigger di pull request sono ora supportati nei repository di GitHub origine. In precedenza, i trigger di pull request erano supportati solo con CodeCatalyst i repository di origine.

14 luglio 2023

[Contenuto aggiornato: Quote per i flussi di lavoro in CodeCatalyst](#)

È stato aggiornato l'[Quote per i flussi di lavoro in CodeCatalyst](#) argomento con la quota massima di tempo per cui un'azione può essere eseguita.

27 giugno 2023

[Contenuto aggiornato: Definizione YAML del flusso di lavoro](#)

È stato corretto un errore di formattazione nel blocco di Compute codice.

27 giugno 2023

Contenuto aggiornato: protezione dei dati	È stata aggiornata la documentazione per includere informazioni aggiuntive sulla replica dei dati.	26 giugno 2023
Nuovi contenuti: Specifica re la versione dell'azione da utilizzare	È stato aggiunto un Specifica re la versione dell'azione da utilizzare argomento.	21 giugno 2023
Contenuto aggiornato: Implementazione in e Account AWS VPCs	Ha chiarito la Quali azioni supportano la visualizz azione delle informazioni sulla distribuzione in? CodeCatalyst sezione.	14 giugno 2023
Contenuto aggiornato: documentazione sui problemi riorganizzata	È stata riorganizzata la documentazione relativa alla maggior parte dei problemi per allinearla meglio alla documentazione generale e ai flussi degli utenti.	31 maggio 2023
Contenuto aggiornato: Issues view switcher	Sono stati aggiornati diversi flussi utente per allinearli allo switcher di visualizzazione dei problemi aggiornato.	31 maggio 2023
Contenuto aggiornato: gestione delle notifiche	È stata aggiornata la documentazione per le notifiche per includere informazioni sulla configurazione delle notifiche personali di Slack.	30 maggio 2023

[Contenuto aggiornato:
gestione delle notifiche](#)

È stata aggiornata la documentazione per le notifiche per includere informazioni sulla configurazione delle notifiche personali di Slack.

30 maggio 2023

[Nuovi contenuti: CodeCatalyst
modello di fiducia](#)

Aggiunto un nuovo argomento con informazioni sul modello di fiducia, che consente di CodeCatalyst assumere il ruolo di servizio nell'ambiente connesso Account AWS. È stata aggiunta una nuova sezione sui principali di servizio definiti per CodeCatalyst. Per informazioni, consulta [Comprendere il modello CodeCatalyst di fiducia](#).

20 maggio 2023

[Contenuto aggiornato:
Connessione degli archivi di
origine ai flussi di lavoro](#)

Sono state semplificate le istruzioni in [Riferimento ai file del repository di origine](#).

10 maggio 2023

[Contenuto aggiornato:
Quote per i flussi di lavoro in
CodeCatalyst](#)

È stato aggiornato l'[Quote per i flussi di lavoro in CodeCatalyst](#) argomento con la lunghezza massima di una quota di valori variabili di output.

10 maggio 2023

Nuovi contenuti: Condivisione di artefatti e file tra azioni	Sono stati aggiunti due esempi: Esempio: fare riferimento a un file in un singolo artefatto e Esempio: fare riferimento a un file in un artefatto quando è presente un WorkflowSource .	10 maggio 2023
Contenuto aggiornato: utilizzo delle richieste pull	È stata aggiornata la documentazione per le richieste pull per includere informazioni sulla configurazione delle preferenze e-mail per gli eventi di pull request.	21 aprile 2023
Contenuto aggiornato: gestione delle notifiche	È stata aggiornata la documentazione per le notifiche per includere informazioni sulla configurazione delle preferenze e-mail per gli eventi di pull request.	21 aprile 2023
Aggiornamenti alle politiche gestite	Sono state aggiunte e AWS politica gestita: AmazonCodeCatalystSupportAccess gestite le politiche. AWS politica gestita: AmazonCodeCatalystFullAccess AWS politica gestita: AmazonCodeCatalystReadOnlyAccess Per informazioni, consulta CodeCatalyst aggiornamenti alle politiche gestite AWS .	20 aprile 2023
Nuovi contenuti: Rimozione di un obiettivo di distribuzione	È stato aggiunto un Rimozione di un obiettivo di distribuzione argomento.	20 aprile 2023

Nuovi contenuti: Tipi di operazione	È stato aggiunto un CodeCatalyst azioni argomento.	20 aprile 2023
Aggiornamenti per la gestione di un utente con il ruolo di amministratore dello spazio in uno spazio	Sono state aggiunte informazioni sulla rimozione o la modifica del ruolo di un utente con il ruolo di amministratore di Space in uno spazio. Per informazioni, consulta Rimuovere o modificare il ruolo di un utente con il ruolo di amministratore dello spazio .	19 aprile 2023
Aggiornamenti per l'amministrazione degli ambienti di sviluppo	Sono state aggiunte informazioni sull'amministrazione degli ambienti di sviluppo come amministratore di Space. Per informazioni, consulta Amministrazione degli ambienti di sviluppo per uno spazio .	19 aprile 2023
Contenuto aggiornato: individuazione e visualizzazione dei problemi	Individuazione e visualizzazione dei problemi Argomento e argomenti secondari riorganizzati.	19 aprile 2023
Contenuto aggiornato: creazione di un progetto CodeCatalyst con un repository di progetti collegato GitLab	Aggiornato Creare un progetto per includere GitLab l'integrazione nella Creazione di un progetto con un repository di terze parti collegato sezione.	19 aprile 2023
Contenuto aggiornato: Configurazione delle immagini di calcolo e di runtime	È stato aggiunto il supporto per l'architettura Arm64 su Amazon Linux 2.	19 aprile 2023

Nuovi contenuti: spostamento dei problemi all'interno dei gruppi	È stata aggiunta la documentazione per lo spostamento dei problemi all'interno dei gruppi sulla Bacheca e le visualizzazioni di Tutti i problemi.	19 aprile 2023
Contenuti aggiornati: Quote per i flussi di lavoro in CodeCatalyst	È stato aggiornato l' Quote per i flussi di lavoro in CodeCatalyst argomento con le quote mancanti e la dimensione totale massima della quota delle variabili di output di una singola azione a 120 KB (da 2 KB).	18 aprile 2023
Nuovi contenuti: Visualizzazione del codice sorgente di un'azione	È stato aggiunto un Visualizzazione del codice sorgente di un'azione argomento.	18 aprile 2023
Nuovi contenuti: Arresto dell'esecuzione di un flusso di lavoro	È stato aggiunto un Arresto dell'esecuzione di un flusso di lavoro argomento.	10 aprile 2023
Nuovi contenuti: aggiunte sezioni per etichettare le risorse per le connessioni degli account tra AWS e Amazon CodeCatalyst	Sono state aggiunte informazioni per etichettare le risorse di connessione dell'account e gestire le politiche IAM per le risorse di connessione. Consulta Utilizzo dei tag per controllare l'accesso alle risorse di connessione dell'account e CodeCatalyst riferimento alle autorizzazioni .	6 aprile 2023
Nuovi contenuti: Tipi di operazione	È stato aggiunto un Tipi di operazione argomento.	6 aprile 2023

Contenuto aggiornato: Azione 'Implementa CloudFormation stack' YAML	È stata aggiornata la descrizione della <code>parameter-overrides</code> proprietà. Ora supporta i file JSON.	5 aprile 2023
Nuovi contenuti: creazione di un progetto CodeCatalyst con un repository collegato GitHub	È stata aggiunta una nuova sezione Creare un progetto intitolata Creazione di un progetto con un repository di terze parti collegato con le istruzioni per creare un progetto che si colleghi al GitHub repository.	5 aprile 2023
Contenuto aggiornato: utilizzo delle notifiche	È stata aggiornata la documentazione per le notifiche per includere informazioni sulla configurazione delle e-mail relative agli eventi del progetto.	31 marzo 2023
Nuovo contenuto	Pubblicazione iniziale della guida Amazon CodeCatalyst Action Development Kit .	31 marzo 2023
Contenuto aggiornato: ha ristrutturato la sezione Spaces in Amazon CodeCatalyst	È stata aggiornata la sezione Spaces rimuovendo le pagine di destinazione e consolidando gli argomenti.	29 marzo 2023

Contenuto aggiornato: Tutorial: distribuire un'applicazione su Amazon ECS	Modificato Passaggio 1: configurare un AWS utente e AWS CloudShell per descrivere come creare un utente in AWS IAM Identity Center anziché AWS Identity and Access Management. La creazione di utenti IAM non è più consigliata.	23 marzo 2023
Contenuto aggiornato: utilizzo dei ruoli	È stata aggiornata la documentazione per i ruoli di amministratore dello spazio, amministratore di progetto e collaboratore per includere le autorizzazioni per il collegamento dei problemi alle richieste pull.	13 marzo 2023
Contenuto aggiornato: utilizzo delle richieste pull	È stata aggiornata la documentazione per le richieste pull per includere informazioni sui problemi di collegamento alle pull request.	13 marzo 2023
Contenuto aggiornato: gestione dei problemi	È stata aggiornata la documentazione relativa ai problemi per includere informazioni sul collegamento dei problemi alle richieste pull.	13 marzo 2023
Nuovi contenuti: Visualizzazione dello stato e dei dettagli di tutte le esecuzioni del progetto	È stata aggiunta una sezione che descrive la nuova pagina di esecuzione del flusso di lavoro aggregato.	8 marzo 2023

Nuovi contenuti: Come posso correggere gli errori «La definizione del flusso di lavoro presenta <i>n</i> errori»?	È stata aggiunta una sezione su come risolvere gli errori «La definizione del flusso di lavoro contiene errori».	7 marzo 2023
Contenuto aggiornato: Creazione di un flusso di lavoro	Sono state aggiornate le istruzioni per riflettere la nuova interfaccia utente.	3 marzo 2023
Nuovi contenuti: Integrazione con universal-test-runner	È stato aggiunto un Integrazione con universal-test-runner argomento.	3 marzo 2023
Contenuto aggiornato: Crea, testa e distribuisce con i flussi di lavoro	Sono state aggiornate varie sezioni per riflettere i nuovi filtri del repository di origine, del ramo e dei nomi dei flussi di lavoro nella pagina di riepilogo dei flussi di lavoro.	2 marzo 2023
Nuovi contenuti: Monitoraggio dello stato di distribuzione tramite commit	È stata aggiunta una sezione sulla visualizzazione della qualità del codice e dello stato di implementazione tramite commit.	27 febbraio 2023
Nuovi contenuti: variabili BranchName " e CommitId "	Aggiunta una nuova variabile BranchName predefinita.	16 febbraio 2023
Contenuto aggiornato: gestione dei membri dello spazio in Amazon CodeCatalyst	Informazioni aggiornate sulla modifica dei ruoli dei membri, sull'invito e sulla rimozione dei membri in due nuove tabelle in base al ruolo assegnato all'utente in CodeCatalyst.	15 febbraio 2023

Contenuto aggiornato: sono stati aggiunti passaggi per la gestione PAT nella console Amazon CodeCatalyst	Sono stati aggiunti passaggi per la visualizzazione, la creazione e l'eliminazione PATs nella console.	15 febbraio 2023
Contenuto aggiornato: Specificazione delle immagini dell'ambiente di runtime	Sono stati aggiunti altri strumenti alla tabella delle versioni degli strumenti di immagine predefiniti.	10 gennaio 2023
Contenuto aggiornato: Condivisione di artefatti e file tra azioni	È stato corretto il percorso di un artefatto.	3 gennaio 2023
Contenuto aggiornato: GitHub Azione 'Azioni' YAML	È stato corretto lo snippet di codice nella Steps sezione.	3 gennaio 2023
Contenuto aggiornato: Connessione degli archivi di origine ai flussi di lavoro	È stato corretto un percorso di origine.	3 gennaio 2023
Contenuto aggiornato: aggiornamento di una pull request	È stata aggiornata la documentazione per includere informazioni sull'aggiornamento dei revisori obbligatori o facoltativi per una pull request.	23 dicembre 2022
Nuovi contenuti: Memorizzazione nella cache dei file tra le esecuzioni del workflow	Aggiunta una pagina per la memorizzazione nella cache dei file in un flusso di lavoro.	20 dicembre 2022
Contenuto aggiornato: utilizzo delle richieste pull	È stata aggiornata la documentazione per le richieste pull per includere informazioni sulle notifiche.	16 dicembre 2022

Nuovi contenuti: AWS CDK azione 'distribuisci' YAML	Aggiunta una nuova <code>CdkRootPath</code> proprietà.	16 dicembre 2022
Nuovi contenuti: Condivisione dell'elaborazione tra le azioni	È stato aggiunto un Condivisi one dell'elaborazione tra le azioni argomento.	14 dicembre 2022
Contenuto aggiornato: Condivisione di artefatti e file tra azioni	Esempi fissi che mostrano come specificare gli artefatti di input.	13 dicembre 2022
Nuovi contenuti: GitHub Azione 'Azioni' YAML	Aggiunta una pagina di riferimento dedicata all'azione GitHub Azioni.	13 dicembre 2022
Contenuto aggiornato: quote per progetti in CodeCatalyst	È stata aggiornata la documentazione con un massimo di 100 progetti in uno spazio.	2 dicembre 2022
Nuovo contenuto	Pubblicazione iniziale della Amazon CodeCatalyst User Guide.	1 dicembre 2022
Nuovi contenuti: Risoluzione dei problemi relativi alle estensioni	È stato aggiunto un argomento per la risoluzione dei problemi che gli utenti potrebbero riscontrare quando utilizzano la funzionalità delle estensioni di terze parti.	18 giugno 2022

AWS Glossario

Per la AWS terminologia più recente, consultate il [AWS glossario](#) nella sezione Reference. Glossario AWS